Doctoral Dissertation

Engaging Learning Concepts and Technologies for Fostering Digital Competences and Programming in Computer Science Education

Mag. Alexander Steinmaurer

Dissertation for the Award of the Academic Degree of a Doctor of Natural Sciences (Dr. rer.nat.) in Computer Science

submitted at

Graz University of Technology

Supervisor

Assoc.Prof. Dr.techn. Christian Gütl
Institute of Interactive Systems and Data Science
Graz University of Technology

Co-Supervisor

Prof. Dr. Andreas Dengel
Department of Computer Science and Mathematics
Goethe University Frankfurt am Main

Graz, June 2023

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material that has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present Ph.D. thesis.

Date, Signature

Abstract

Computer science education is essential in today's technology- and data-driven world. It equips people with the fundamental knowledge and skills to navigate, understand, and contribute to digital life. It opens up a wide range of career opportunities and promotes digital literacy among learners. Engaging learning approaches are needed to make these learners interested in computer science. These approaches enhance student motivation, active participation, and a deep understanding of concepts. In addition, they foster critical thinking, problem-solving skills, and creativity. However, the essential question is how students can be engaged in computer science, to be more precise, in digital competencies and programming education. Therefore, design principles and strategies for developing learning technologies and educational scenarios are derived from several studies conducted within this thesis's scope. Another central factor is the educational evaluation of these learning experiences to investigate their effectiveness. Further, this thesis proposes a methodological approach for designing, developing, and evaluating educational technologies and scenarios. The *Engaging Learning Tree* model is introduced for this. Additionally, the *Factors for Engaging Learning in Computer Science* (FELCS) model is presented that highlights the influence of five factors on engaging learning approaches: teaching and learning concepts, instructional strategies, digital learning platforms, educational assessment, and computer science education. In summary, the main goal of this dissertation is to create an engaging learning environment for learners and teachers in computer science education. The approaches and models presented in this thesis are intended to inspire future generations of computer science.

Kurzfassung

Ein fundierte Informatikausbildung ist in der heutigen technologie- und datengesteuerten Welt unerlässlich. Durch sie werden die Menschen mit den grundlegenden Kenntnissen und Fähigkeiten ausgestattet, um sich im digitalen Leben zurechtzufinden, es zu verstehen und beizutragen. Sie eröffnet ein breites Spektrum an Karrieremöglichkeiten und fördert die digitale Kompetenz der Lernenden. Um das Interesse der Lernenden an der Informatik zu wecken, bedarf es ansprechender Lernkonzepte. Diese Ansätze fördern die Motivation der Schüler:innen, die aktive Teilnahme und ein tiefes Verständnis der Konzepte. Darüber hinaus fördern sie kritisches Denken, Problemlösungsfähigkeiten und Kreativität. Die entscheidende Frage ist jedoch, wie Schüler:innen für die Informatik, genauer gesagt für digitale Kompetenzen und Programmierunterricht, begeistert werden können. Daher werden aus mehreren Studien, die im Rahmen dieser Arbeit durchgeführt wurden, Gestaltungsprinzipien und Strategien für die Entwicklung von Lerntechnologien und Lernszenarien abgeleitet. Ein weiterer zentraler Faktor ist die pädagogisch-didaktische Evaluierung dieser Lernerfahrungen, um ihre Wirksamkeit zu untersuchen. Darüber hinaus wird in dieser Arbeit ein methodischer Ansatz für das Design, die Entwicklung und die Evaluierung von Lerntechnologien und -szenarien vorgeschlagen. Hierfür wird das Modell des *Engaging Learning Tree* vorgestellt. Außerdem wird das *Factors for Engaging Learning in Computer Science* (FELCS)-Modell vorgestellt, das den Einfluss von fünf Faktoren auf ansprechende Lernansätze hervorhebt: Lehr- und Lernkonzepte, Unterrichtsstrategien, digitale Lernplattformen, Assessment und Informatikausbildung. Das Hauptziel dieser Dissertation besteht darin, eine ansprechende Lernumgebung für Lernende und Lehrende im Informatikunterricht zu schaffen. Die in dieser Arbeit vorgestellten Ansätze und Modelle sollen zukünftige Generationen von Informatiker:innen inspirieren.

Acknowledgements

After finishing my diploma degree in March 2020 and intending to start immediately with my Ph.D. I was looking forward to interesting and educational years with the possibility to travel to conferences, work on interesting cooperations and maybe have the chance to go abroad. My master's defense was on the 5th of March, just a few days before Covid-19 disrupted these optimistic plans. This means the first two years of my Ph.D. were characterized by lockdowns, home office, and distance learning. However, throughout my entire Ph.D. journey, there were a couple of remarkable people that made this time unforgettable.

First and most I want to thank my advisor Christian Gütl. Christian, you always gave me the right amount of advice, support, and guidance. You gave me the freedom to develop personally and professionally and to discover my own research field and my identity as a researcher. I learned a lot from you about academia and research. You taught me to think in slices and take one step at a time. You were always understanding of my passion for teaching and you always kept my back when I had a heavy workload.

I also want to thank my co-supervisor Andreas Dengel. Unfortunately, we met at the last metres of my Ph.D. but I really enjoyed working and learning with you. You supported my ideas and I really enjoyed the discussions with you about research methods, educational approaches, and learning technologies.

Next, I want to thank my colleagues from the CoDiS Lab Graz - Sylvia Ebner, Chiara Russ-Baumann, and Johanna Zeisberg for all the lovely conversations. I really enjoyed our lunch meetings and the wine tasting. Thanks also for the open office doors when I once just wanted to chat or complain about anything. My special appreciation goes to Alexander Nussbaumer. Within the last years, I could really learn a lot from you. You were a great discussion partner while writing my thesis and gave me valuable input. I always had a great time with your funny ideas like Spots, the fabulous JUCS explorer, or your fantastic wall calendar.

I started my Ph.D. almost at the same time as my two colleagues Aleksandar Bobić and Igor Jakovljević. Thank you both for your input during the last few years. I really enjoyed the exchange with you (especially during our late-night working sessions).

Since I spent most of my time at the Inffeldgasse, I have to say thank you, to my colleagues and good friends. Especially Thorsten Ruprechter and Niklas Hopfgartner. You made the first time in the office always funny. I think back with great enjoyment to our nightly discussions about teaching, research, and politics. You guys had always an open ear for me and gave me great input on my research projects. You guys had also a huge impact on my research direction, thanks for that! Another huge thank you goes to Gabi Leitner. Thank you so much for your help with all the organizational stuff and the nice conversations.

A huge part of my research, but also my daily life at TU Graz was related to teaching and learning. I would therefore like to say thank you to an excellent team that has always supported me in the best possible way. Thank you Aleksandar Karakaš, I really enjoyed teaching with you, I think we really did a great job! Within the last few years, I worked with fantastic teaching assistants, I want to thank especially David Kerschbaumer, Michael Guttmann, and Sebastian Gürtl. Your passion for teaching was always inspiring for me, even when I had times when everything was too much.

During my Ph.D. I had the honor to (co-)supervise 15 Bachelor's thesis and 6 Master's thesis. During this time I worked with several great students on interesting research projects. My special thanks go to the following students who really supported me with their work: David Eckhard, Sebastian Gürtl, Daniel Pollhammer, Michael Umfahrer, Martin Sackl, and Christoph Schatz.

Finally, I want to say thank you to my parents, my girlfriend, my brother, and all of my friends. All of you encouraged me during the last few years and you supported me in all situations. All I can say is that I wouldn't be where I am now without this great support.

Contents

1 Introduction

"Computer science is not about computers, any more than astronomy is about telescopes"

Edsger W. Dijkstra

Computer science is a fascinating field because it allows for endless innovation. Its influence can be seen in various aspects of our lives, from technological advancements to how society works. Traditionally, computer science combines areas such as logic, mathematics, and electrical engineering. In recent years, it has given rise to new disciplines such as data science, geoinformatics, computer linguistics, artificial intelligence, business informatics, and computational social science, among countless others. Computer science has affected almost every aspect of daily life, impacting various fields, even for those who are not computer scientists themselves.

Since computer science permeates all these areas, it is essential to start developing digital skills as early as possible. There are many initiatives and programs to engage young people in computer science, even starting in kindergarten. Technologies such as simple programmable robots, child-friendly programming environments, microcontrollers, or logic puzzles should foster children's digital competencies and skills. Subjects around computer science are an integral component of today's K-12 education. National curricula, standards, frameworks, or extracurricular efforts are intended to improve computer science education. The learning objectives of these initiatives are as wide-ranging as the field of computer science itself. It encompasses various areas such as computational thinking, problem-solving, social implications of technology, digital content creation, or programming.

1.1 Motivation

Over the past few decades, technology has transformed how humans learn and become essential to modern education. Emerging technologies and digital trends such as mobile technologies, mixed reality, large language models, and artificial intelligence are already used in various educational contexts and support both teaching and learning. (Hashim, 2018). The use of technology in the classroom naturally raises scientific questions related to advantages and disadvantages, but also opportunities and risks.

One of the key benefits of technology in education is that it can facilitate more personalized and flexible learning experiences (H. Peng et al., 2019). Technologies such as adaptive learning and intelligent tutoring systems can tailor the learning experience to each student's needs and abilities, providing more targeted support and feedback (Shemshack et al., 2021). This can improve learning achievements by enabling students to learn self-paced and receive individual feedback.

Another important benefit of technology in education is that it can support more collaborative and social learning experiences. Online discussion forums, collaborative tools, and social media platforms can facilitate communication and collaboration among learners, enabling them to collaborate on projects and share knowledge and ideas. This can help develop important skills such as communication, teamwork, and critical thinking, essential to succeed in the modern workplace (Al-Samarraie & Saeed, 2018).

In addition to these benefits, technology in education can also provide opportunities for immersive and engaging learning experiences. Virtual and augmented reality technologies can create interactive and immersive learning environments that can help to visualize abstract concepts. These technologies enable hybrid learning environments, combining traditional classrooms with virtual spaces. This facilitates the students' skills in problem-solving, communication, collaboration, but also critical thinking (Dunleavy et al., 2009).

Overall, a plethora of research studies show the impact of technology on learning and its potential to enhance educational achievements greatly. Nevertheless, it is important to recognize that technology alone is not a cure-all for all educational challenges. It should be employed thoughtfully and used with effective teaching methodologies to maximize its benefits aiming for ideal learning outcomes and meaningful learning experiences.

1.2 Objectives

This dissertation focuses on engaging learning in computer science education, to be more precise, in digital competencies and programming education. Two central research questions form the core of this thesis. To answer these questions, several scientific studies have been conducted. Learning technologies and educational scenarios were designed, developed, and evaluated within these studies.

Research Question 1

What design principles and strategies should be employed in developing learning technologies and educational scenarios to ensure an engaging learning experience in digital competencies and programming education?

The first research question aims to answer what design principles and strategies are relevant for designing and developing learning technologies and educational scenarios. For this purpose, the *FELCS* model (Factors for Engaging Learning in Computer Science) (see Figure 1.1) was developed consisting of five components. According to this model, engaging learning in computer science is directly influenced by four components: i) teaching and learning concepts, ii) instructional strategies, iii) online learning platforms, and iv) educational assessment. The foundation for the entire model is v) computer science education. Within the scope of this thesis, the individual components of the model, but also the connection between these components, will be analyzed and discussed. This model should help to identify which factors influence engaging learning.

Research Question 2

How can the effectiveness of different types of engaging learning experiences and their characteristics be evaluated within the context of digital competencies and programming education?

While the first research question (RQ1) emphasizes the design principles of learning technologies and educational scenarios, this research question is dedicated to evaluation aspects. All learning experiences that are developed within this thesis follow the process model of the *Engaging Learning Tree* (see Figure 1.2). Within this model, both a data- and user-centric evaluation are proposed to make assumptions regarding the effectiveness of a learning experience. Several evaluation methodologies and instruments will be analyzed and discussed within this thesis.

1.3 Methodology

This work aims to design, implement and evaluate various learning technologies and scenarios in specific fields of computer science education, such as digital competencies and programming education. In addition, the impact of these educational

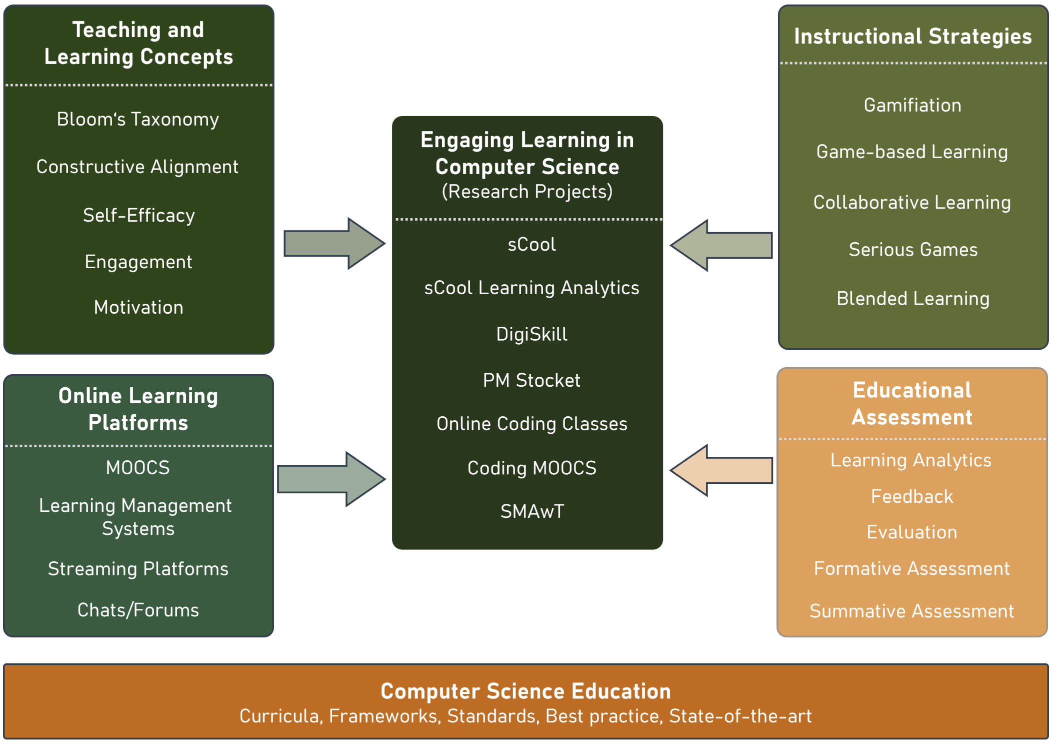

Figure 1.1: This figure shows the *FELCS* model, which shows four factors that influence engaging learning experiences in computer science.

approaches on students in secondary school and higher education will be investigated in different learning contexts. Figure 1.2 presents the *Engaging Learning Tree* model, which is a process model for the design, development, and evaluation of engaging learning experiences.

An initial motivation initiates the beginning of each educational scenario. Within this context, three initial motivators have been identified and are building the roots of engaging learning: i) a user study, ii) an educational dataset, or iii) a research gap. *User studies* open the floor for further questions conducted in an educational context. A study can be conducted in formal learning contexts (schools or universities) and informal contexts (online communities). The initial motivation can also be an *educational dataset* that is retrieved from learning interventions or learning technologies. Another trigger for an initial motivation can be an identified *research gap*, which means that there are open questions worth a closer investigation in the current research literature. One of these motivators or a combination of them leads to a comprehensive *literature survey*. A robust stem in terms of the Engaging Learning Tree is an extensive review of the current literature. This aims to understand theoretical models and related work better. A solid background helps narrow the research questions and set a specific focus. The theoretical foundation

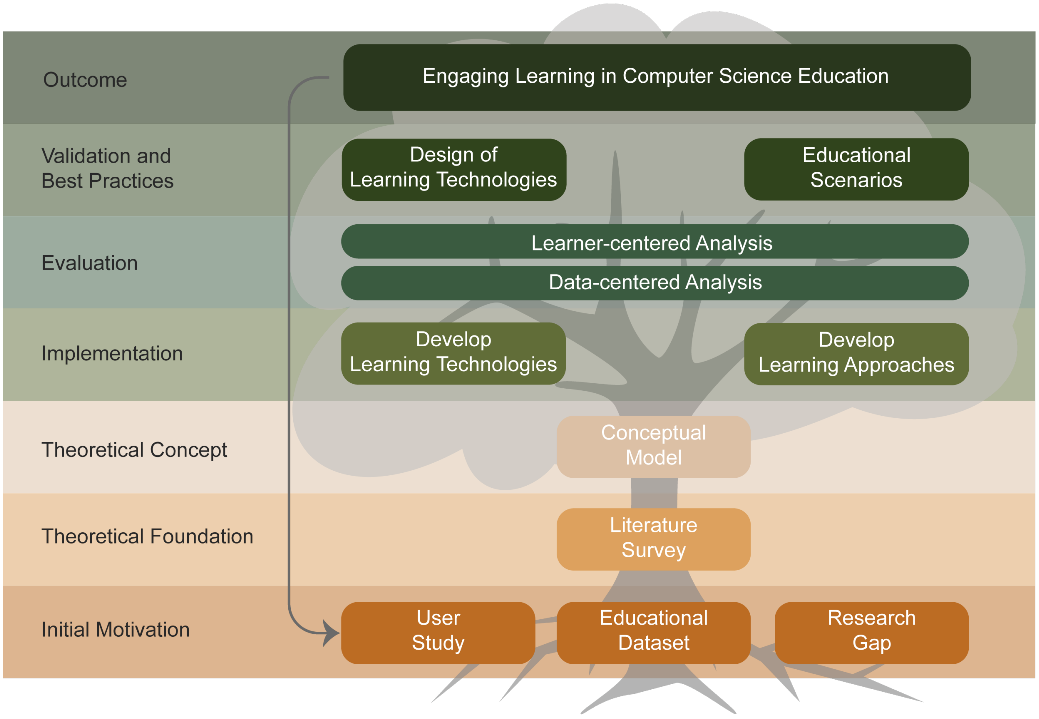

Figure 1.2: Engaging Learning Tree - Methodological approach within this thesis.

paves the way to a *conceptual model*. This high-level abstraction model includes key components and constraints for learning technologies or approaches.

Further, the tree runs into two parallel branches, which run side by side. The left branch is related to *learning technologies*. A learning technology is developed based on a conceptual model. This can either be a new technology implementation or a modification of existing ones. This learning technology will be applied and evaluated in an educational context. The evaluation step is both learner- and data-driven. Learner-oriented evaluations can be questionnaires, expert evaluations, interviews, or focus groups, while data-oriented evaluation approaches are behavioral and learning analytics based on user-generated data. This can be qualitative data, such as user surveys, and quantitative data, such as interaction log data. The step of *Design of Learning Technologies* follows the evaluation. All findings and conclusions that result from the evaluation are gathered for implications and best practice recommendations. Validation is a critical step in the design of learning technologies to ensure their effectiveness and impact. This step also includes considering the technology's social, ethical, and educational aspects.

The right branch is related to *learning approaches and scenarios*. These approaches are usually characterized by a learning technology used to deliver them. They are related to digital learning and will therefore generate educational data. Again, a learner- or data-oriented evaluation of the learning approach within an educational context is conducted. Again, this evaluation can compromise qualitative or quantitative data. This helps to understand better the learner's perspective of a learning approach or technology. The observations and findings will present well-evaluated educational scenarios where valuable implications can be derived.

Finally, both branches - learning technologies and educational approaches - run together to the treetop, which is *Engaging Learning in Computer Science Education*, which is the actual outcome. This process can be iterative, which means that an engaging learning experience can - again - be the basis for an initial motivation and consequently be starting point for another educational question or project.

1.4 Publications and Contribution

Several studies have been conducted to answer the research questions defined in this thesis. The FELCS model (Figure 1.1) illustrates the connection between *teaching and learning concepts*, *instructional strategies*, *online learning platforms*, and *educational assessment* and their impact on engaging learning approaches and technologies. Computer science education builds the foundation since this thesis focuses on digital competencies and programming education. The influence of these factors on engaging learning was observed in different studies.

The following peer-reviewed publications are the results of these studies and are building the main contributions of the dissertation:

1. **Steinmaurer, A.**, Sackl, M., & Gütl, C. (2021). *Engagement in In-Game Questionnaires - Perspectives from Users and Experts*. In: Proceedings of 7th International Conference of the Immersive Learning Research Network (iLRN), 2021, pp. 1-7.
2. **Steinmaurer, A.**, Tilanthe, A.K., & Gütl, C. (2021). *Designing and Developing a Learning Analytics Platform for the Coding Learning Game sCool*. In: Proc. 14th Int. Conf. Interact. Mob. Commun. Technol. Learn.
3. **Steinmaurer, A.**, Eckhard, D., Dreveny, J. & Gütl, C. (2022). *Developing and Evaluating a Multiplayer Game Mode in a Programming Learning Environment*. In: Proceedings of the 8th International Conference of the Immersive Learning Research Network (iLRN), 2022, pp. 1-8.
4. **Steinmaurer, A.**, Bajramovic, A., Pollhammer, D. & Gütl, C. (2022). *Learning Security Awareness in Email Communication Using a Platform for Digital Skill*

Teaching. In: Proceedings of 2022 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE).

5. **Steinmaurer, A.** & Gütl, C. (2022). *Implementation and Experiences of a Flipped Lecture Hall – A Fully Online Introductory Programming Course*. In: Proceedings of the 25th International Conference on Interactive Collaborative Learning (ICL).
6. **Steinmaurer, A.**, Schatz, C., Krugel, J. & Gütl, C. (2022). *Analyzing Behavioral Patterns in an Introductory Programming MOOC at University Level*. In: Proceedings of the 8th IEEE Learning with MOOCS (LWMOOCS).
7. **Steinmaurer, A.**, Dengel, A., Umfahrer, M., Zöhrer, K., Kogler, P., & Gütl, C. (2023). *A SMAwT Approach for Raising Social Media Awareness in Secondary CS Education - Maybe a Little Bit Too Much?* Manuscript submitted for publication.

Article 1, Article 2, and Article 3 build on previous work on the serious game *sCool*, which was initially developed at Graz University of Technology by Kojic et al. (2018) and further developed and evaluated by Steinmaurer et al. (2019). Feedback and assessment are essential components of an effective learning process for both the learners and teachers. For this reason, an engaging way to collect player feedback was integrated into the sCool platform using in-game questionnaires. The implementation raised interest in further assessment aspects, which led to the development of a learning analytics platform. Combining survey data and players' interaction from the video game helps educators design and develop meaningful learning scenarios and foster the students' learning process. After numerous workshops and evaluations using the sCool platform, the idea of a multiplayer game type came up, to foster collaborative learning. For this reason, a cooperative game type was developed and evaluated in school classes and with teachers.

The development of the *DigiSkill* platform (Article 4) was driven by the growing importance of digital skills in school education. The web-based platform offers a modular and adaptable environment that allows teachers to design courses for their students. Although the platform offers various modules, including coding, spreadsheets, data visualization, mail, and web browsing, the focus is on IT security awareness. A course was created on this topic to encourage student engagement and was subsequently assessed to determine their learning outcomes and level of engagement.

In addition to skills like coding and IT security, social media literacy is a fundamental component of many educational programs. To address this need, the *Social Media Awareness Training* (SMAwT) platform (Article 5) was created, drawing on expertise in the design and development of learning technologies. The SMAwT course is built upon the Dagstuhl Triangle, a model for digital education that incorporates technological, socio-cultural, and application-oriented perspectives to cultivate social media skills. The SMAwT project aims to promote social media competencies by integrating these three perspectives.

Article 6 and Article 7 compromise introductory programming courses in higher education. Many students and a heterogeneous group of learners usually characterize these courses. Paper Article 6 presents the implementation of a fully online course in a flipped classroom setting and evaluates the students' experiences. Article 7 focuses on a massive open online course (MOOC) that covers the fundamentals of object-oriented programming and is offered by the Technical University of Munich. Within the study, the learners' behavioral patterns are analyzed to reduce drop-out rates and increase their engagement and achievement.

This dissertation centers around the intersection of technology and education. In addition to the already mentioned contributions, several other papers were written during the Ph.D. While developed in the same context, the following papers and projects were primarily collaborative efforts with other researchers.

8. Mosquera, C. K., **Steinmaurer, A.**, Eckhardt, C., & Gütl, C. (2020). *Immersively Learning Object Oriented Programming Concepts With sCool*. In: 6th International Conference of the Immersive Learning Research Network (ILRN).
9. Pirker, J., **Steinmaurer, A.**, & Karakas, A. (2021). *Beyond Gaming: The Potential of Twitch for Online Learning and Teaching*. In: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1. Association for Computing Machinery, New York, NY, USA, 74–80.
10. **Steinmaurer, A.**, & Gütl, C. (2022). *Computerspiele im Informatikunterricht. In Digitale Spiele und fachliches Lernen* (Vol. 1, pp. 211–247).
11. **Steinmaurer, A.**, Mesarec, B., Mesarec, T., Pietroszek, K. & Gütl, C. (2022). *Mobile XR Interface for Quantum Computing*. In: 1st Int. Workshop on Analytics, Learning & Collaboration in eXtended Reality - ACM International Conference on Interactive Media Experiences: IMX 2022 (pp. 243-248). Aveiro, Portugal.
12. Rebol, M., Pietroszek, K., Ranniger, C., Hood, C., Rutenberg, A., Sikka, N., **Steinmaurer, A.** & Gütl, C. (2022). *Work-in-Progress-Volumetric Communication for Remote Assistance Giving Cardiopulmonary Resuscitation.* In: 8th International Conference of the Immersive Learning Research Network (ILRN), 1–3.
13. Rebol, M., **Steinmaurer, A.**, Gamillscheg, F., Pietroszek, K., Gütl, C., Ranniger, C., Hood, C., Sikka, N. & Rutenberg, A. (in 2023). *CPR Emergency Assistance Through Mixed Reality Communication*. In: Frasson, C., Mylonas, P., Troussas,

C. (eds) Augmented Intelligence and Intelligent Tutoring Systems. ITS 2023. Lecture Notes in Computer Science, vol 13891. Springer, Cham.

14. Dengel, A., **Steinmaurer, A.**, Müller, L. M., Platz, M., Wang, M., Gütl, C., Pester, A., Morgado, L. and Amoona, N. (in press). Research Agenda 2030: The Great Questions of Immersive Learning Research. *9th International Conference of the Immersive Learning Research Network (iLRN2023)*.

1.5 Thesis Structure

Chapter 2 represents the theoretical foundation of this dissertation and covers different aspects of engaging learning technologies. The structure of this chapter is based on the FELCS model (Figure 1.1). Section 2.1 gives an overview of the field of computer science education. This compromises a historical overview of the discipline, curricula, standards and framework. Section 2.2 covers fundamental learning and teaching concepts, including taxonomies and learning objectives. This section also gives an overview of learning relevant concepts such as self-efficacy, motivation, and engagement. In Section 2.3, instructional strategies in computing education that are relevant within the scope of this thesis are presented. These strategies are traditional learning approaches, game-based learning and gamification, blended learning, and computer science unplugged. Section 2.4 compromises different online learning platforms covered in this thesis. This covers MOOCS, learning management systems, streaming platforms, and serious games. Another central aspect of the learning process is assessment which is covered in section 2.5. Related work is covered in section 2.6. Chapter 3 contains all research studies conducted within this thesis. Finally, Chapter 4 discusses and concludes the findings of this thesis.

2 Engaging Learning Technologies

"In learning you will teach, and in teaching you will learn."

Phil Collins

Technology is a key component in nowadays teaching and learning. It is used in kindergarten, schools, higher education, and also further education. This thesis focuses on engaging learning in the field of computer science education. For this reason, it is necessary to give an overview of relevant concepts, frameworks, and models that are commonly used in research.

Firstly, the discipline of computer science education and its related curricula, standards, and frameworks are introduced. Building upon this foundation, relevant teaching and learning concepts are then introduced. In traditional learning situations, the primary focus is on the interaction between teachers and students. Teachers design and deliver instructional content to impart knowledge and skills to students actively engaged in the learning process. Therefore, one obvious question can be asked: *How can the learning content be effectively taught?* For this reason, learning taxonomies have been developed to categorize and organize learning outcomes. Another important aspect, especially when covering engaging learning approaches, is to explore possibilities to increase the students' motivation and engagement.

A way to promote effective teaching is through meaningful instructional strategies. For this reason, concepts such as game-based learning, gamification, blended learning, or unplugged learning are introduced in computer science education. Especially in online learning, the educational content and instructional strategies are often delivered by online platforms such as MOOCS, learning management systems, or social networks. Finally, different aspects of educational assessment are covered as an instrument to receive valuable insights into the students' learning and achievement. By using assessment effectively, educators can enhance the learning experience.

These components build the foundation of engaging learning in computer science education. Therefore, this chapter is structured analogously to the FELCS model (see Figure 1.1) described in the introduction.

2.1 Computer Science Education

Computer science is a discipline that permeates almost all areas of our daily lives. Therefore, it is also an integral part of school curricula and can be found in many university degrees. Computing education compromises a wide range of topics, including coding, computational thinking, social implications of computing, digital communication, digital content creation, and much more. Through computer science education, students learn not only the technical aspects of computing but also gain problem-solving abilities, critical thinking skills, and a better understanding of the ethical implications of technology. Computer science education encourages creativity, innovation, and collaboration, empowering individuals to design and develop innovative solutions for complex problems. It also prepares students for careers in various fields. By imparting computational thinking and digital literacy, computer science education equips individuals to make meaningful decisions and participate in the digital economy and society. Even though computing is a relatively new field, it plays an important role in nearly every type of education.

2.1.1 A Brief History of Computing Education

The story of computer science education goes back to the mid-20th century. Back in the day, computer science was not an independent discipline; it was primarily in the domain of mathematics or engineering. When computers became widespread, universities offered courses in computer programming or related fields. No curricula were available at this time, so universities had to develop their own ones (Tedre et al., 2018). In the 1960s, the discussion on the nature of computer science as an independent subject emerged, and conferences were held related to this topic. In 1962 the *Curriculum Committee on Computer Science* was formed within the Association for Computer Machinery (ACM). This committee developed the so-called *Curriculum '68*, the base for computer science programs from many US universities for a decade. This curriculum recommended several semester hours and specific subjects and topics for a computer science major (Austing et al., 1977; Gupta, 2007).

Before the 1970s, computers were large and expensive. They were mainly available for universities and companies. Therefore, computer science education has primarily been part of higher education for many years. In the 1970s, computers became more accessible. In 1972 Dennis Ritchie developed the C programming language at Bell Laboratories (Ritchie & Kernighan, 1988). This language is strongly associated with the UNIX operating system that Kenneth Thompson and Dennis Ritchie developed in 1969. Over the following years, the popularity of C highly increased in different contexts. Until now, C is a programming language used in many universities as the first programming language (Fu et al., 2017).

The occurrence of personal computers in the 1980s strongly drove computing education. Computers became widely available in companies, schools, and also at home. This required users to grow their computer literacy and skills. Another important aspect was the increasing popularity of the BASIC programming language in the 1980s. Programming languages such as BASIC, Pascal, and C were widely used in schools and universities (Lamb & Johnson, 2011). The biggest acceleration in computer science education occurred in the early 1990s when the World Wide Web became increasingly available.

In 2003 the Scratch programming language was developed at the Massachusetts Institute of Technology (MIT). This visual language was mainly developed for children to write code and learn to program. Scratch introduced the concept of block-based programming, where graphical command blocks are used to compose programs (Lamb & Johnson, 2011). This inspired many block-based approaches, such as PocketCode, Blockly, or AppInventor.

Overall the 2000s witnessed significant developments and advancements in computing education. With the rapid growth of technology and the Internet, computing education expanded and evolved in many ways. Online learning platforms and resources emerged, providing new possibilities for self-paced and collaborative learning (Palaigeorgiou & Papadopoulou, 2019). The invention of mobile devices opened the doors for mobile apps in schools. Additionally, there was a growing recognition of the importance of computational thinking, problem-solving skills, and digital literacy across disciplines, resulting in the integration of computing in K-12 education (Yadav et al., 2016). The 2000s also saw an increased emphasis on coding and programming skills, with initiatives promoting coding education and coding bootcamps. Furthermore, efforts were made to bridge the gender gap in computing by encouraging girls and women to pursue careers through programs, and organizations focused on promoting diversity and inclusivity (Cheryan et al., 2015). Overall, the 2000s marked a period of significant growth and diversification in computing education, reflecting the increasing role of technology in society and the need for digital skills in various domains.

2.1.2 Curriculas, Standards, and Frameworks

Not only schools have an impact on the topics of computer science education, various companies and institutions also contribute to the design of standards and frameworks. Organizations such as the Computer Science Teachers Association (CSTA), code.org[1], and CSforAll aim to make computer science more visible and available. These programs have different orientations, focuses, and goals. code.org, for example, is dedicated to promote female students and minorities in school education. Initiators of these programs intended to increase the quality of programming education in schools, which was underrepresented in many curricula. They offer learning materials for both learners and teachers. Educators are provided with additional resources or lesson plans in many of these programs. This goes from simple puzzles over robotics to artificial intelligence.

However, besides these extracurricular initiatives, computer science has gained significant recognition as a school subject over the whole globe. In Europe, many countries have incorporated computer science into their educational systems, recognizing its importance in the digital age. Some countries, such as Estonia, have implemented comprehensive coding and programming curricula from an early age already. Estonia's *ProgeTiiger* program started in 2012 and is funded by the Estonian Ministry of Education and Research aiming to bring computer science education already in primary schools (Hsu et al., 2019). Other countries, such as the United Kingdom (especially England), have already made *Computing* a mandatory subject at the age of 5 years (Sentance & Csizmadia, 2017). Similarly, in the United States, computer science education has seen a growing emphasis in recent years. Many states have introduced computer science standards and initiatives to ensure that students are equipped with the necessary skills for future careers in technology. Additionally, coding and computational thinking have gained prominence in the curriculum, reflecting the increasing demand for digital literacy.

K–12 Computer Science Framework

The K-12 Computer Science Framework (K-12 Computer Science Framework Steering Committee, 2016) was developed in collaboration with influential institutions such as Association for Computing Machinery (ACM), Code.org, Computer Science Teachers Association (CSTA), Cyber Innovation Center, and National Math and Science Initiative. They aimed to develop computer science education guidelines in primary and secondary schools (K-12). The framework provides a comprehensive

[1]https://code.org/

set of learning objectives and practices for computer science education, including computational thinking, programming, and computer systems and networks. It is designed to be flexible and adaptable to different contexts and curricula and to support the development of equitable and inclusive learning environments for all students. The framework includes five core concepts and seven core practices. The concepts are i) computing systems, ii) networks and the Internet, iii) data and analysis, iv) algorithms and programming, and v) impact of computing. The seven core practices compromise: i) fostering an inclusive computing culture, ii) collaborating around computing, iii) recognizing and defining computational problems, iv) developing and using abstractions, v) creating computational artifacts, vi) testing and refining computational artifacts, and vii) communicating about computing.

The framework serves as a guide for developing standards, lesson plans, curricula, or extracurricular programs. Many various initiatives are motivated by the K-12 CS Framework. One related standard is the so-called *CSTA K-12 Standards*. Another effort is the Model Curriculum for K-12 CS from ACM (Tucker, 2003a).

CSTA K-12 Standards

The *Computer Science Teachers Association* is a community of computer science teachers mainly based in the US and Canada aiming to support K-12 educators. The CSTA K-12 Standards were released in 2017 and consist of five core concepts, which are again divided into subconcepts (Computer Science Teachers Association, 2017):

- Computing Systems: Devices, Hardware & Software, Troubleshooting
- Networks & the Internet: Network, Communication & Organization, Cybersecurity
- Data & Analysis: Storage, Collection, Visualization, & Transformation, Interfence & Models
- Algorithms & Programming: Algorithms, Variables, Control, Modularity, Program Development
- Impact of Computing: Culture, Social Interactions, Safety, Law, & Ethics

The five main concepts are part of each educational level. CSTA K-12 defines five levels, beginning from 1A (grades K-2), 1B (grades 3-5), over 2 (grades 6-8) to 3A (grades 9-12) and 3B (grades 11-12).

Digital Competence (DigComp) Framework

At the European level, digital literacy efforts are implemented through the Digital Competence Framework for Citizens (DigComp) (Vuorikari et al., 2022). The reference framework was first defined in 2006. Version 2.1, introduced in 2017, comprises five competent areas with 21 competencies assigned to these areas. These areas are divided into i) information and data literacy, ii) communication and collaboration, iii) digital content creation, iv) safety, and v) problem solving. EU countries implement this framework differently. Austria, for example, has four competence models:

- **digi.komp4** - Digital competencies and educational examples for elementary schools (until 4th grade)
- **digi.komp8** - Digital competencies and educational examples for secondary school lower level (until 8th grade)
- **digi.komp12** - Digital competencies and educational examples for secondary school higher level (until 12th grade)
- **digi.kompP** - Digital competencies and educational examples for teachers

National CS Curricula

Hubwieser et al. (2015) systematically analyzed curricula from 12 countries using qualitative text analysis. The authors concluded that K-12 curricula in computer science are implemented in many different ways in the countries. These differences can be identified already when comparing terms. In the field of computing education, 40 different terms are used in different fields. When it comes to technological aspects, terms such as computer and technology, computer technology, digital technology, etc., are used in the curricula. The authors further identified several concepts that are covered in all observed countries, such as problem-solving, programming, operating systems, or algorithmic concepts.

Exemplary the Austrian computer science curriculum for general secondary schools is shown. Computer science is a mandatory subject in the 9th grade consisting of four sub-categories (Micheuz et al., 2017):

- **Applied computer science** - Standard software for publication, calculation, visualization; information acquisition, systematization, structuring, and evaluation. Using information systems for the purpose of learning.
- **Information systems** - Operating systems, computer networks, terminals
- **Practical computer science** - programming languages, algorithms, automaton,

data structures, databases, database modeling

- **Computer science, people, and society** - Importance of Computer Science, Data Security, Legal Aspects, Computer Science Professions

2.2 Teaching and Learning Concepts

Teaching and learning are two processes within an educational context. While the aim of teaching is to impart skills and knowledge to students, learning is the process where learners acquire these knowledge and skills. Teaching and learning are highly related, it includes different instructional strategies and methods. These concepts are important for effective learning and enhancing students' understanding. Teaching and learning are multi-perspective concepts that go way beyond the scope of this thesis. For this reason, this section covers the most important ones, which are the foundation for engaging learning. Among the relevant concepts are taxonomies, constructive alignment, self-efficacy, engagement, and motivation. By incorporating these teaching and learning concepts into educational practices, educators can create meaningful and engaging learning experiences that promote deep understanding, critical thinking, and long-term retention of knowledge and skills.

2.2.1 Taxonomies

Learning taxonomies are educational frameworks used to organize learning objectives and learning outcomes. They are commonly used in education and instruction for designing curriculums, assessments, and instructional strategies. A highly influential taxonomy is *Bloom's Taxnonomy* developed in 1956 by Benjamin Bloom and a team of cognitive psychologists. The taxonomy has three domains, cognitive, affective, and psychomotor. The most commonly applied domain is the *cognitive* domain, which consists of six levels that build upon each other. These levels are: i) Remembering, ii) Comprehending, iii) Applying, iv) Analyzing, v) Synthesizing, and vi) Evaluating (Bloom et al., 1956). The taxonomy was revised by Anderson and Krathwohl (2001) and consists of a two-dimensional framework that includes knowledge and cognitive process dimension. The authors define four types of knowledge: factual, conceptual, procedural, and meta-cognitive knowledge. Compared to Bloom's original taxonomy Anderson and Krathwohl used verbs instead of nouns to label the six levels. Additionally, both categories on top were re-positioned in the revised taxonomy.

- **Factual Knowledge** is related to the basic terminology and specific details

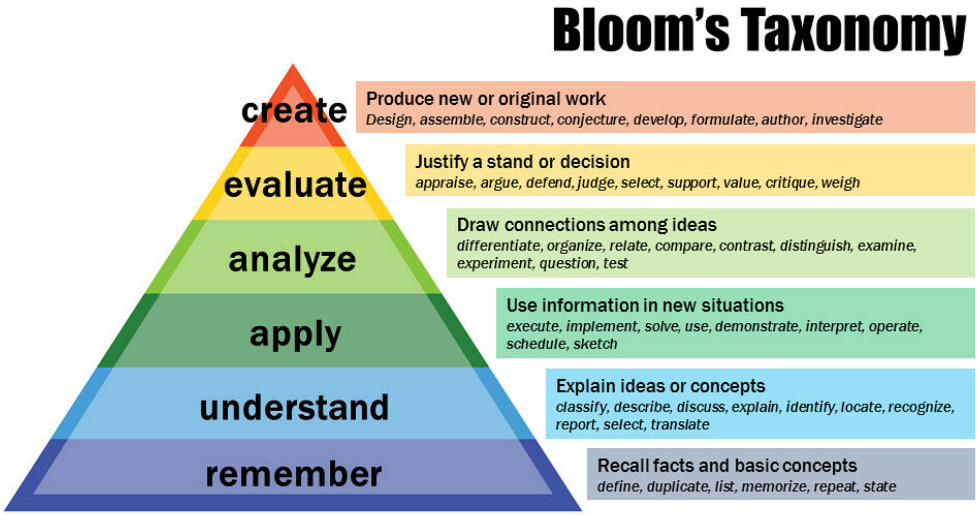

Figure 2.1: Bloom's Taxonomy Revised consists of six levels, that build upon each other. From "Bloom's Revised Taxonomy," by Vanderbilt University Center for Teaching, 2016, https://www.flickr.com/photos/vandycft/29428436431. CC BY.

and elements that are needed to understand a subject and solve problems.

- **Conceptual Knowledge** are classifications, categories, theories, models, or principles. They are needed to understand the link between basic elements within a larger system and how they interact with each other.
- **Procedural Knowledge** is about subject-specific skills, algorithms, techniques, or methods. This type of knowledge determines the criteria for when to use which procedures.
- **Meta-cognitive Knowledge** covers strategic knowledge and self-knowledge, but also knowledge about cognitive tasks (context). Meta-cognitive knowledge is the awareness of the own cognition.

The cognitive process dimension has six levels, that are similar to Bloom's original taxonomy including i) Remember, ii) Understand, iii) Apply, iv) Analyze, v) Evaluate, and vi) Create. Again, the levels are in a hierarchical order, from lowest to highest thinking skills. Figure 2.1 illustrates these levels and exemplary shows verbs that are related to each level. These action verbs are a suitable way to express learning objectives on each level (Adams, 2015).

Bloom's taxonomy is successfully applied in many areas of K-12 and higher education. Furthermore, Bloom's taxonomy is also widely used in various areas of computer science education. Well-known standards or activities such as CSTA K-12 or Bebras are influenced by this taxonomy (Dagiene et al., 2020; Robins, 2019). In a literature review from Masapanta-Carrión and Velázquez-Iturbide (2018) the authors argue that Bloom's taxonomy was by far most used in the reviewed articles. It is mostly used in CS introductory courses (CS1) at a university level, but also in high schools. The taxonomy is mainly used for assessment purposes. However, there are also difficulties reported when using the taxonomy. The major problem is classifying specific activities into taxonomy levels. A similar observation was made by C. G. Johnson and Fuller (2006). They conducted an analysis where a group of assessors had to assign different assessments to a taxonomy level. The study showed a high mismatch between the levels *application* and *analyze*. They hypothesize that in the field of computer science, the level of application might be more relevant. Therefore, an additional level *Higher Application* could be added. This level goes beyond applying and means critical dealing with the work.

Besides Bloom's taxonomy C. G. Johnson and Fuller (2006) and Masapanta-Carrión and Velázquez-Iturbide (2018) illustrates the so-called *Structure of the Observed Learning Outcome* (SOLO) taxonomy is also used in the field of computing. The taxonomy was developed by J. B. Biggs and Collis (1982) consisting of five levels of understanding.

1. At the **pre-structural** level, the learner lacks any understanding or knowledge of a concept or task. They may have misconceptions or make errors due to a lack of familiarity with the topic.
2. At the **uni-structural** level, the learner acquires one relevant aspect or concept related to the task. They have a limited understanding and can only apply or describe one piece of information.
3. The **multi-structural** level involves the learner grasping multiple independent aspects or concepts related to the task. They can identify and understand several disconnected pieces of information, but they may have issues integrating them or seeing the bigger picture.
4. At the **relational** level, the learner can make connections and see relationships between the various aspects or concepts related to the task. They can analyze and organize information, identify patterns, and understand how different elements relate to one another.
5. The **extended abstract** level represents the highest level of understanding. Learners at this level can think critically and creatively, extending their understanding beyond the given information. They can apply their knowledge to new contexts, generate new ideas, and develop a deep and sophisticated understanding of the topic.

Other than Bloom's taxonomy, SOLO taxonomy is learner-focused and emphasizes the quality and depth of understanding, rather than the cognitive processes. It highlights the progression from surface-level understanding (uni-structural and multi-structural) to deep and connected understanding (relational and extended abstract).

Lister et al. (2006) used the SOLO taxonomy for the evaluation of novice programmers. They argue that educators have to assess the students' skills on a relational level. This shows that students can read lines of code and bring them into context and structure. A study by Brabrand and Dahl (2009) shows that in the field of computer science, the competencies related to the relational level are dominant since they are often related to verbs such as 'explain', 'analyze', 'implement', 'compare', and 'construct'.

2.2.2 Constructive Alignment

Constructive alignment (CA) is an educational principle that emphasizes the alignment of learning objectives, teaching activities, and assessment methods. The concept was developed by educational psychologists John Biggs and Catherine Tang in 1994. The main idea behind constructive alignment is to ensure that there is a meaningful connection between what students are expected to learn, how they are taught, and how their learning is assessed (J. Biggs & Tang, 2011).

The starting point in constructive alignment is to define clear and specific learning outcomes or objectives. These outcomes describe the knowledge, and skills that students should acquire by the end of a course or instructional unit. Learning outcomes are typically written using action verbs that indicate observable and measurable behaviors, allowing educators to assess students' achievement (J. Biggs & Tang, 2011).

As soon as the learning outcomes are defined, the teaching activities and learning experiences are aligned with those outcomes. This involves selecting appropriate instructional strategies, resources, and assessments that support the intended learning goals. The teaching activities should be designed in a way that enables students to actively engage with the content, develop the desired skills, and achieve the learning outcomes (J. Biggs & Tang, 2011).

Finally, the assessment methods and criteria are aligned with the intended learning outcomes and teaching activities. The assessments are designed to measure how students have achieved the desired learning outcomes. They should be authentic and meaningful, providing students with opportunities to demonstrate their understanding and application of knowledge and skills in real-world contexts (J. Biggs &

Tang, 2011).

Constructive alignment results in a shift from teacher-centered to learner-centered education where students construct their knowledge (Schreurs & Dumbraveanu, 2014). This approach is especially common in higher education (J. Biggs, 2014). Various studies showed the effectiveness of constructive alignment and its positive impact on the students' engagement and motivation (Hailikari et al., 2022; Larkin & Richardson, 2012)

2.2.3 Self-Efficacy

The concept of self-efficacy was introduced by psychologist Bandura (1978) (also known for his social cognitive theory). Self-efficacy is related to self-confidence but focuses on the personal belief in the own capabilities to reach a desired learning outcome. The belief in one's capabilities does not have to match with the actual skills; various studies showed that there is indeed an overestimation of the own capabilities (Artino, 2012). Academic self-efficacy describes the learners' beliefs and attitudes regarding their academic capabilities. Learners with low self-efficacy tend to be afraid of tasks, avoid or postpone them and give up easily. (Hayat et al., 2020). This means that self-efficacy determines how much effort a person puts into a task and how much it is enjoyed.

Self-efficacy is an important concept in educational contexts for both students and teachers. The concept was also applied in the field of computer science education, especially in programming. Steinhorst et al. (2020) developed an instrument to evaluate self-efficacy in an introductory course to programming. The questionnaire involves 20 items and can be applied to various programming languages and paradigms. Hutchison-Green et al. (2008) conducted a qualitative study regarding self-efficacy among engineering students in their first semester. They found out that self-efficacy is influenced by students comparing the speed of their performance, their contribution to group work, their grades, and the amount of work they performed with their peers.

2.2.4 Engagement

The concept of engagement goes back to the definition of *student involvement* by Astin (1984). Astin defined student involvement as *"the amount of physical and psychological energy that the student devotes to the academic experiences."* Students are involved if they participate in extracurricular activities, interact with faculty members, or are immersed in academic work.

Kuh et al. (2007) define engagement as *"participation in educationally effective practices, both inside and outside the classroom, which leads to a range of measurable outcomes"*. Another definition is from Ben-Eliyahu et al. (2018), which define engagement as *"[...] the intensity of productive involvement with an activity"*. Student engagement is a multi-dimensional concept, however many of these concepts are named differently but mean the same thing (Deng et al., 2020). Halverson and Graham (2019) reviewed several models and definitions regarding student engagement. These models include different indicators of engagement with various subscales. The authors present a conceptual framework for engagement consisting of two indicators: cognitive and emotional engagement. This framework is especially developed in the context of blended learning environments. The most discussed dimensions of engagement consist of three components (Ben-Eliyahu et al., 2018; Joshi et al., 2022; Nazamud-din et al., 2020):

- **Affective** - Affective engagement refers to the emotional and attitudinal connection an individual has towards a particular task, topic, or learning experience. It involves feelings of interest, enjoyment, and motivation that drive one's active participation.
- **Behavioral** - Behavioral engagement relates to observable actions and behaviors individuals exhibit during a task or learning process. It contains active involvement, participation, and effort put forth to complete activities, tasks, or assignments.
- **Cognitive** - Cognitive engagement involves individuals' mental processes and intellectual efforts in a task or learning situation. It encompasses higher-order thinking skills, such as critical thinking, problem-solving, and knowledge application, indicating deep understanding and active mental involvement.

2.2.5 Motivation

Motivation plays a vital role in education, shaping students' engagement, learning outcomes, and overall academic success. When motivated, students demonstrate higher effort, persistence, and enjoyment in their learning endeavors. Deci and Ryan (1985) developed the concept of *self-determination theory* (SDT). Figure 2.3 shows the main components of the self-determination theory. According to the authors, competence, autonomy, and relatedness are the basic psychological needs of all humans. Competence is about feeling skilled and capable in what we do. It is the need to solve tasks and improve abilities. It is about facing challenges, gaining knowledge, and becoming proficient at things to feel confident in skills and abilities. Autonomy is about having the freedom to make own choices and decisions. It is the need to feel in control of actions rather than being forced by others. Relatedness refers to the need for individuals to experience meaningful connections, belongingness, and social interactions with others. It involves the desire to establish and maintain positive relationships, feel understood, supported, and connected to others, and have a sense of belonging to a social group or community.

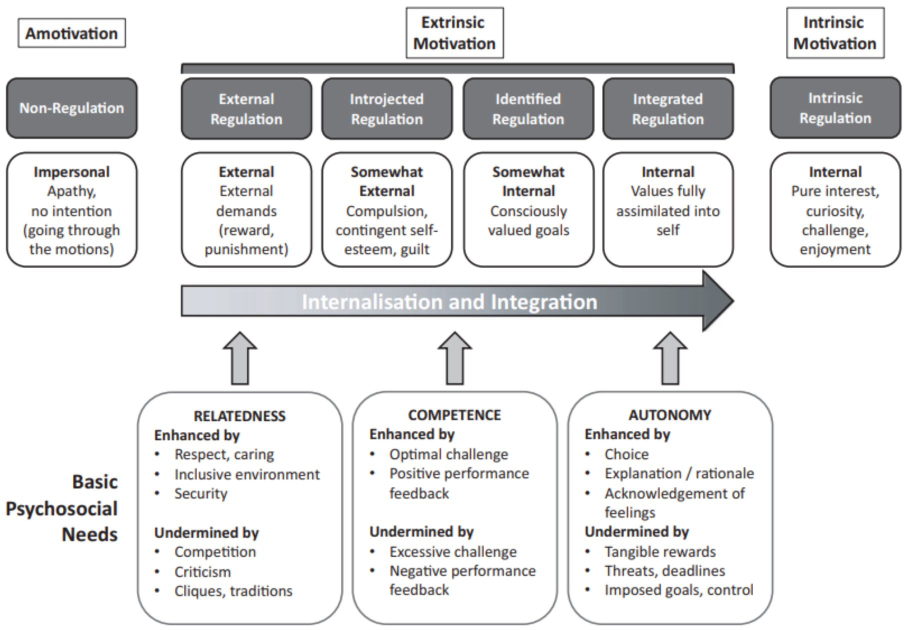

Figure 2.2: Overview of Deci and Ryan's self-determination theory (Cook & Artino, 2016).

Motivation exists in three dimensions: intrinsic, extrinsic, and amotivation. Intrinsic motivation, driven by internal factors such as curiosity, personal interest, and the desire for mastery, fosters a deep and lasting engagement with the subject matter. Extrinsic motivation, including rewards, recognition, and grades, can also serve as valuable incentives to push students' academic performance. Amotivation refers to lacking motivation or having a complete absence of motivation. It occurs when individuals perceive a task as irrelevant, or when they feel a lack of control or competence in the activity. In this state, individuals may show disinterest, passivity, and a lack of engagement or effort in pursuing the task or goal (Cook & Artino, 2016).

Educators can create supportive learning environments encouraging intrinsic motivation by providing meaningful and challenging tasks, promoting autonomy and choice, and fostering a sense of competence and belonging. For this reason, SDT is especially an appropriate framework for motivation in online learning environments

(K.-C. Chen & Jang, 2010).

2.3 Instructional Strategies in CS Education

Instructional strategies are approaches and techniques used by educators to teach specific learning content. According to Saskatchewan Education (1991), instructional strategies can be categorized, even though a clear distinction is not always possible.

- **Direct Instruction** - These strategies are teacher-directed and used when information should be provided. Methods of direct instruction are lectures, demonstrations, or explicit teaching.
- **Indirect Instruction** - Students construct their knowledge through problem-solving or inquiry, which means they are highly involved. The role of the teachers is to pre-teach or to navigate students in a certain direction by setting specific tasks. Project-based learning, problem-solving, or case studies are appropriate methods of indirect instruction.
- **Interactive Instruction** - Interactive instruction involve active collaboration between students and teachers and students. It emphasizes discussions, group activities, and hands-on activities.
- **Experiential Learning** - Students learn by working on real-life examples, experiments, or simulations. Experiential learning can be performed in the classroom but also outside of the classroom. This can be simulations, games, stories, models,s or model building.
- **Independent Study** - In this highly student-centered strategy, teachers provide content and guide the students. This fosters the individual development of the students and their abilities. Students can work on areas of interest, conduct surveys and work on their goals. These strategies compromise computer-assisted instruction or research projects.

Dengel and Gehrlein (2022) conducted a study with 16 teachers to investigate preferred teaching methods. They discovered that problem-based learning, project work, and programmed instructions are the most popular approaches.

2.3.1 Game-based Learning and Gamification

Game-based learning (GBL) is an educational approach where games are used to support teaching and learning. This means games are integral to the educational process (Perrotta et al., 2013). Game-based learning is, per see, not limited to digital

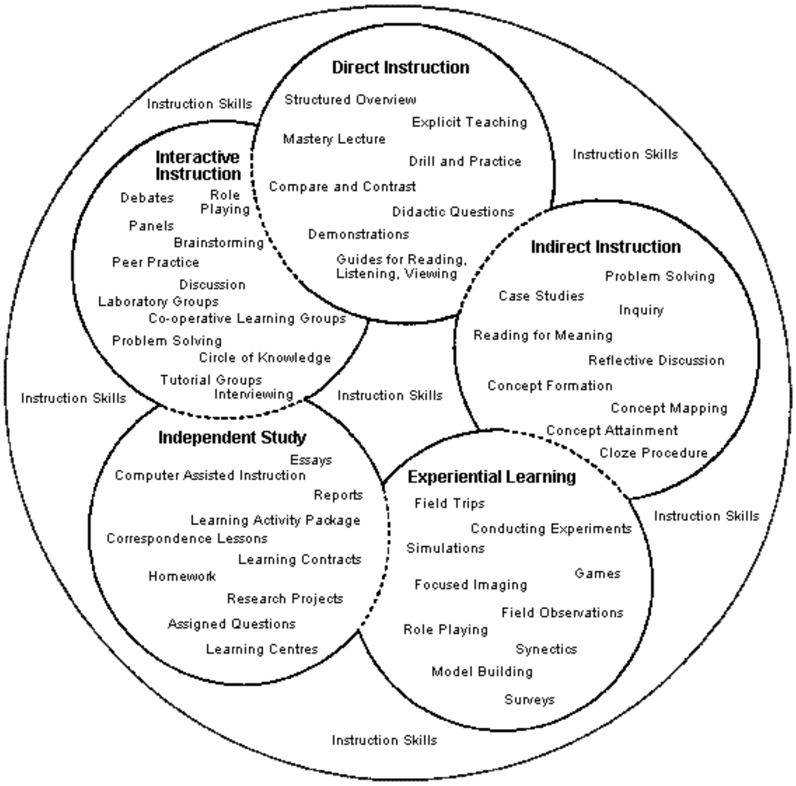

Figure 2.3: Instructional Strategies (Cook & Artino, 2016).

learning and also includes analogous games. However, it is mainly related to digital learning.

Games have the potential to engage learners and provide a learning environment. The main purpose of traditional video games is entertainment, however, *serious games* have the purpose of educating learners. These games mainly include specific learning objectives the game. Plass et al. (2015) developed a model describing the basic structure of games respective game-based learning environments. The model consists of three components: i) challenge, ii) feedback, iii) response.

According to Ke et al. (2015) the following facets of engagement are related to game-based learning:

- Affective Engagement
- Cognitive Engagement
- Content Engagement
- Gameplay Relevance
- Congregation of the four facets

In general, the idea of game-based learning is to integrate games into teaching and learning. However, another approach is to integrate not an entire game (such as a video game), but game elements. According to Deterding et al. (2011) gamification is the use of game elements within a (traditional) educational context. Game components and mechanics are added to a non-gaming situation to encourage engagement. Gamification elements can be (Toda et al., 2019):

- **Points**: Points are a common gamification element used to track progress and provide a sense of accomplishment. They can be awarded for completing tasks, achieving milestones, or demonstrating desired behaviors.
- **Badges**: Badges are visual representations of achievements or skills attained. They serve as a form of recognition and can motivate learners by providing a tangible representation of their accomplishments.
- **Leaderboards**: Leaderboards display rankings or scores of participants, creating a competitive environment. They foster engagement and encourage learners to strive for improvement by comparing their progress.
- **Levels**: Levels signify progression within a learning system. Learners start at a lower level and advance by completing challenges or accumulating points. Levels provide a sense of mastery and give learners a clear path to follow.
- **Quests**: Quests present learners with specific tasks or challenges. They provide a narrative structure and context, making the learning experience more engaging and immersive.
- **Feedback**: Feedback is an essential gamification element. It provides learners with information about their performance, progress, or areas for improvement. Effective feedback reinforces positive behaviors and guides learners towards desired outcomes.
- **Rewards**: Rewards can be tangible or intangible incentives given to learners for accomplishing specific goals. They can include unlocking new content, accessing additional features, or receiving virtual or physical rewards, such as digital items or certificates.
- **Avatars**: Avatars are digital representations of learners. They allow individuals to personalize their gaming experience and create a sense of identity within the learning environment.
- **Challenges**: Challenges present learners with tasks that require problem-solving or critical thinking. They promote active engagement and encourage learners to apply their knowledge and skills practically.
- **Progress Bars**: Progress bars visually depict learners' progress toward a goal. They provide a sense of direction, motivation, and accomplishment as learners can see how far they have come and how much further they must go.

- **Choice**: Choice is a gamification element that empowers learners by offering them options within the learning experience. It allows learners to make decisions that impact their journey, fostering a sense of autonomy and personalization.

Serious games and game-based learning use educational (video) games within a learning context. However, the idea of *gamification* is *"the use of game design elements in non-game contexts"* (Deterding et al., 2011). Abdul Rahman et al. (2018) argues that gamification positively impacts students' achievements, involvement, and motivation. Principles of gamification are found in different educational and non-educational contexts. Communities such as Stack Exchange use up and down votings and badges to engage users to ask and answer questions. Popular learning platforms such as Duolingo or Coursera are also applying gamification concepts. Elements such as levels, badges, or points keep the platform's users engaged and motivated (Nah et al., 2014).

2.3.2 ICT-based Learning

ICT (Information and Communication Technology) based learning refers to using technology, specifically digital tools and resources, to support and enhance learning. It involves integrating various information and communication technologies into educational practices to facilitate more effective and engaging learning experiences (Ratheeswari, 2018).

ICT-based learning encompasses various technologies, including computers, mobile devices, interactive whiteboards, educational software, online learning platforms, multimedia resources, and internet-based applications (Stoykova, 2015). These technologies can be utilized for various educational purposes, such as delivering instructional content, facilitating interactive and collaborative learning activities, providing virtual simulations and experiments, offering personalized feedback and assessment, and promoting independent research and exploration (Jena, 2015).

ICT-based learning provides many possibilities for education. It is already an indispensable component in nowadays K-12 and higher education. According to Toro and Joshi (2012), ICT-based learning has the following implications in higher education: i) student-centered learning, supporting knowledge construction, iii) anyplace learning, iv) anytime learning, v) information literacy. ICT-based learning provides students with a clear structure, materials, communication technologies, lecture notes, tutorials, teacher-student interaction and much more.

2.3.3 Peer Learning

Peer learning, also known as peer-to-peer learning or collaborative learning, refers to a learning approach where students actively engage with their peers to acquire knowledge, skills, and understanding. It involves students working together in groups or pairs to share ideas, discuss concepts, solve problems, and provide feedback to one another. In peer learning, students take on both the role of learners and teachers as they contribute to each other's learning through interaction and collaboration. The emphasis is on creating a supportive and inclusive learning environment where students can exchange knowledge, perspectives, and experiences (Brindley et al., 2009; Jackson & Bruegmann, 2009).

Peer learning can be implemented in various educational settings, including classrooms, online learning environments, and informal learning spaces. It complements traditional teaching methods by promoting active learning, student engagement, and the development of critical thinking and interpersonal skills (Rivadeneira & Inga, 2023).

There are various forms and types of peer learning:

- **Peer tutoring** - Peer tutoring and cooperating learning are the most commonly used forms of peer feedback. Peer tutoring can be either where peers have different levels of cognitive development or peers have closer levels with a focus on co-construction (Thurston et al., 2007).
- **Peer mentoring** - An experienced student, often in a higher grade or more advanced level, acts as a mentor to guide and support a less experienced student. Mentors provide advice and encouragement and share their personal experiences (Topping, 2005).
- **Peer assessment** - Peers assess their learning outcomes against each other. This means that they take on the role of the teacher and exchange qualitative feedback. Since school students tend to give average ratings for their peers, this form is more commonly used in higher education (Topping, 2005).

2.3.4 Project-based Learning

Project-based learning (PBL) is an instructional strategy where students are engaged in learning by working on a real-world project by solving a complex problem. Usually, students work collaboratively in teams to work on the project. Students have an active role in making decisions with defined responsibilities within the project. The benefits of project-based learning are that students can apply their theoretical knowledge within a practical context. Their problem-solving skills will be increased as well. However, it has additional benefits related to soft skills such as improvement of teamwork, responsibility, or responsibility (Indrawan & Jalinus, 2019). Project-based learning is frequently used in higher education, especially in engineering education. Shpeizer (2019) suggests using technologies such as learning management systems, mobile technology, and collaborative tools for effective project-based learning. For PBL to succeed, several other levels must be considered. The importance of PBL must be emphasized by the organization. Further, the institution must acknowledge the additional time and effort for educators and students. Problem-based learning is different from traditional learning approaches. Therefore, it requires certain training for faculty and students and support structures.

Larmer and Mergendoller (2023) developed the *Gold Standard PBL* (GSPBL), which are two models for project-based learning. One model is dedicated to seven project design elements; the second includes seven project-based teaching practices. Figure 2.4 shows all seven elements of project-based learning. The center of this framework are the learning goals where students should gain in-depth content knowledge and acquire other skills (such as soft skills) as well. These learning objectives are also the base for the teaching practices from Figure 2.5. The seven elements contain all aspects that are relevant for the teachers to design successful PBL experiences. This *Gold Standard PBL* has been applied and evaluated in different educational contexts. Sayuti et al. (2020) designed a PBL setting in a school within an English-speaking class. They proposed several activities over eight weeks and linked these activities to elements of the GSPBL. Another approach was designed for a first-year engineering course linking models and activities to PBL elements (M., 2020).

2.3.5 Blended Learning

According to Garrison and Kanuka (2004), blended learning is *"blended learning is the thoughtful integration of classroom face-to-face learning experiences with online learning experiences"*. This means blended learning consists of two components: a face-to-face setting and online elements. It has great potential for both learners and teachers since it provides a flexible and adaptive learning environment where students can engage with course materials, collaborate with peers, and receive personalized instruction. By seamlessly integrating online and offline components, blended learning aims to optimize the educational experience, improve the students' engagement, and foster critical thinking and problem-solving skills (Gan et al., 2015; Owston et al., 2013).

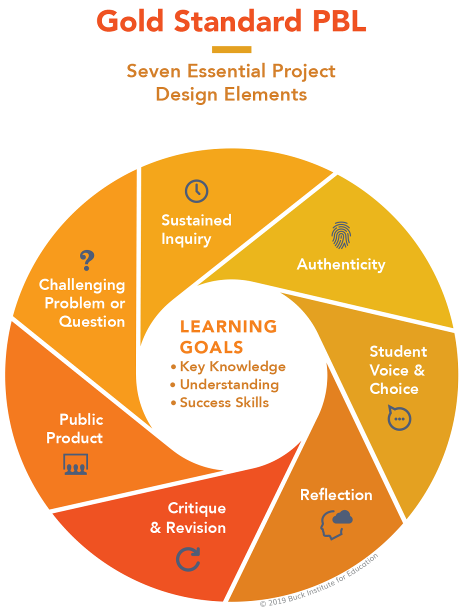

Figure 2.4: PBL Gold Standard Design Elements

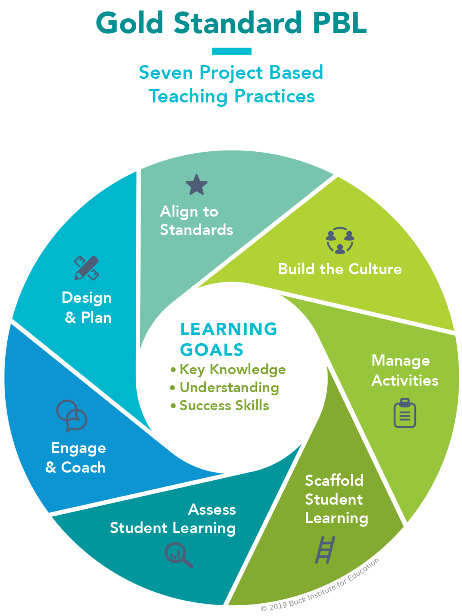

Figure 2.5: PBL Gold Standard Teaching Practices

Blended learning offers a multitude of benefits that contribute to improved educational outcomes. Firstly, it enhances accessibility and flexibility by enabling students to access course materials and engage in learning activities at their own pace and convenience. This flexibility accommodates the diverse needs and learning styles of students, promoting inclusive education (Draffan & Rainger, 2006). Secondly, blended learning facilitates personalized learning experiences, allowing instructors to individualize content, pace, and instructional strategies to individual student needs. The online component provides opportunities for self-paced learning and individualized feedback, while face-to-face interactions enable social and collaborative learning experiences. Lastly, blended learning fosters the development of digital literacy and technological skills, preparing students for the demands of the digital age (Watson, 2008).

Despite the numerous benefits, the implementation of blended learning has certain challenges. Faculty and staff may require training and support to effectively integrate technology into their teaching practices, ensuring that the online and offline components align seamlessly. Moreover, technological infrastructure and access to reliable internet connectivity can pose obstacles, particularly in resource-constrained areas. As blended learning continues to evolve, it is essential to address these challenges and identify best practices to maximize its potential (Garrison & Kanuka, 2004).

2.3.6 Unplugged Learning

Computer Science Unplugged (CS Unplugged) is a widely used pedagogical concept that was introduced by Tim Bell, Mike Fellows, and Ian Witten in 1999 (Bell & Witten, 1998). It is a collection of hands-on learning activities and materials that should engage students in computational thinking and computer science. The primary goal of CS Unplugged is to make computer science accessible to a wide range of learners, including those who may not have access to computers or technology resources.

There is a great number of unplugged activities aiming to teach abstract concepts of computer science in a playful way. CS Unplugged activities often involve games, puzzles, and simulations that illustrate key computer science ideas. Nishida et al. (2009) define the following main characteristics of unplugged activities:

- No computers: During an unplugged activity no computers are used.
- Games: The activities are usually playful activities to engage students in learning something new.

- Kinaesthetic: Real, physical objects are a key element of unplugged activities.
- Student directed: The activities are highly student-centered, which means that they are often group activities that promote collaboration.
- Easy implementation: The required material is easy to get and prepare.
- Growing body of ideas: Educators can easily participate and contribute to the project.
- Sense of story: Engaging stories and narratives can be used to motivate young learners.

Unplugged approaches are widely used in recommendations such as ACM K-12 or CSTA Standards for K-12 (Tucker, 2003b). However, these approaches are also applied in higher education such as Harvard's well-known CS50 course where unplugged activities are used as a visualization for sorting algorithms. Various studies investigated the effect of CS Unplugged. Literature shows mixed evidence, however, Bell and Vahrenhold (2018) state that cooperation of programming and unplugged activities show the best results.

2.4 Online Learning Platforms

Online learning platforms have revolutionized education delivery over the last few years. They offer diverse and accessible learning experiences to individuals worldwide. There are an immense number of different learning platforms that also have different characteristics. One distinction is between synchronous and asynchronous communication within platforms. Synchronous means that the communication occurs simultaneously, while asynchronous means that the communication occurs at different times (Kozaris, 2010). Another classification of platforms is between self-paced, highly learner-centered, and instruction-led, highly teacher-centered learning.

Even though different characteristics define learning platforms, there is no commonly used taxonomy. Podmurnyi (n.d.) are listing eight types of learning platforms:

- **Learning Management Systems (LMS)** - A LMS is a software application that enables online administration, delivery, tracking, and management of educational courses and training programs.
- **Social Learning Platforms** - A social learning platform combines the features of an LMS with social networking elements, fostering interaction and knowledge sharing among learners, resulting in higher completion rates and

increased satisfaction due to peer feedback and group dynamics.

- **Online Training Platforms for Business** - Business-focused e-learning platforms prioritize assessing knowledge and skills for job prospects and employees. This enables organizations to ensure appropriate skill levels while offering customized programs tailored to individual learners' needs and pace.
- **Massive Open Online Courses** - MOOCs are e-learning platforms offering courses with free and paid options, allowing users to earn certificates upon completion and offering flexibility to accommodate learners' schedules.
- **Destination Site for Learning** - Online course platforms such as Coursera and Udemy offer a diverse range of classes from various providers, utilizing different payment and access models while also providing course authoring tools for instructors and offering learners access to course content in multiple formats and opportunities for engagement through forums and discussion groups.
- **Learning Experience Systems (LXS)** - LXS is a personalized, immersive education system that addresses the limitations of traditional LMS by providing user-focused, intuitive learning, interactive experiences, and a range of formats, authoring tools, and social learning features.
- **Custom-Built Online Education Platform** - Custom e-learning platforms are tailored to the specific requirements of businesses or educational institutions, allowing for full control and customization of the learning experience. However, the development cost may be higher than pre-built online course authoring tools.
- **Learning Apps** - The growing popularity of microlearning and interactive educational games in e-learning app development reflects the need for convenient, time-efficient learning that enhances knowledge retention and engagement.

This large number of different learning platforms would go way beyond the scope of this thesis. For this reason, those platforms will be covered in the following subsections used within this dissertation.

2.4.1 MOOCS

During the last years *Massive Open Online Courses* (MOOCS) had a significant impact on online education. MOOC as a term was first used by Stephen Downes and George Siemens in 2008. They intended to connect a wide variety of learners using online tools. Later in 2011, videos were created at Stanford University that were openly available. The course *Introduction to Artificial Intelligence* had over 160,000 students enrolled over the entire globe. Later on, platforms such as Udacity and edX were formed with the goal to provide open online courses (Baturay, 2015).

Two terms are commonly used related to MOOCs: cMOOC and xMOOC. cMOOCs (Connectivist MOOCs) focus on the building of networks of learners and fostering collaborative learning experiences. They emphasize learners' autonomy and the construction of knowledge using social interactions. These MOOCs are characterized by personal learning paths which support self-directed learning, communities, and interaction between learners where they discuss, share resources, or collaborate on projects. xMOOCs (Extended MOOCs) are traditional or structured MOOCs that are more instructor-centric and usually offered by educational institutions. They often deliver pre-recorded video lectures related to a specific topic by expert instructors. They are structured using a pre-defined curriculum with modules and units. Learners work on quizzes, assignments, and exams during the course. They highly rely on assessment to evaluate the learners' understanding and progress and offer certificates at the end of the course (Fidalgo-Blanco et al., 2016).

MOOCs are highly used in higher and further education since they can be easily scaled and accessible. However, there are also challenges for MOOC providers. Less than 10 % of students complete a course. For this reason, there are many approaches to engage students to complete MOOCs and motivate them (Cvetković, 2016, p. 125). Onah et al. (2014) analyzed MOOC attrition and identified several reasons for MOOC dropouts:

- **No real intention to complete** - Many users enroll in MOOCs out of curiosity or to learn more about the MOOC format rather than the subject itself. This broad range of background, intention, and participation is seen as a by-product of the open-access nature of the courses, and it is suggested that statistics on course completion should not include users who do not intend to participate fully, as it is challenging to measure and assess their learning outcomes accurately.
- **Lack of time** - Even motivated students may struggle to complete a course due to time constraints and high workload, regardless of their initial intentions. The standardized format of MOOCs fails to accommodate individual learners' diverse needs and varying learning speeds, resulting in some students requiring more or less time to grasp the learning materials.
- **Course difficulty and lack of support** - The difficulty level of a course and the lack of required background knowledge can hinder students' progress. Insufficient mathematical skills and a lack of understanding of complex topics often lead students to discontinue their participation. Students frequently

express dissatisfaction with the lack of peer support and instructor guidance when faced with challenging subjects.

- **Lack of digital skills or learning skills** - To succeed in online learning, users must study independently and use different technologies and formats. Even if someone is familiar with regular technology, it can still be hard to learn new systems quickly.
- **Bad experiences** - MOOC participants have mentioned several barriers to continued participation, such as inappropriate behavior of peers, disorganized forums, study group attrition, poor learning materials, and technical problems within the platform.
- **Starting late** - Students who join a course late find it difficult to catch up, resulting in lower outcomes for them. It is not just about catching up with the learning materials; latecomers struggle to fit into existing support groups and learning networks, and they often have trouble navigating the forums when discussions have already progressed.
- **Peer review** - Courses that rely on peer grading often have lower completion rates. Peer grading requires more effort from students; some may be unhappy with the concept or lack proper training. Negative experiences, such as unhelpful feedback and plagiarism in peer work, can also demotivate participants.

Wilkowski et al. (2014) observed four groups of participants within MOOCs: i) Students that complete all parts that are necessary within a course are called *Completers*. ii) Students that partially want to learn new things out of curiosity or due to educational needs are called *Casual learners*. iii) Students that want to have access to the course and want to get some information about it are called *Observers*. iv) Finally, students that are just registered for a course but never show up are called *No-shows*.

Well-established MOOC platforms are:

- **Coursera** - Coursera[2] offers many courses from top universities and institutions worldwide. It covers diverse subjects such as computer science, business, arts, and humanities. Coursera also provides specialization programs and professional certificates.
- **edX** - Founded by Harvard University and MIT, edX[3] offers global MOOCs from prestigious universities and institutions. It offers courses across various disciplines, including science, engineering, humanities, and social sciences.

[2] https://www.coursera.org/
[3] https://www.edx.org/

- **Udacity** - Udacity[4] focuses on courses in technology-related fields, including programming, data science, artificial intelligence, and self-driving cars. It offers nano degree programs designed to enhance specific skills for career advancement.
- **FutureLearn** - FutureLearn[5] provides various courses from universities and cultural institutions. It covers healthcare, social sciences, languages, and creative arts. FutureLearn also offers programs with academic credit.
- **LinkedIn Learning** - Formerly known as Lynda.com, LinkedIn Learning[6] offers online courses and video tutorials covering various topics, including business, technology, creative skills, and personal development. It is particularly useful for professional development and acquiring specific skills.

2.4.2 Learning Management Systems

Learning Management Systems (LMS) have become integral to modern education, offering comprehensive platforms for managing and delivering educational content. These systems provide a wide range of features and tools that facilitate the organization, administration, and tracking of learning activities. According to Turnbull et al. (2020), learning management systems have the following features:

- **Course Management** - Course management features in an LMS involve delivering course material to students, including content management, class scheduling, content audits, and the ability for users to contribute to content creation in their personal space.
- **Assessment** - Activities that should be assessed must be collected and analyzed, such as assignments, tests, projects, etc. The system should further provide learners with information about their progress.
- **Tracking progress** - The users' engagement and behavior must be tracked to minimize the attrition rate. The reporting should include log-in, interaction with activities, etc.
- **Gradebook** - The system should provide all relevant information regarding grading and the instructor's feedback.
- **Communication tools** - Communication can occur synchronously using videoconferencing, chats, or asynchronously using discussion forums or wikis.
- **Social connectivity** - An LMS should encourage building a community of learners and enable features such as forums or chats.

[4] https://www.udacity.com/
[5] https://www.futurelearn.com/
[6] https://www.linkedin.com/learning/

- **Security and privacy** - The system should be secure and sensitive to the learners' data.
- **Ubiquitous access** - Since learners use different platforms and devices to interact with the LMS all of them should be supported.

Learning management systems also provide opportunities for adaptive learning. In this way, tasks and materials can be provided individually for learners. Analysis tools also make it possible to perform data-driven evaluations and thus support learners by alerting them to problems at an early stage. Additionally, gamification elements such as rewards, certificates, or badges can engage learners in MOOC active participation within the LMS (Thakur et al., 2014).

Several learning management systems, such as Moodle, Canvas, Google Classroom, and Blackboard, are used. Moodle and Canvas are two systems that are well-established in schools and universities. Both platforms are open-source and can therefore be provided by the institutions. They have common features including assessment, grading, different instructional methods, and skill tracking (Mpungose & Khoza, 2022).

Xinogalos et al. (2020) observed which factors positively correlate with satisfaction within a programming course. An important factor is the content and quality of a course. The teacher's attitude towards TEL as well as diverse assessment options. Students further engage in discussion forums and active involvement. The authors mention that, especially in large-scale teaching aspects such as managing lectures, announcements, submissions, learning content, student support, or communication, must be considered. Therefore, LMS provides comprehensive and satisfying possibilities.

2.4.3 Streaming Platforms

Video streaming platforms have a high value in the area of education. According to X. Chen et al. (2021), there are two types of live video streaming platforms, videoconferencing and live-streaming platforms. They define videoconferencing as *"synchronous video and audio communication cress geographic sites"* while live-streaming is *"an emerging practice of broadcasting video of oneself in real-time to an online audience"* (X. Chen et al., 2021). The video-sharing platform YouTube has shifted from pure entertainment to education. Smith et al. (2018) conducted a study with 4,594 Americans with the findings that 51 % of all participants use YouTube for learning. One of the advantages of YouTube as an educational platform is that it provides valuable resources from experts. The videos can be watched at every time and from everywhere on different devices. Another benefit is the community that builds among channels, where learners and content creators can exchange. YouTube allows self-paced and self-directed learning (Srinivasacharlu, 2020).

Another platform that initially had a different target audience is Twitch. Twitch is mainly known for the live streaming of video games, including lively conversations between streamers and the audience. Increasingly, educational content is also available on Twitch, including live lectures, Q & A sessions, or interactive workshops in various fields. Pirker et al. (2021) conducted a study within two university courses in a CS degree. They discovered four main issues and stated recommendations for lecturers that want to use Twitch in teaching. The main challenges are spam and bots, a high level of interaction and engagement, technical issues, and a clear stream structure within the lecture context. The authors emphasize especially the interaction between students and the streamers (lecturers) as a promising possibility to ask questions and include formative assessment elements.

2.4.4 Serious Games

The general goal of games is to entertain their players. In contrast, serious games are designed specifically for educational settings. Serious games have similar game mechanics to regular video games. Kickmeier-Rust (2009) proposed a hyper-cube taxonomy for serious games, including i) purpose (enjoyment or learning), ii) reality (imitating real contexts or high abstraction), iii) social involvement (single-player or massive multi-player), and iv) activity (active or passive game types). Another taxonomy is from Laamarti et al. (2014), which identified five areas: i) application area, ii) activity, iii) modality, iv) interaction style, and v) environment.

There is a large number of domains where serious games can be used. According to Stapleton (2004), serious games can are applied in seven markets: in K-12 edutainment, higher education, health care, corporate, military, non-government, and others Wattanasoontorn et al. (2013) show how serious games are used in the health sector. In a meta-analysis, they classified serious games into three subjects player, health, and game application. The subject of players divides into games for patients and non-patients but also games for professionals and non-professionals. Health is related to the stage of disease for health monitoring and detection, but also treatment and therapy. Finally, the game application can be the game's purpose (entertainment, health, acquiring skills) and functionality (game engine, platform, game interface, etc.)

Since the main purpose of serious games is to educate, assessment is an important area that has to be considered. This further raises the question of how effective (regarding learning performance) serious games are and how they can be integrated into an educational setting. According to Bellotti et al. (2013), serious games (game-based learning) are effective in motivating students and to achieve learning goals at low taxonomy levels (Bloom's taxonomy). They suggest meaningful in-game assessment as an important way for efficient learning experiences even on higher taxonomy levels.

2.5 Educational Assessment

Educational assessment encompasses a range of processes to gather information about students' knowledge, skills, and abilities. It uses qualitative and quantitative data to inform instructional decisions and support student learning. Assessment can be formative, providing ongoing feedback and guidance to students and educators throughout the learning process. This type of assessment focuses on identifying strengths and weaknesses, offering timely interventions, and promoting self-regulated learning. Formative assessment often involves qualitative data, such as observations, interviews, and open-ended responses, allowing a deeper understanding of student progress and learning experiences. On the other hand, summative assessment is typically conducted at the end of a learning period to evaluate student achievement and determine their overall level of mastery. It often utilizes quantitative data, such as standardized tests or numerical scores, to measure performance against predetermined criteria. Evaluation encompasses the systematic analysis and interpretation of assessment data to determine the effectiveness of educational programs, policies, or interventions. It involves using qualitative and quantitative data to make informed conclusions and decisions. Learning analytics, a growing field, leverages data analysis techniques to gain insights from educational data, supporting personalized learning, instructional improvement, and evidence-based decision-making. It involves systematically collecting, analyzing, and interpreting data generated during educational activities. Combining qualitative and quantitative data, formative and summative assessment, feedback and guidance, evaluation, and learning analytics contribute to a comprehensive approach to educational assessment, facilitating meaningful insights into student learning and promoting educational improvement.

2.5.1 Learning Analytics

Learning analytics is defined as *"learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs."* (Long & Siemens, 2011). Educational Data Mining (EDM) is related to Learning Analytics (LA). Both fields share the goal of leveraging data for educational improvement. They analyze educational data to enhance teaching and learning. While EDM applies data mining techniques to extract insights from educational data, LA encloses a broader range of approaches, including data mining, statistics, machine learning, and visualization, to measure, analyze, and optimize learning and the learning environment (Du et al., 2021).

Approaches to learning analytics can also be applied in game-based environments. Meaningful data can be gathered using data mining models to understand how learners interact with a game and get information on their learning process. Interesting educational conclusions can be drawn by combining data mining models with visualization. Therefore game learning analytics (GLA) are a way to improve the learners' assessment (Alonso-Fernandez et al., 2022).

Learning analytics can further track the students' attitudes and engagement using learning platforms. In traditional settings, qualitative data such as interviews, self-efficacy scales, or surveys provide subjective information about the students. Interaction data (time-on-task, reaction time, guide use, number of videos watched, page jumping, number of completed assignments, etc.) can deliver an additional layer giving objective data. This can be used for dropout prediction or early warning to support the students (Caspari-Sadeghi, 2022).

2.5.2 Evaluation

Evaluation is a term that is used in various subjects with different meanings. Wanzer (2021) systematically analyzed different definitions of evaluation. The definition of the American Evaluation Association has commonly used: *"Evaluation is the systematic process to determine merit, worth, value, or significance."* The definition of American Evaluation Association is understood in a broader sense, a narrower definition of the field of education given M. Miller et al. (2012): *"Evaluation is a systematic process of collecting, analysing and interpreting in formation to determine the extent to which pupil's are achievement instructional objectives."*

According to Gafoor (2013) evaluation is an ongoing process consisting of three phases:

1. Identifying and defining the intended outcomes,
2. constructing or selecting tests and other evaluation tools relevant to the specified outcomes, and
3. using the evaluation results to improve learning and teaching.

2.5.3 Assessment

In the area of education, assessment is used to determine a person's learning process by evaluating, measuring, and documenting it (Maclellan, 2001). Walvoord and Anderson (1998, p. 3) define assessment as a *"systematic collection of information about student learning, or programs of student learning, for the purpose of improving that learning."* The authors emphasize the aspect that learners should improve themselves. The authors name three components related to assessment: i) articulate the goals for student learning, ii) gather information about how well students are achieving the goals, and iii) use the information for improvement.

Assessment can take different forms, such as tests, exams, projects, portfolios, observations, interviews, or self-assessments. It can assess various dimensions, including cognitive abilities, content knowledge, problem-solving skills, creativity, critical thinking, and communication skills.

Assessment is also highly used in digital learning contexts. The goal of assessment is not just to examine the desired learning outcome; it should rather give educators and learners information about the current learning process. According to Arnold et al. (2018, p. 313) there are five types of digital assessment:

1. **Advisory** - This type of assessment takes place before an intervention takes place and should give orientation. Examples are online self-assessments or aptitude tests.
2. **Diagnostic** - This assessment form happens at the beginning of an educational intervention. It aims to classify the learners or to provide admission. Examples are admission tests, selection tests, e-portfolios, or placement tests.
3. **Formative** - Formative assessment is an ongoing process during the learning process. t involves gathering information and providing feedback to students to guide their learning and improve their understanding (Black et al., 2003). The primary purpose of formative assessment is to monitor student progress, identify areas of strengths and weaknesses, and adjust instruction accordingly. It helps teachers and students identify gaps in understanding, set goals, and

make instructional decisions to support learning. Examples are audience response tests, code analysis, quizzes, e-portfolios, or intermediate tests.

4. **Summative** - Summative assessment is conducted at the end of a unit, course, or instructional period to evaluate student learning outcomes and assign grades or determine overall proficiency (Black et al., 2003). Summative assessments are typically more formal and standardized. The primary purpose of summative assessment is to make judgments about student achievement and measure overall learning outcomes. Examples are pre-and post-tests, online exams, oral exams, or e-portfolios.
5. **Quality-assuring** - This assessment form occurs after an educational intervention. The goal is to improve the intervention for the future. This includes plagiarism checks, teacher evaluations, and progress tests.

2.5.4 Feedback and Guidance

Feedback and guidance refer to providing learners with information, support, and direction to enhance their understanding, performance, and overall learning experience. Feedback involves providing specific and constructive information about a learner's performance, progress, or understanding, highlighting strengths and areas for improvement. Guidance is more forward-looking and aims to support learners' ongoing learning journey. An immense amount of research literature has already addressed feedback and its impact. They have concluded that there is a strong effect between feedback and student achievements (Wisniewski et al., 2020). Feedback should refer to the knowledge of the result (how successfully something was performed) and the knowledge of performance (Chevalier et al., 2022).

In their meta-analysis Wisniewski et al. (2020) analyzed 435 empirical research studies. The authors differentiated between three types of feedback: i) reinforcement/punishment, ii) corrective feedback, and iii) high-information feedback. These feedback forms are transmitted orally, written, and video-, audio, or computer-assisted. The more info included in the feedback, the more effective it is. This means that reinforcement/punishment feedback or corrective feedback has lower effects.

2.6 Related Work

The previous sections aimed to overview related theories, models, and concepts. These foundations are the base of several engaging learning technologies and educational concepts. In the following section, two related areas are illustrated in detail.

Technologies and concepts are discussed concerning the theoretical foundation that is given in Sections 2.1 - 2.5. This dissertation covers several secondary and higher education approaches, focusing on both digital competencies and programming education. For this reason, the following three sections are presented to exemplify the treatment of the topic.

Engaging students in different topics of computer science is not new in research. Various studies implemented learning technologies and educational scenarios to increase the students' motivation aiming for better academic achievement. A common approach for engaging students is by using elements of gamification. In a case study by Ibáñez et al. (2014) the authors evaluated the effectiveness and engagement of gamification within a C programming course in higher education using the *Q-Learning-G platform*. This platorm includes elements such as levels, leaderboards, or badges. Their study showed that the gamification elements (especially badges) motivated the students to work beyond the required tasks. However, the authors also discuss whether rewards reduce or increase the students' motivation.

Pirker et al. (2014) introduced a model called *Motivational Active Learning* (MAL) aiming to engage students in computer science. MAL was evaluated within a higher education course on Information Search and Retrieval. It emphasizes on collaborative learning, where students work in small subgroups on tasks. Students receive immediate feedback on small tasks and quizzes to have a formative assessment on their performance. Gamification elements such as badges and leaderboards should further motivate students. MAL provides an environment in which errors are allowed, and students can retake activities.

2.6.1 Game-based Learning in Programming Education in School

Game-based learning in programming education has emerged as a captivating and effective approach to engage students in the learning process. This method offers an interactive and immersive experience that appeals to students' natural affinity for technology and gaming by incorporating game mechanics and simulations. Within the game-based learning environment, students actively solve programming challenges and apply their knowledge in practical contexts, fostering critical thinking and problem-solving skills. Elements such as levels, quests, rewards, and leaderboards further enhance student motivation and create a sense of achievement and progression. Collaborative features enable students to work together, promoting teamwork, communication, and interpersonal skills. By providing personalized learning experiences and immediate feedback on student performance, game-based

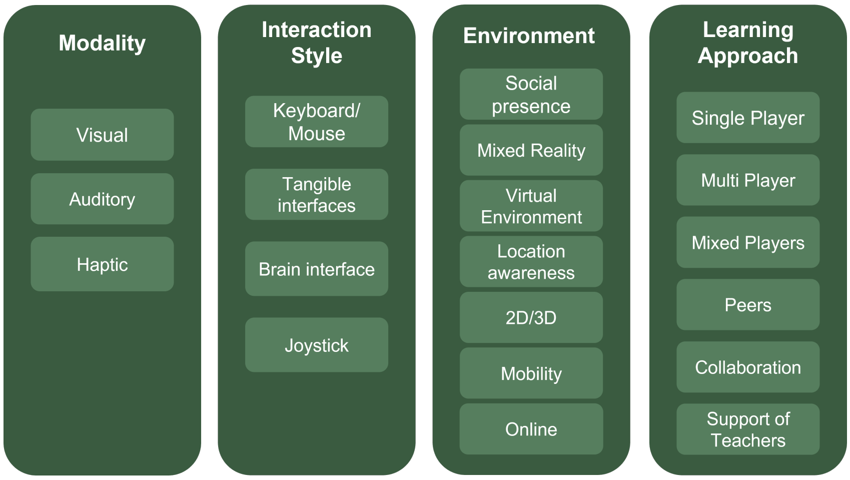

Figure 2.6: Classification of Educational Coding Games (Combéfis et al., 2016a).

learning in programming education enhances student engagement and supports the development of programming skills applicable in real-world scenarios.

According to Combéfis et al. (2016a), there are three main categories of game platforms related to programming education:

- Learn to Code
- Learn Algorithmic Thinking
- Learn to Create Games

Further Combéfis et al. (2016a) proposes a classification for educational programming games consisting of four categories: modality, interaction style, environment, and learning approach. Figure 2.6 shows the classification, including their elements.

However, their evaluation is essential to using serious games in education. Shahid et al. (2019) analyzed evaluation methods for game-based approaches in a meta-analysis of serious games for programming education. They argue that there is a lack of a proper framework for the evaluation of serious games. They found out that gameplay statistics and observations are usually missing within evaluations. The most frequently used evaluation methods are experimental designs (pre-and post-tests), formal interviews, questionnaires, feedback, and assessment tools.

Learn to code

Lindberg et al. (2019) conducted a survey where they investigated seven national CS curricula regarding programming education. They further reviewed serious games in the field of programming education and analyzed the covered concepts. They chose 29 games for different grades and analyzed the genres and the covered programming topics. Most of the curricula' respective games follow a block-based approach to introduce students to programming. They observed that there are games for beginners, intermediate, and advanced programming skills. The authors argue that the observed serious games support different learning theories, such as motivation and engagement. However, the authors identified a lack of social learning possibilities in the existing tools.

Combéfis et al. (2016a) categorizes serious games in programming education into four possible activities. Serious games such as RoboBUG (Miljanovic & Bradbury, 2017) are teaching students how to (i) fix bugs. This tool guides students on how to debug actual program code. Students learn how to trace code, use print statements to identify bugs, but also how to use breakpoints to analyze a program during runtime. Another category is games where players have to (ii) recover parts of the code. The serious game *Code Saga* consists of different components (Tacouri & Nagowah, 2021). One component is Code Blocks, a game where players have to order a list of given code blocks to execute the code correctly. The (iii) code writing category lets users write code from scratch. Code Hunt is a serious game where players can write program code in both the Java or C# programming language (Tillmann et al., 2014). Finally, Codemonkey is a game where players (iv) program an agent. Using CoffeeScript or Python, the players can program an avatar (a monkey) to reach certain goals.

Learn Algorithmic Thinking

Games within this category aim to develop and improve the students' skills in algorithmic thinking. Kazimoglu et al. (2012) are introducing the game *Program your robot* where students acquire skills in computational thinking. They are programming a robot using basic block commands. The game provides a limited instruction set which is represented by instruction blocks such as move commands or commands to call procedures.

Learn to Create Games

There are several secondary and higher education approaches for learning programming by creating video games. Visual programming environments such as Scratch provide accessible platforms to develop games easily. Scratch uses a block-based approach where code blocks with different functionalities are used to composite visual programs. Seralidou and Douligeris (2021) are designing a course for secondary school students with six hours to learn the fundamentals of object-oriented programming. Within a study with 52 participants, including questionnaires and pre-and post-tests, the authors could observe that the students' skills improved and their attitude towards learning to program increased as well.

2.6.2 Large-scale Teaching of Programming in Higher Education

A teaching and learning context that is especially interesting in terms of engagement is large-scale courses in higher education. Due to the higher number of students, course formats must be scalable, resulting in tension between automation and individualization. A common approach in such settings is blended learning strategies. Medeiros et al. (2019) conducted a systematic literature review on introductory programming courses, including 100 papers. Within this, they analyzed which previous skills are relevant for success in programming, students' difficulties, and teachers' challenges. They reported several categories where literature encountered difficulties for students when learning to program. One of the significant problems is related to the students' motivation and engagement in programming introductory courses. The results show that the students' engagement, motivation, and positive learning process correlate. However, motivating students and keeping them engaged is also perceived as a problem for educators.

Marasco et al. (2017) implemented a flipped classroom in an introductory programming course in higher education. They successfully run this course, including mainly instructor-led videos, quiz questions with weekly deadlines to keep students motivated, and a final project. Based on their experience, they revised the course and added weekly in-class tutorials to practice and ask questions. Maher et al. (2015) developed four different flipped classroom courses for computer science education at their university in subjects such as human-computer interaction or introduction to media computation. They emphasize collaborative approaches such as pair programming as essential in programming courses. Elmaleh and Shankararaman (2017) compared a traditional course with a flipped classroom course for a programming course. The students showed better grades in the flipped classroom setting, but the students also had to spend more time on the flipped model due to preparation. The findings show that this additional effort did not impact the students' motivation.

Another commonly used approach within large-scale teaching is Massive Open Online Courses (MOOCs). Alario-Hoyos et al. (2017) analyzed a MOOC with 6,335 learners within an edX course on introduction to programming. This course includes video-based lectures and interactive activities. Over a period of 5 weeks, the fundamental concepts of the Java programming language were taught. The authors investigated that time management causes attrition. They suggest early estimation of weekly workload and individual workload for each assignment to support the students. Alario-Hoyos et al. (2016) suggest a high level of interactive activities within MOOCs, such as quizzes or coding activities. Tools such as Codeboard or Blockly can be easily integrated into MOOCs and provide additional value.

2.7 Summary and Discussion

This chapter explains the concepts that are the base for engaging learning experiences in computer science education. Figure 2.7 shows an adapted model of Figure 1.1 highlighting all areas that influence engaging learning. When discussing engaging learning technologies and educational scenarios, well-established teaching and learning models are essential. Research showed that self-efficacy, engagement, and motivation are three concepts that are frequently analyzed when considering the effectiveness of an intervention. The students' engagement and motivation are highly related to their performance and academic achievements. Enduring learning experiences require focusing on higher taxonomy levels such as anaylze, evaluate and create. The link between learning objectives, activities, and assessment is called constructive alignment. Connecting learning activities and instructional strategies are vital to reaching these high levels. When strategies such as project-based learning or peer learning are used well, these higher levels can be addressed. In the literature, game-based learning is considered suitable for motivating and engaging students when introducing a new topic. Even though there is no common taxonomy for online learning platforms, there is a consensus on the most frequently used platforms. Features that are relevant in almost all proposed platforms are communication and collaboration among peers and teachers, highly available learning content, but also instruments to keep track of the learning process. Monitoring the learning process is closely related to learning analytics, which is a key component of educational assessment. Overall, an assessment of the learning process (both summative and formative) is needed to support the learners and provide them with a meaningful learning experience.

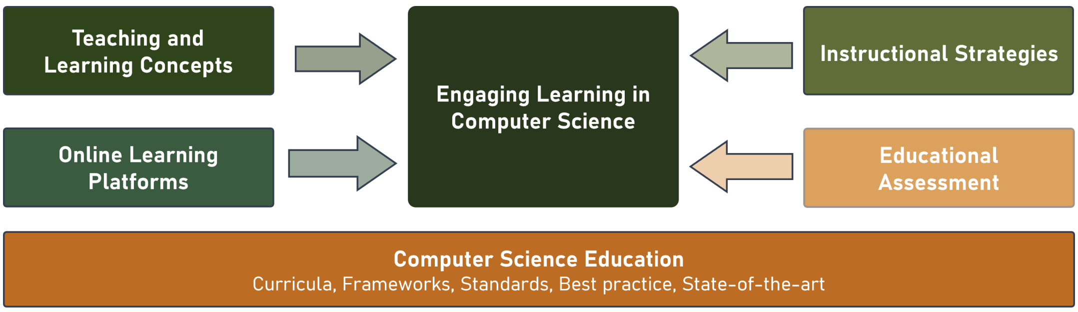

Figure 2.7: This figure shows a simplified version of the FELCS model from Figure 1.1.

The concepts and models that are introduced and presented within this chapter are the foundation for many other approaches as well. Section 2.6 shows that instructional strategies such as collaborative and game-based learning and online platforms such as MOOCs or LMS are used in different learning contexts. These concepts and models are the base for all studies and interventions conducted within this thesis's next chapters.

3 Publications

"Science is not only a disciple of reason but also one of romance and passion."

Stephen Hawking

A fundamental question framing this thesis's entire research is: *"How can computer science education be designed to be both engaging and enduring?"* Engaging education is influenced by several factors that have to be carefully considered and implemented. To actively stimulate students' interest, capture their attention, and promote active participation the following components are essential: teaching and learning concepts, instructional strategies, online learning platforms, and educational assessment platforms. These factors were connected in the FELCS model, which is introduced in Section 1.2 of this thesis. The initial question was divided into two research questions that are also stated in Section 1.2. In Chapter 2 all of these facets have been described in detail and are building the theoretical foundation (or stem) of engaging learning (see Figure 1.2). Additionally, the previous Chapter illustrates the broad spectrum of computer science education and focuses on the most important areas in the context of this thesis. This thesis further aims to consider engaging learning in computer science through different lenses, including formal and informal education and secondary and higher education.

Several research studies have been conducted within this thesis to answer the research questions. The *Engaging Learning Tree* is the central methodology after which all studies have been designed and conducted. Table 3.1 contains an overview of all studies performed to answer this thesis question and further shows all papers published as a result of these studies. The columns contain the key aspect of the research study but also list the four components of the FELCS model with the specific concepts that are applied in each study.

The first studies have been conducted within the scope of the serious game *sCool*. These studies have observed the impact of game-based learning in secondary education on enhancing students' engagement. The focus was on feedback and assessment in learning environments to analyze the students' level of engagement but also their learning process (Article 1 and Article 2). The game-based environment is mainly related to computational thinking and coding topics. Within these articles, the role of serious games in increasing student involvement and improving their understanding of computer science principles was either observed. Since the previous version of sCool did not facilitate collaborative learning, a concept for a multiplayer game type was designed and developed. The students' performance and their engagement in the learning activity has been evaluated (Article 3).

Based on the experience and findings on the design and development of learning technologies and educational scenarios, the *DigiSkill* (Article 4) tool was designed, developed, and evaluated according to the *Engaging Learning Tree* model. The aim of this study was to investigate a highly flexible and interactive web-based platform that raises awareness for IT security topics using a narrative approach. In general, this system provides a student-centered learning experience focusing on digital skills such as data literacy, coding, and security. Despite the many possibilities this tool offers in terms of learning content, the focus of the study was on security awareness. The findings of this study showed which factors define a successful learning experience in digital competencies using learning technologies. Further, it was observed that IT security is essential in today's computer science education, but existing tools partially cover these topics. Research in this area raised the question of how learning technologies and approaches can support issues where technological and societal aspects intersect.

Motivated by the findings of the DigiSkill intervention, a social media awareness training (SMAwT) was designed, developed, and evaluated using the *Engaging Learning Tree* model. Based on initial workshops with educators and CSEd researchers, the concept for the training has been developed. A study was set up, including secondary schools in Germany and Austria, to investigate the learning platform and the approach. The findings (Article 5) of this study show how social media awareness aspects can be integrated into computer science education using a multi-perspective approach (Dagstuhl Triangle).

The previous studies are all emphasizing secondary schools. However, this thesis aims to cover several aspects of engaging learning. Higher education has many opportunities for interesting interventions due to the variety of learning contexts. To receive a high contrast to the comparatively small groups of learners in secondary education, large-scale introductory courses on a university level are considered for the next studies. These courses are traditionally characterized by large and heterogeneous groups of learners, high drop-out rates, and challenges in providing individual feedback and assessment. For this reason, the data of a large-scale MOOC about teaching object-oriented programming at the university level was analyzed (Article 6). Within this study, users' behavioral patterns were observed to identify motivation, problems, and dropouts and conclude how high levels of engagement can be reached in such courses. MOOCs are (traditionally) defined by an informal learning context where students learn student-centered in a self-paced way. A similar approach is blended learning, which usually takes place in a more formal way. Blended learning is a commonly used instructional strategy in university courses. Since blended learning and MOOCs are two essential concepts in higher education (especially in online teaching), another study was conducted aiming to identify design principles and strategies for engaging learning experiences in online courses (Article 7).

Each study aims to answer the overall research questions. The following subsections will describe all studies, their motivation, contribution, and model in detail. Furthermore, the individual contribution of each author is declared according to the CRediT author statement[1]. Finally, the main findings derived from these studies serve as the foundation for addressing the research questions in Section 3.8.

[1]https://www.elsevier.com/authors/policies-and-guidelines/credit-author-statement

Table 3.1: This table contains all studies that are the output of this Ph.D. thesis.

Study	**Teaching and Learning Concepts**	**Instructional Strategies**	**Online Learning Platforms**	**Educational Assessment**
Engagement in In-Game Questionnaires (Article 1)	Engagement Motivation Constructive Alignment	Game-based Learning Serious Game	Learning Management Systems	Learning Analytics Evaluation Formative Assessment Summative Assessment
Learning Analytics Platform for Serious Game (Article 2)	Constructive Alignment	Gamification Game-based Learning	Learning Management Systems	Learning Analytics Feedback Evaluation
Collaborative Learning in a Serious Game (Article 3)	Self-Efficacy Engagement Motivation	Game-based Learning Collaborative Learning Serious Game	Learning Management Systems In-game Chat	Learning Analytics Feedback Evaluation
An Online Platform for Security Awareness (Article 4)	Bloom's Taxonomy Constructive Alignment Self-Efficacy Engagement Motivation	Gamification Blended Learning	Learning Management System	Learning Analytics Feedback Evaluation

Study	Teaching and Learning Concepts	Instructional Strategies	Online Learning Platforms	Educational Assessment
Social Media Awareness Training (Article 5)	Bloom's Taxonomy Constructive Alignment Self-Efficacy Engagement Motivation	Gamification Blended Learning	Learning Management System	Learning Analytics Feedback Evaluation Formative Assessment Summative Assessment
Behavioral Pattern Analysis of Programming MOOC (Article 6)	Self-Efficacy Engagement	Blended Learning	MOOCS Learning Management System	Learning Analytics Feedback Evaluation Formative Assessment Summative Assessment
Online Teaching in Introductory Programming Course (Article 7)	Bloom's Taxonomy Constructive Alignment Self-Efficacy Engagement Motivation	Blended Learning	Learning Management System Chat/Forums Streaming Platforms	Learning Analytics Feedback Evaluation Formative Assessment Summative Assessment

3.1 Article 1: Engagement in In-Game Questionnaires

Motivation

Within the design of game-based environments, three components are relevant according to Plass et al. (2015): challenge, feedback, and response. Previous work (Steinmaurer et al., 2019) on the serious game *sCool* shows that challenge and response are considered within the game's design. A good balance between the player's skills and the game's challenge should bring the learners into the state of *flow* (Csikszentmihalyi, 1990). The game environment is equipped with the actual video game and an online platform to provide the players with content and teachers with basic learning analytics. Even though the existing web platform gives some insights into the learning process, the feedback component shows potential for further research. However, besides feedback, this also addresses assessment within the platform. Assessment is not only important in (serious) games; it is also fundamental for the entire learning process. A seamless integration of assessment aspects into games is central to their success (Bellotti et al., 2013).

Contribution

A concept for integrating feedback components in a game environment was designed, implemented, and evaluated in a study. The existing web platform was extended so educators could define feedback and assessment instruments and their appearance in the video game. In this way, learners should be provided with a fun and engaging way to give feedback that supports educators to assess and improve the learning process. The study aimed to evaluate both learners' and teachers' perspectives to identify how in-game questionnaires can be used for feedback and assessment in an engaging way. For this reason, an A/B study was conducted with secondary school students within a classroom setting. In addition, 14 experts from teaching, computer science, game development, and pedagogy participated in a questionnaire regarding the pedagogical and educational aspects of the approach.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - Former studies on sCool showed, that collecting meaningful feedback within a serious game environment has its pitfalls. The students get distracted due to context switches, for example when switching to third-party survey tools or analog feedback with sheets. This leads to a low number of responses and a quality loss in the data.
- **Theoretical Foundation** - A literature survey was conducted to consider similar approaches. Studies about feedback within immersive platforms have been considered.

- **Theoretical Concept** - Based on the literature survey a concept for in-game questionnaires has been developed.
- **Implementation** - The implementation phase included both, the technical implementation of the in-game questionnaire and the development of an educational approach to include these questionnaires in learning.
- **Evaluation** - Within the evaluation phase, an in-class user study was conducted with 22 students. In addition, an evaluation with 14 experts was conducted to receive the teachers' perspectives as well. The survey data (mainly qualitative data) was then analyzed.
- **Validation and Best Practices** - The results of the study gave insights, into how in-game feedback can be collected and how this can be used to improve the students' learning experience.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Formal analysis, Writing - Original Draft, Writing - Review, Editing
- **Martin Sackl:** Software
- **Christian Gütl:** Supervision

Steinmaurer A., Sackl M., & Gütl C. (2021). *Engagement in In-Game Questionnaires - Perspectives from Users and Experts*. In: Proceedings of 7th International Conference of the Immersive Learning Research Network (iLRN), 2021, pp. 1-7.

Engagement in In-Game Questionnaires - Perspectives from Users and Experts

Alexander Steinmaurer, Martin Sackl & Christian Gütl

Abstract

Questionnaires are important instruments to gather information from people in a wide range of application scenarios, such as analyzing the responses for evaluations, collecting preferences, or retrieving self-estimations in learning settings. However, based on the setting and design of a questionnaire participating can be boring or frustrating and consequently have a negative impact on results or disengage users. Literature reports negative findings in immersive simulations and learning games, where assessment and questionnaires are provided outside the learning experience. In this paper, we introduce an approach to integrate questionnaires into a game environment. Thereby, we want to provide learners with an engaging way to answer questions. To encourage them in participating, they receive rewards for completing questions. We performed two evaluations, an A/B study with 22 participants and an evaluation with 14 experts in subject-related fields. We could show that learners are more engaged in the in-game questionnaire and integrated questions make them more likely to respond. We could also identify strategies to obtain more reliable responses such as asking questions right after a task or including gamification elements. Findings can contribute to design more engaging applications or learning environments where evaluation and feedback have significance for both educators and learners.

3.1.1 Introduction

Feedback and evaluation have a major impact on all sorts of application scenarios, and focusing on education in particular it is key to be considered in student's learning process. Receiving feedback from students not only allows to collect and support their preferences for interacting with the application but it is also a valuable source for educators to support students in their learning process. Focusing on the learning process, feedback can address various aspects, such as indicating a lack of understanding in a specific task or expressing high-level difficulties. Receiving feedback can be done both orally or written. When dealing with more complex feedback or with a large number of students written feedback is more feasible. Written feedback has another major advantage over oral feedback, it can be anonymous which makes it more reliable. To simplify the task of conducting surveys and analyze the results different tools can be used. There is a large number of tools which fits these requirements such as LimeSurvey[2], Google Forms[3], or Qualtrics[4]. These tools are often used when asking specific questions in between or after certain tasks.

In previous studies related to learning achievements, engagement, or usability aspects we encountered that conducting surveys can be demotivating for pupils and students. Participants often feel forced to answer questions which led to biased answers. In particular, in immersive simulations or virtual lab settings and learning games, students get disengaged and interrupted in their learning flow completing questions and tasks outside the learning environments.

In this paper, we are introducing an approach to conduct surveys integrated into the game environment sCool (Steinmaurer et al., 2019). sCool is a serious game for teaching computational thinking and coding. It is developed within a cooperation between Graz University of Technology and Westminster University for Review. To improve the video game and the teaching content different evaluations (usability, learning progress, motivation, etc.) have been conducted. The aim of this project is to make the task of evaluation and assessment more engaging and less disruptive. We therefore want to answer three defined research objectives: **RO1:** Are integrated questionnaires more engaging for students when playing games? **RO2:** Do in-game questionnaires disturb a learner's game-flow or his or her feeling of presence in the game? **RO3:** Do in-game questionnaires lead to a different NASA TLX (Task Load Index) (Hart, 2006) measured usability? In order to answer these research objectives the main contributions of our work are:

- Developing an in-game questionnaire covering the most common types of questions.
- Comparing in-game with external questionnaires in an evaluation with 22 participants.
- Conducting a survey with 14 experts of various scientific fields to receive heterogeneous feedback on different aspects.

In this paper, we will first introduce some related work in the area of (integrated) questionnaires in learning games. Afterwards, we will describe the implementation of the in-game questionnaires, followed by a section in which we describe an evaluation of the newly functionality. In this section, we will describe the settings

[2] https://www.limesurvey.org/
[3] http://forms.google.com/
[4] https://www.qualtrics.com/

and instruments, the participants of the study as well as the procedure. Finally, we will present and discuss the results and will end this paper with a conclusion and future work.

3.1.2 Related Work

In serious games, collecting students' information is key to adjust the learning and gaming flow accordingly, and might include information about emotions, learning preferences, learning effectiveness and performance. There are different approaches such as analyzing user data collected automatically by the system, or analyzing the feedback from participants. Bellotti et al. (2013) discuss ways to determine skills and learning outcome retrieved by serious games, also including in-game assessment approaches. In-game approaches to assess player performance or engagement can provide detailed material and are a big desire to reduce limitations in other default approaches (Short et al., 2011).

In learning settings, gamification is used as well, such as gamified questionnaires. The term *gamification* has been becoming more popular over the last few years, in which video games have started to play an enormously influential role and are even used more and more often to provide learning content in form of serious games. Deterding et al. (2011) have introduced a well-known definition of this term as *"the use of game design elements in non-game contexts"* and state, that *gamification* is rather referred to *games* than *playing*. This definition is based on three components, which have to be differentiated:

- Gamefulness
- Gameful interaction
- Gameful design

In this context *gamefulness* is related to the quality, *gameful interaction* are the components providing this quality and *gameful design* can especially in serious games be seen as the game elements (Deterding et al., 2011).

A serious game with an integrated, gamified questionnaire is *LenguaDrive* (Frommel et al., 2015), which is a racing game to learn other languages. In the game, learners collect items in form of translations of a given word in native language. They have to switch lanes on a road to select these items. An integrated questionnaire should help to evaluate the game-play and is implemented in two ways: an overlaying questionnaire screen and a game element called *emotion road*, in which learners switch to another road and like collecting items, answer questions by switching lanes and also confirm the answer afterward. To find out the best speed of the car (to give learners enough time to decide and answer), a pre-study has been conducted. In an evaluation with both of the game's questionnaire types, participants were assigned to one of two groups and played the game including answering questions. These involved questions about distractions, natural movements, and immersive activities of the participants while playing the game. Results showed, that a higher experience of presence than interrupting the gameplay was the case for the group using the *emotion road*.

To improve the user experience and engagement in (online) surveys (Harms et al., 2015) are proposing a way to use gamification of surveys. In a case study, two designers applied the gamification process on a sports-based online survey for teenagers. The main goal was to involve several mini-games and allow participants to explore different survey areas openly. Furthermore, they wanted the users to accompany an avatar through this survey and could earn coins by answering questions, which could then be redeemed in a shop. Results of this case study showed a positive outcome of the practical usefulness of this approach by retrieving more positive feedback and the fact, that users spent more time in a gamified version of the survey. Surprisingly, the gamified version retrieved a slightly lower response rate.

3.1.3 Designing and Implementing In-Game Questionnaires

Previous experiences of low return rates and uncompleted answer sets of evaluations in the context of sCool motivated us to integrate questionnaire features into the learning game. In order to make the task of answering questionnaires more engaging and to increase the number of responses, we defined some requirements for our serious game environment.

The sCool environment is developed in a flexible client-server architecture with a REST API in between. Educators can design a whole course in the web application and all content is provided to the client as game experiences both for desktop and mobile devices (further technical details can be found at Kojic et al. (2018) or Steinmaurer et al. (2019)).

Based on the overall design of sCool, the requirements for the questionnaire features are defined as follows:

- Enable teachers and course designers to create the questionnaires in sCool's web application.
- Support for most commonly used types of questions (single-choice, multiple-choice, binary answers, Likert scale, open-ended questions)
- High flexibility in the appearance of questions (after tasks, concepts, end of

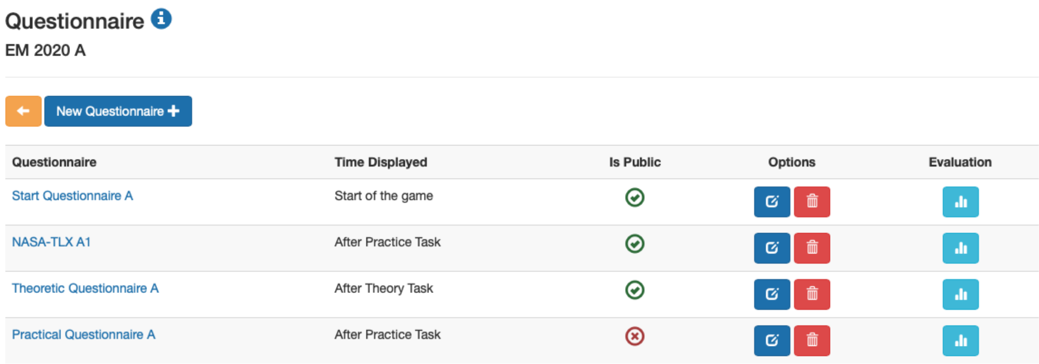

Figure 3.1: Educators can create several questionnaires within one course. The backend provides educators with basic functionality such as modifying or deleting questionnaires. It is also possible to define general rules for the whole questionnaire that are inherited for all questions.

the game, etc.)

- Rewards for participation such as items or additional equipment within the game environment.
- Create templates for certain standard questionnaires (such as usability, difficulties, etc.)
- Storing the player's answers and related data (timestamp, question skipped, etc.) within the server's database.
- Basic data analyzing within the tool (mean, standard deviation, diagrams, etc.) and data export for further analysis by an external tool.

Since the learning tool does not only involve the game itself, but also the web platform for educators, the implementation of the integrated questionnaires was developed in two parts: The server and web application for creating the questionnaires by educators, and the integration into the game experience for the students. The current database schema had to be extended by three tables: *Questionnaires*, *Questions* and *Answers*. Additionally, two modules of the ASP.NET web application were updated. The admin web backend, in which educators are able to create courses and tasks for the game, was extended by functionality for creating questionnaires and the API for communication between server and game was modified.

On the web platform, users are able to create several questionnaires per course, including again several questions. Figure 3.1 shows questionnaires created for one course over the web platform. As shown in Figure 3.2 a single questionnaire contains several questions. There is also the possibility to mark a questionnaire as public and let other educators duplicate this questionnaire to their own course.

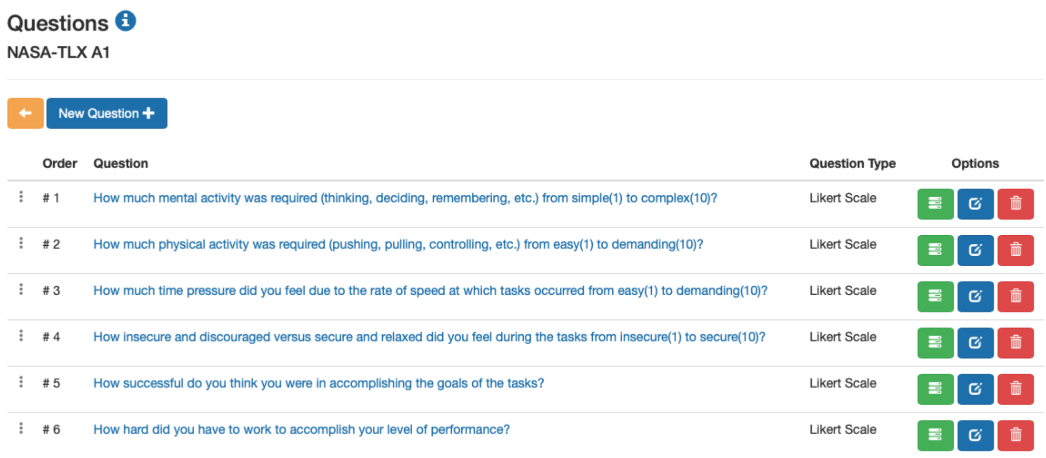

Figure 3.2: One or more questions are assigned to a single questionnaire. Each question can have a specific type (Likert scale, multiple-choice, etc.) Since some questionnaires require a defined order, educators can define it.

When creating questions, users are able to define its title, select the type of question (open question, Likert scale, single-choice or multiple-choice), labels for answers, and moment of appearance while playing the game. In order to add gamification aspects, users are able to set rewards for answering questions, which learners can earn credits to buy special items. Figure 3.3 shows all information that is relevant when creating a questionnaire. Besides editing and deleting questionnaires, users are also able to statistically evaluate given answers in an extra view. This should enable educators to get some insights into the evaluation results and this should be a starting point for further data analysis.

In the video game sCool, a module for interaction with the questionnaires was implemented. The view of a single question was added in form of a full-screen pop-up, shown if a certain action is triggered. As presented in Figure 3.4 the question is using the game's user interface to integrate the module into the game-flow. The appearance of a question (or a series of questions) depends on the educator's configuration in the web application. This is possible when starting a new course, as well as after successfully passing different tasks in the game. Learners are always able to skip questions and re-answer them at a late moment. For this purpose, a menu item was added in the course overview, which lets the user answer skipped questions. If a user plays the same task multiple times, the questionnaire will only be shown, if the user has not answered it before.

In the top right corner of the question pop-up, a coin and the number of rewards is displayed, which he or she can earn by answering single questions. These special rewards can then be used to buy items in the in-game store. For questions that are

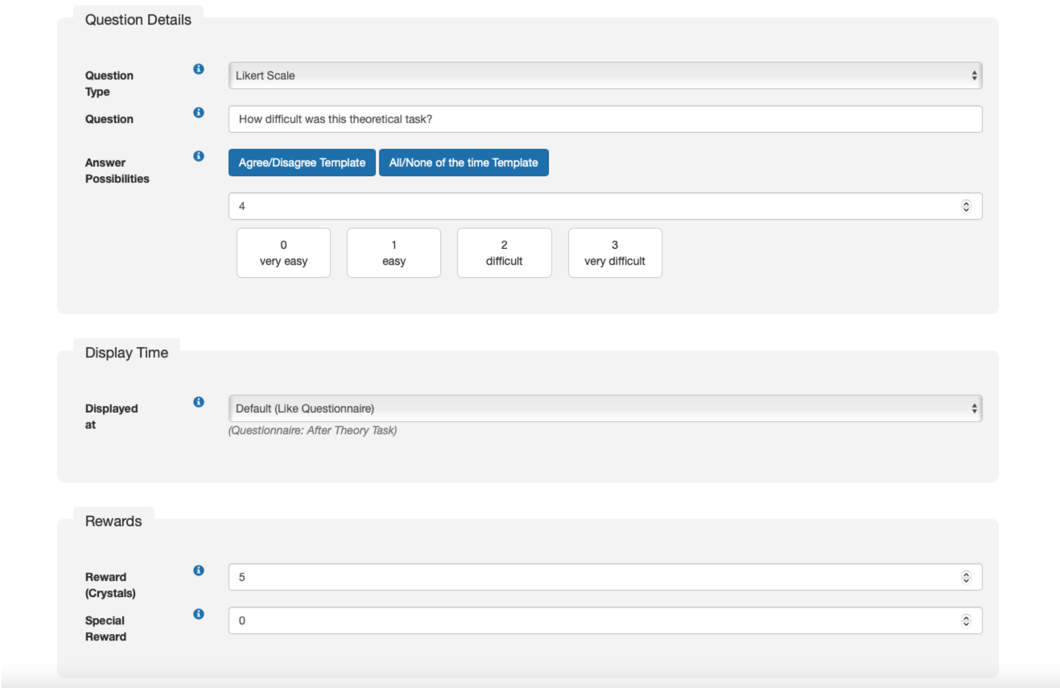

Figure 3.3: Different settings can be selected for a corresponding question. Besides general information such as the questionnaire's name and category, more specific options can be chosen.

more valuable, it is also possible to unlock certain items, that just can be acquired by answering questions.

3.1.4 Evaluation

Two different studies were conducted, in order to evaluate the new implemented in-game questionnaires and to be able to answer all defined research questions: an A/B-testing user study and an expert study.

Settings and Instruments

In order to evaluate the integrated questionnaires in sCool an A/B-testing user study has been conducted. In this study, participants were automatically assigned to one of two groups, where each group played two rounds of the game, each round with one of both approaches including i) in-game questionnaires, and ii) external questionnaires. The in-game questionnaire was evaluated using the video game's survey module. For the external questionnaires, the LimeSurvey tool was used. Furthermore, interactions with the game were recorded anonymously for analyzing purposes. This data includes the learning rates of the learner's attempts and success rates per task.

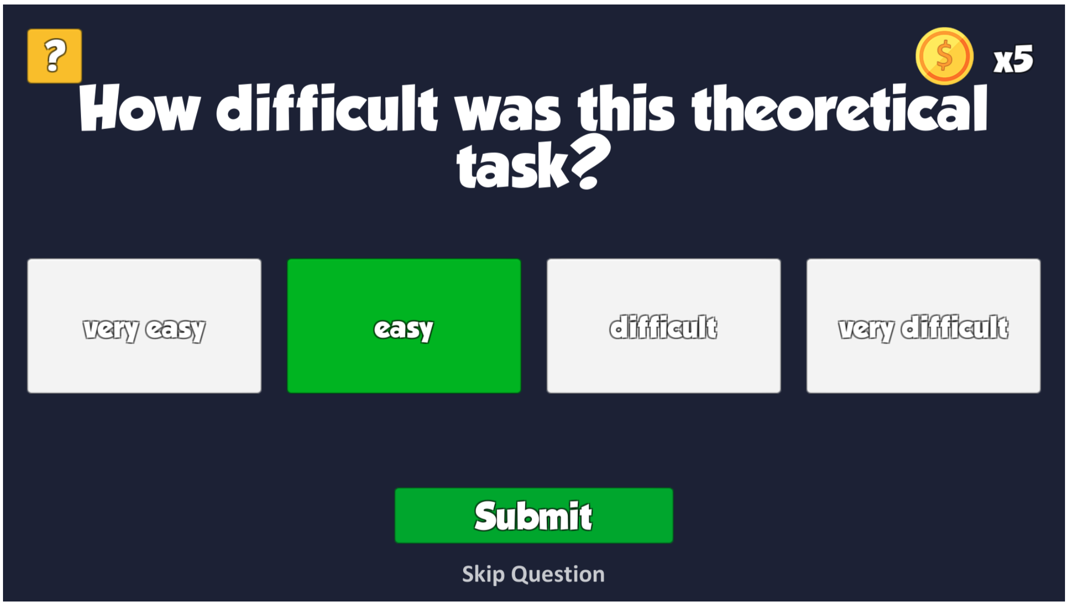

Figure 3.4: The questions and the labels are displayed in full-screen size within the video game. Depending on the category of the question is displayed differently in the game. The screen also shows the value of a specific question (upper right corner).

Table 3.2 shows questions that all participants of the A/B user study received. Group A answered these questions within the video game and group B answered them in the external tool. Table 3.3 shows the questions of the final questionnaire that were answered using LimeSurvey at the end of the evaluation activity. In addition to the game-specific questions, learners were asked to fill out a NASA TLX (Hart, 2006) survey, to measure their workload using the type of questionnaire. These NASA-TLX questionnaires were shown to the player as integrated game elements, as well as an external questionnaire. This survey involves six Likert scale questions in a range from 1-10 regarding different perspectives of the learners:

- Mental activity
- Physical activity
- Time pressure
- Frustration
- Performance
- Effort

In order to extend the perspectives on in-game questionnaires, a second evaluation was conducted. Therefore, experts from different fields participated in an additional study. They were asked to answer questions related to various aspects of in-game questionnaires (see Table 3.6). Therefore, the experts filled out general questions about environments for conducting questionnaires in educational contexts. After these initial questions they watched two videos, the first video introduced our approach of answering questions using the sCool video game and the second one was about answering the same questions using LimeSurvey. As a third activity, the experts respond on a Likert scale and provide open-ended feedback about these two approaches. In contrast to the participants of the A/B-testing study, all expert users answered the questions using the LimeSurvey tool.

Participants

We primarily asked computer science students from Graz University of Technology to participate in the user study. The video game's purpose is to learn basic concepts in coding, and computer science students can mainly focus on the game's environment and questionnaire module instead of solving tasks. However, we also wanted to include opinions on novice programmers and therefore asked some non-experienced students to participate in this A/B study. In total 22 students took part in this evaluation. The participants were equally assigned to both testing groups.

The second evaluation was conducted with experts of relevant fields to cover the interdisciplinary aspects. Figure 3.5 shows the research fields of the experts, whereby multiple selections were possible. In total 14 experts took part in the evaluation. The majority of them (10 out of 14 people) obtain a Ph.D. in one of their selected fields, the other experts have a Master's Degree. The range of professional experience is between 2 to 35 years, the mean is 17.53 years (SD=9.69).

Procedure

In regards to the A/B testing group A started with the in-game questionnaires in the first round and used the external questionnaire in the second one. Group B played the same order of tasks and answered the same questions, but started with the external questionnaires and used the in-game questionnaires later.

Each round of the game included one concept-learning task[5] and two practical tasks (6 tasks in total per participant) and a questionnaire at the beginning of the round, one after the concept-learning task and one after the second practical task. The integrated approach included three questionnaires: one at the start of the round,

[5]The serious game divides tasks into concept-learning and practical tasks. In concept-learning tasks, learners acquire new knowledge in an explorative way while practical tasks focus on applying these concepts in an engaging programming environment. More information regarding the video game and game types can be found in the related paper Steinmaurer et al. (2019).

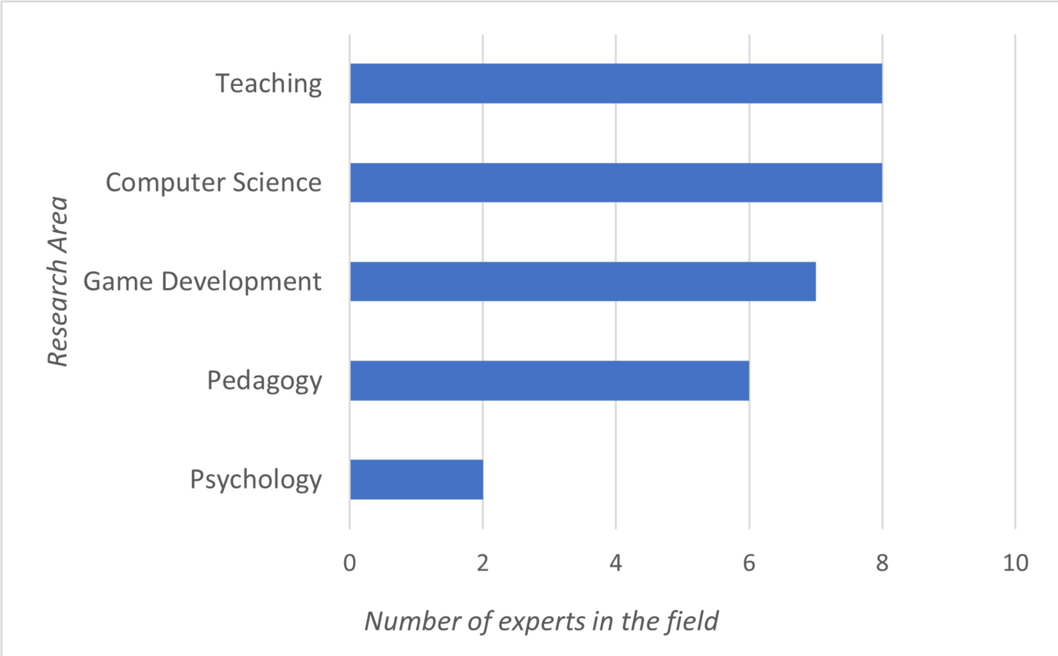

Figure 3.5: This Figure shows the expert's fields. 8 out of 14 do have professional experience in teaching and computer science. 7 are in the field of game development. 6 experts are working in pedagogy and 2 are working in the area of psychology.

one after the concept-learning task, and one after the practical task, whereas the external questionnaire was only used after a whole round in order to keep the player's game-flow high.

After playing both rounds of the game, the participants were asked to fill out a final questionnaire involving three Likert scale questions about the personal preference of questionnaires, the level of difficulty, and the level of disruption of the game-flow. Additionally, two text fields for thoughts, opinions, and possible problems were given. See Table 3.3 for the included questions.

The expert's filled out a questionnaire covering 12 items in two phases. In the first phase (before the videos), the participants answered three open-ended questions regarding the advantages and disadvantages when conducting a questionnaire in the same environment and about reasons why participants do not complete surveys. In addition, three Likert scale questions were asked, covering the impact on a person's flow (#1), the accuracy of answers (#2), and participation rate (#3). In phase two (after the videos) the participants were asked to elaborate advantages and disadvantages of our approach compared to an external tool. These open-ended questions were accompanied by Likert scale items related to gamification and user

Table 3.2: Questions After In-Game Tasks.

Question	Category
How old are you?	Single-choice
How would you rate your programming skills?	Single-choice
How difficult was the concept-learning task?	Likert scale
The code editor is easy to use.	Likert scale
What type of student are you?	Single-choice
Have you ever played an educational game before?	Yes/No
Did you pass the concept-learning task at first try?	Yes/No
The practical tasks descriptions are easy to understand	Likert scale

Table 3.3: Questions of Final Questionnaire.

Question	Category
Please enter your username	Open Question
Which type of questionnaire did you like more?	Single-choice
Which type of questionnaire was easier to use?	Likert scale
Which type of questionnaire delivered a better game-play?	Likert scale
Please explain your ratings	Text-area
Were there any problems with the in-game questionnaire?	Text-area

experience in the in-game environment.

3.1.5 Results and Discussion

User's Evaluation

Overall 26 people started with the study, but just 22 participants solved all tasks and answered the final questionnaire and thus completed the whole evaluation. Since the study was conducted as an unsupervised online study, it cannot be exactly reasoned, why some participants did not finish the study, but in debriefings with a small number of participants, it was recorded that some of them forgot or over-read the instructions or simply did not get back to the instructions page after finishing the second round of the game. Since the final questionnaire was on the externally used tool, regarding **RO1** (engagement in answering questions) a higher participation rate in using the in-game approach for the final questionnaire is only a matter of conjecture.

When evaluating the success rates of the learner's tasks, some slight differences in the approaches can be discovered. Therefore, the in-game and the external method of both groups were separately evaluated. The in-game questionnaires showed an overall success rate of 98.48% and an average of 1.92 attempts per task. On the other side, when using external questionnaires the success rate is at 95.45% and had 2.23 attempts per task. These results show a slightly (but not significant) improved performance of the learners when using the in-game approach and regarding **RO2** could therefore be interpreted as a slightly better game flow with this approach.

Regarding the participation in answering questions, the results almost showed no difference. While retrieving (almost) 100% level of answering questions in the in-game questionnaires (with the exception of one single question not answered by a single player), the external approach also retrieved a 100% participation rate for group B, but only 92.85% for group A. This means, that participants did not use the external questionnaires more likely after one round with in-game questionnaires and thus the integrated questionnaire seems to be more engaging (**RO1**).

Table 3.4 shows the results (mean) of the given answers about workload regarding answering questions of group A and Table 3.5 shows the results of group B. To answer **RO3**, from these results it can be observed that both groups stated very similar feelings about mental activity and time pressure, which show a (slightly) higher workload when using external questionnaires. Both groups also shared the same opinion on performance and effort, which show a higher feeling of performance in using in-game questionnaires and a higher effort in using an external questionnaire tool. The only results which differ between the two groups

Table 3.4: NASA-TLX results from group A.

Scales	In-game (Mean)	External (Mean)
Mental activity	2.71	3.55
Physical activity	3.14	2.18
Time pressure	2.00	2.82
Frustration	7.64	4.27
Performance	8.43	7.45
Effort	2.43	3,55

Table 3.5: NASA-TLX results from group B.

Scales	In-game (Mean)	External (Mean)
Mental activity	2.42	5.23
Physical activity	2.00	3.00
Time pressure	2.42	2.46
Frustration	3.33	6.23
Performance	7.67	4.46
Effort	2.08	6.00

are physical activity and frustration.

The final questionnaire, which both groups were asked to fill out on the external tool, showed a clear tendency to in-game questionnaires. Participants could select on a range from 1 (= in-game approach) to 10 (= external approach), i) which type of questionnaire they liked more, ii) which one was easier to use and iii) which one is delivering a better game-play. The results show a mean of 2.73 regarding the learner's preference of the questionnaire, a mean of 3.23 regarding the simplicity, and a mean of 3.45 regarding game-play. Learners were also able to use text fields to give feedback. Most of the answers covered the same opinions like:

- *"I don ´t really see a point to why not put those questionnaire in the game. It might take the user's attention from the 'playing' part of the game to a more 'why/what did I do' to answer the next question which seems to be more fitting for this type of game."*
- *"It just feels easier and more natural to do the questionnaire during or within the game so your thought process continues and is not interrupted by switching from the game to an external questionnaire."*
- *"Navigating to another website, in order to do a questionnaire is a bit cumbersome and interrupts the game-play."*.

Examples of negative opinions are:

Table 3.6: Questions, Mean and Standard Deviation for Expert's Evaluation.

ID	Statement	M (SD)
#1	Asking questions within the same environment has a positive impact on the person's flow.	2.21 (0.69)
#2	Asking questions right after a certain task increases the accuracy of the answers.	1.93 (1.21)
#3	The participation rate increases when asking questions within the same tool.	2.21 (0.69)
#4	Using gamification elements such as additional items, additional levels, or bonus points increase the participation rate.	1.57 (0.94)
#5	The diversity of participants is influenced by different gamer/learner types (such as achievers, or learner's that perform mainly well).	2.21 (0.80)
#6	Rewards have an influence on the validity of the answers.	2.14 (1.09)
#7	The consistency of answers is different when comparing an in-game to an external approach.	1.71 (0.73)
#8	Within gaming environments learner's are more likely to answer questions.	1.64 (0.93)

- *"The UI elements overlapped each other from time to time"*
- *"gameplay is interrupted by having in-game questionnaires"*

Expert's Evaluation

The experts identified different advantages and disadvantages in the in-game approach. A central benefit is the ability to ask questions right after a certain task within the same system. By doing so, users get less distracted and stay engaged in the game environment: *"Breaking them up like this makes it easier to complete, more enjoyable and less fatiguing."*

Adding gamification elements such as rewards can increase the participation rate. Users get the feeling of providing more valuable answers by assigning rewards to answers. The experts also argue that gamification elements have the additional benefit of increasing the participation rate (M=1.57, SD=0.69). They are little concerned that rewards do have an influence on the validity of the provided answers (M=2.14, SD=1.09). Nevertheless, some experts think that too many questions can lead to a behavior where learners tend to answer questions just to receive rewards.

Another important aspect is the user's flow while being engaged in the game. The experts predominantly see a positive impact on the flow (M=2.21, SD=0.69). Since the questions can be asked right after a task the answers get more veritable. However, when asking questions between in-game tasks too often, this can also be perceived as distracting. Especially when answering serious questions the user needs full mental capabilities, which will probably be not the case within the game environment. Due to the less formal context of the game, especially aspects related to learning or the environment can lead to good results.

To get a better understanding of reasons to not participate in surveys, we collected the responses of the experts and categorized them:

- **Previous Experience:** Prior experiences have a major impact on future participation in surveys. If participants have experienced that their opinion is relevant they are more likely to do evaluations again. When having a positive experience with a previous survey a person is more willing to participate.
- **Time Constraints:** People tend to avoid participation or complete surveys when having time constraints. This could be on the one hand related to a long survey with many items or on the other hand to limited time resources (for example when answering questions within class).
- **Privacy Concerns:** Evaluations do have a negative connotation in regards to privacy. Participants are feared of possible consequences and answers can be biased to some opinions. Privacy concerns can be negatively strengthened when there is a possible link between a person and its data (such as the same environment, identifiers, etc.)
- **Motivational Aspects:** The participant's personal involvement in a certain topic is important for attending a survey. They want to see a purpose in a survey and its importance. The contribution should feel useful.
- **Personal Aspects:** The decision to participate in a study can also be influenced by subjective factors such as acceptance of a topic, or the usability of the tool. People might also feel that they do not have a strong opinion related to the topic or do not have enough expertise.

3.1.6 Conclusion and Future Work

In this paper, we have presented an approach to conduct surveys in a serious game environment. Due to the diverse expertise of the expert evaluation, we were able to identify advantages and disadvantages in our in-game survey approach. Additionally, the results of the user evaluation showed, like in the evaluation of the learning game *LenguaDrive* (Frommel et al., 2015), more engagement and less effort in solving tasks and answering questions. Integrating a survey into a game environment can increase the player's flow, especially when answering questions related to the tasks. In contrast to Harms et al. (2015) which retrieved a slightly lower response rate, in our study we obtained a slightly higher response rate with in-game questionnaires. Both learners and experts argue that using the same environment simplifies the evaluation.

Gamification. Combining questionnaires with gamification can increase the participation rate and the reliability of answers. In a game-based system elements such as rewards, bonus points, etc. can be a motivation for learners. Instead of rewards gamification could be used in the other direction, for example, to prevent users from losing points for wrong answers.

Engagement. The evaluations showed that the game experience stays high, when not getting distracted by changing to a different tool. Integrating surveys into the user interface of the game also makes the questions appear as part of the system.

Security and Privacy. To get meaningful results a trustful environment is necessary where users feel safe. In both evaluations, the participants respond that privacy and security are important to get honest answers. On the one hand, environments, where users might be identified, have a disadvantage in confidence. On the other hand, using the same environment for answering questions can also increase the trust of security because no third-party software is used.

Future Work. The evaluation is limited to a low number of participants. We plan to conduct further research with a broader group of students. Based on the feedback of some participants we also plan to make some improvements to the design of the in-game questionnaires to improve readability on mobile devices. We also want to extend gamification aspects and include more elements such as additional levels or game types as a reward. It would also be very interesting to discover, which times are suited best for showing questions to the players in order to not disturb their game-flow.

3.2 Article 2: Learning Analytics Platform for Serious Game

Motivation

Previous work on in-game questionnaires within the serious game sCool already emphasized the importance of assessment within learning environments (Steinmaurer et al., 2022). A sophisticated assessment further requires learning analytics to monitor the learners' process. Figure 1.1 shows that engaging learning is highly influenced by educational assessment - an area lacking in the serious game. The process of game learning analytics involves i) choosing data, ii) capturing data, iii) aggregating data, iv) analyzing data, and v) deploying results (Shoukry et al., 2014). Learning analytics in game-based learning approaches is relevant in coding education as it allows educators to track and analyze students' progress, identify areas of difficulty, and provide personalized feedback. By generating data from gameplay interactions, learning analytics can offer valuable insights into students' coding skills, problem-solving strategies, and learning patterns, enabling targeted interventions and instructional adaptations for improved learning outcomes.

Contribution

In the existing sCool environment, assessment and learning analytics were just available superficially. The requirements of a learning analytics platform were analyzed, a conceptual design was created and the system was implemented. An evaluation of the learning analytics platform was conducted including usability and educational aspects. Therefore, an usability evaluation with 31 experts was conducted first, followed by an evaluation with 8 teachers. This study aimed to understand how such a system should be designed and implemented, focusing on coding education within a game-based environment.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - When it comes to learning analytics, the literature indicates that standardized methodologies are missing in the field of computer science education (Alonso-Fernández et al., 2022; Serrano-Laguna et al., 2017; Shoukry et al., 2014). In addition, previous studies with the existing sCool platform show that teachers require more information than the already existing dashboard provides.
- **Theoretical Foundation** - A literature survey about learning analytics approaches in game-based learning (primarily related to coding and computational thinking) was accomplished.
- **Theoretical Concept** - The theoretical concept is based on the results of the literature survey and findings from previous studies with sCool. A central

point of the concept was to include the already collected user data for the learning analytics.

- **Implementation** - A web-based prototype for a learning analytics platform was developed. Additionally, educational approaches have been developed on how the platform can be used.
- **Evaluation** - The developed prototype was evaluated in two studies with technical experts regarding usability and with domain experts regarding its educational abilities.
- **Validation and Best Practices** - The findings from the study helped to get a better understanding of how learning analytics tools should be designed and which elements are used by educators. Implications for practitioners and researchers have been derived form these findings.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Formal analysis, Writing - Original Draft, Writing - Review, Editing
- **Anil Kumar Tilanthe:** Software, Methodology, Resources, Writing - Original Draft
- **Christian Gütl:** Supervision

Steinmaurer, A., Tilanthe, A.K., & Gütl, C. (2021). *Designing and Developing a Learning Analytics Platform for the Coding Learning Game sCool*. In: Proc. 14th Int. Conf. Interact. Mob. Commun. Technol. Learn.

Designing and Developing a Learning Analytics Platform for the Coding Learning Game sCool

Alexander Steinmaurer, Anil Kumar Tilanthe & Christian Gütl

Abstract

Learning Analytics are a valuable way for teachers and educators to support their student's learning progress. Data can be used to identify issues, provide personalized learning, and improve the overall quality of learning. There are diverse types of educational data which are available for instance log data, given answers, or source code. Just as diverse as the data are, so are the data sources such as learning management systems (LMS) or serious games. However, analyzing and visualizing a huge amount of data in a valuable educational way can be challenging. A learning analytics tool should support learners and educators as good as possible and find a balance between an optimal overview and complex evaluations and interpretations. In this paper, we describe our approach to create a learning analytics software for the serious coding learning game sCool. We outline the concept, architecture, and development of the application. The tool was evaluated in two separate evaluations with i) 31 technical experts and ii) 8 domain experts (teachers). The evaluations showed, that the users rated the usability of the system high above average (M=79.02; SD=10.69) in terms of System Usability Scale. Nevertheless, the evaluation with the domain experts revealed, that there is room for improvements regarding educational usage. Finally, we summarize our experience and results of the evaluation to outline relevant requirements and implementation details for a learning analytics platform. In this way, we want to help educators, researchers, and developers when designing and implementing learning analytics tools.

3.2.1 Introduction

The COVID-19 pandemic has highlighted the importance of effective distance learning with widespread school closures across the world (UNESCO, 2021). Serious games provide an effective, engaging, and immersive learning environment where students can participate in the learning process remotely as well. The term serious games refer to *"games that do not have entertainment, enjoyment, or fun as their primary purpose"* (Michael R. & Chen, 2006). Most academics refer to the use of digital games for educational purposes by the term serious games (Abt, 1981; Djaouti et al., 2011).

The need for modern learning technologies emerges among many educational fields. Computational thinking and coding are essential skills in today's digital world (Barr & Stephenson, 2011; Kanaki & Kalogiannakis, 2018). Many novice programming learners face various difficulties while learning coding skills (Lahtinen et al., 2005). These difficulties have to be acknowledged and recognized to support students as well as possible.

There are several ways to help students when learning to program such as personalized code review by educators or pair programming (Albluwi & Salter, 2020). Getting feedback on a learner's source code is necessary to encounter possible improvements and to increase coding skills. Therefore, educators need to identify students' progress and analyze submitted programming solutions to understand how students learn, plan improvement of the course, and also support and guide the learning experience. Educators and developers could also discover various strategies undertaken by students for course completion. Our study tries to identify how a learning analytics application could assist educators in improving course content based on student's performance (Chaudy et al., 2014; Hauge et al., 2014; Verbert et al., 2013). The study also attempts to analyze how a learning analytics application assists educators to gain an overview of students' performance. Additionally, the user interface of the learning analytics application is evaluated using System Usability Scale and NASA-TLX questionnaires for its usability and acceptability.

In this work, we present a learning analytics tool for a serious game to learn to code. Besides students, such learning systems should also support educators to get a quick overview of a class's learning progress. For this reason, we conducted two studies with different objectives. The first study was conducted with 31 participants with computer science background. The second study was an expert's evaluation with 8 teachers investigating the educational benefit from our system. The following research objectives were defined to evaluate the developed platform.

- Does the platform support all requirements teachers have in a learning analytics tool?
- Are teachers capable to analyze students' learning progress?
- Are there any major usability issues in the current prototype?

3.2.2 Related Work

Digital games can collect vast amounts of detailed information about students learning process compared to traditional methods. The gathered information can be used for behaviour and learning analysis and assessments. Digital games can capture student's inputs, the number of attempts, strategies used for course progress, time allocation to various course stages, problem-solving sequences, or programming solutions submitted (Department of Education, 2010a). The data can be analyzed to provide feedback to various stakeholders involved, improve course content and instructions, gain an overview of student's performance, or for discovering how students learn to program (Chaudy et al., 2014). Digital games are also able to capture student's actions as they work on the course tasks or programming assignments, such as individual keystrokes, programming errors, code edits, and compilations (Ihantola et al., 2015).

The insights from the data gathered by educational games can be displayed to different groups of people such as learners, educators, parents, or developers via learning dashboards. Learning dashboards integrate information from learning tools to provide a comprehensive visual representation of student's progress (Department of Education, 2017). The insights and feedback ensure that relevant data inform decisions about learning and course content.

There are various serious games for teaching programming skills focusing on learning to code, algorithmic thinking, or creating games for learners of various age groups and previous experiences (Combéfis et al., 2016b). *CodeMonkey*[6] is an app and web-based educational game where kids can learn to code with CoffeeScript and Python. Players can also use a block-based course, where they can drag-and-drop blocks of code to control an avatar. The players can learn many programming concepts such as variables, objects, conditionals, function calls, or loops. CodeMonkey also provides a dashboard where educators can track the learning progress of students and see code submitted. The teachers can see the progress of each of the participating students, and export results and progress, or receive a more detailed analysis. They can also see all programming concepts such as loops, functions, and objects and student's proficiency in each of them respectively.

Ozaria[7] is another web-based serious game for learning to code. It is an immersive story-based fantasy learning environment where programming courses are taught in JavaScript and Python programming languages. It is designed for both in-person

[6] https://app.codemonkey.com/
[7] https://www.ozaria.com/

and remote learning settings. The players control an avatar using code to fulfill tasks. The players are also provided with audio and textual hints to help them complete programming tasks. Ozaria also provides a dashboard for students where they can see their courses and progress. The dashboard is also available for teachers showing a class view of students and their progress. Teachers can see all levels completed by students and their in-game progress and assigned levels as well as the code submissions of students.

The mobile learning game *sCool*[8] is an educational game for learning programming skills (Kojic et al., 2018; Steinmaurer et al., 2019). sCool provides an immersive and engaging experience to students following a narrative of the escape of a space shuttle and its crew members from an alien planet. The players learn various programming concepts such as sequencing, loops, or data types in a concept-learning part. It is followed by a practical programming part where the players apply the previously learned programming concepts using the Python programming language to control a robot avatar. The programming section consists of draggable code blocks which get converted to editable Python commands. In additional, the code can be adapted using the device keyboard. The game is highly flexible since game-related data is sent and received using a web application for educators. In this way, teachers can create courses and get an overview of the class' learning progress.

3.2.3 Design and Implementation

A web-based learning analytics application was developed to provide insights into student's activities and course progress to stakeholders, comprising educators, teaching assistants, or game developers, of the mobile learning game sCool. The application focuses on empowering insights into student's activities and analysis of courses, rather than making a decision tool. The empowered educators could utilize their knowledge to improve course content or gaining an overview of their class students. The students can see the structure of the entire course and their course progress within the mobile game itself.

Data Collected

The sCool mobile game captures player's game-related data at a granular level such as dragged code blocks, compiled code, deleted code, interpreter errors, submitted answers, or results code execution. This data is collected and stored in the sCool database for further analysis (Ihantola et al., 2015). The time duration of every game session and information such as points are transmitted to the web server using a REST API. All the data is stored in a relational database. Some of the log data is

[8]https://scool.codislabgraz.org/

also stored as JSON.

Requirements

The players' game-related data are collected in the sCool database should be processed to extract meaningful information which must be provided in simplistic ways (Freire et al., 2016; Owen & Baker, 2019). The information should empower educators with insights into student's course progress and performance. Interactive data visualization tools could be used to make information easily available. The focus of the application should be on lower-dimensional plots to make the visualization easy to understand. For simple statistics, bar charts, or scatter plots could be used to display game metrics of student's game-related data. Graphs such as time series, or Gantt charts could be used to visualize student's game timelines and strategies undertaken for course completion.

Users should be able to just access data that is relevant for his or her learning activity. Within the learning analytics tool, educators create so-called *learning activities* which are in-class activities with a certain group of students. The educators should be able to access data of only their learning activities. Only administrators should have access to the data of all students on the application. Hence, the users were divided into two main groups of educators and sCool game administrators representing the stakeholders. An administrator can gain additional insights by comparing various learning activities, whereas an educator can only compare within their learning activities. Both serious game and web application do not store sensitive data of students, such as names, email addresses, or phone numbers.

Development Details

Plotly's Dash[9] is used as the application framework as it is an open-source full-stack framework for building web analytic applications with interactive visualization. The application connects with the sCool database to access game-related data. The queried data is used to create a Pandas[10] DataFrame, which is a 2-dimensional data structure containing labeled axes (rows and columns), for storing and processing data (Pandas DataFrame, 2021).

The code solutions submitted by students are parsed to create an Abstract Syntax Tree (AST) to identify programming concepts learned and used by the students for each of the practical tasks. Python's *ast* module is used to generate the AST. The log data stored as JSON string in the database is used to extract information such as UI interactions in the game. Besides that, each code execution within the game

[9] https://dash.plotly.com/introduction
[10] https://pandas.pydata.org/docs/index.html

is transmitted to the server giving detailed information on the code producing progress.

When a user selects a learning activity, the related data is displayed using various visualization techniques. The important numbers and information are highlighted and presented. Interactive horizontal bar charts are used to display sorted information as a simple visualization technique for easy interpretation.

The user interface is designed to be responsive and adaptive to various screen sizes. It is divided into three main sections. The *details* section comprises an overview of a single learning activity. The users can gain information such as the number of student participants, points collected, task-wise completion rates, student-wise task completion, and programming concepts learned and used by students. The *student's* section provides a detailed overview of a specific student. This compromises detailed information such as points collected, programming concepts learned, course progress, game interactions, and a timeline. In the *custom* section, users can create custom graphs by selecting features, graph types, and other information. Information such as task-wise errors by the students, or course-wise errors faced by students can be easily discovered using the custom section. The application components follow Model View Controller (MVC) design pattern (Leff & Rayfield, 2001).

3.2.4 Evaluation

Within this research project we have conducted two evaluations with a different focus. The first evaluation (study 1) mainly involved technical experts such as programmers, researchers, or graduated computer science students to get perspectives and opinions on usability, system design, and interaction. The second evaluation (study 2) focused on domain experts and so teacher's conducted the evaluation to get information on the system's characteristics in an educational setting. Both groups were asked to work on several tasks on the system and additionally fill-out a questionnaire regarding system-related properties such as usability.

Participants

Study 1. The first evaluation (system's usability evaluation) was conducted with 31 participants in total. A vast majority of the participants (25 people) have a background in computer science with different professional experiences. Table 3.7 shows that most of the participants (20 people) obtain a Master's degree in computer science. Seven people hold a Bachelor's degree in computer science and two people have a PhD and a high school graduation. In terms of the gender

Table 3.7: Overview of the participants of study 1.

Gender		Highest Degree		Profession	
23	Male	20	Master's degree	14	Expert (CS-related)
8	Female	7	Bachelor's degree	8	Other Employment
		2	PhD	5	Full-time student
		2	High School	4	Teacher

distribution 23 males and 8 females participated the evaluation. A predominant part of the participants (14 people) is working in computer science-related areas.

Study 2. The domain expert's evaluation covers eight teachers, 5 female and 3 male. The evaluation was taken in the scope of a teaching training for junior teachers. The mean professional experience is 1.06 years (SD=0.63). All of them hold a Bachelor's degree from a university (n=2) or a college of education (n=6). Seven are teacher's at a vocational schools and one as a teacher at the college of education. All of them have their degree in computer science education (with a strong focus on multimedia and design).

Instruments

For the evaluation in study 1 the participants were asked to complete 11 tasks within the learning analytics environment. The tasks were designed as multiple-choice questions covering different aspects of the system such as analysis on course, user, or concept level. Each question was worth 1 point so participants could receive 11 points in total when answering all questions correct. The tasks were assigned to one of four categories covering: i) tasks solved in the game (3 tasks), ii) comparing students (2 tasks), iii) used coding concepts (2 tasks), iv) student information (3 tasks), and v) custom plots (1 task).

After each group of tasks (five times in total) the participants received a modified version of the NASA-TLX (NASA, 1986) to investigate i) mental demand, ii) temporal demand, iii) overall performance, iv) effort, and v) frustration level. This assessment instrument's goal was to evaluate the workload of certain tasks to get a better understanding on the system's complexity. The questions of the NASA-TLX were answered on a five-items Likert scale ranging from very low to very high.

The System Usability Scale (SUS) (Brooke, 1995) was used to easily receive an overview of the user's perceived usability. Within this 10 items questionnaire the participants rated the items on a 5 point Likert scale from strongly disagree to strongly agree. Using the SUS questionnaire we wanted to get a quick evaluation on the overall usability and some possible issues to tackle them in future improvements.

The value of the system usability scale can be between 0 and 100, and it is common (Brooke, 1995) that values above 68 are considered as above average usability.

The focus of the evaluation in study 2 was different, since it relates to the system's educational usage. The users were asked to solve five different tasks within the learning analytics platform. Table 3.8 contains all given tasks. Additionally the teacher's were asked *"I found task x ..."* after each question rating from very easy to very hard. Finally, they received questions whether they would use the system and where they see room for improvements.

Table 3.8: The participants of the expert's evaluation were given five tasks to different in-game analysis. The tasks covered different analytics where various parts of the web application should be tested. A task was classified as solved, when a participant answered it fully correct, if this is not the case it is counted as wrong. Participants that were not able to solve a task (e.g. provided no answer) are classified as N/A. The difficulty is calculated using the mean of a 5-items grading scheme from 1 (very easy) to 5 (very hard).

#	Question	correct	wrong	N/A	Difficulty
T1	Find out which practical task the fewest players were successful at. What is the number of this task?	2	3	3	3.5
T2a	In which practical task were variables used most frequently? What is the number of the task?	5	1	2	3.63
T2b	How many players used variables in this task?	3	3	2	2.63
T3	Which players were able to solve the level *"Calculate the Fibonacci Sequence and check it with the storage"* (Task 64)?	6	-	2	2.13
T4	How many attempts did the player *'dabod'* need to complete the level *"Decide what you have to print to the console"* (Task 63)?	5	1	2	3
T5	Which players had the most syntax errors when creating their codes?	1	3	4	3.63

Procedure

The system evaluation was sent out to 60-80 people in computer science-related fields. The questionnaire was provided using Google Forms and shared via communication channels of various research groups, schools, and tech companies. 31 people participated and completed the evaluation. At the beginning of the survey, the participants were introduced to the key concepts of the video game. Afterwards, the users were asked to watch a brief video (5:30 minutes) that introduces the participants to the web application. The next step covered 11 different data analysis tasks which are assigned to a certain category where participants should use the tool to get some information on the student's or the group's progress. Each category was completed by answering five NASA-TLX questions regarding workload. After all tasks in the learning platform, the participants additionally completed the System Usability Scale. The questionnaire did not query any personal information and thus is fully anonymous.

The SUS data showed that there might be some potential outliers. They were detected using standard deviation method. Due to the number of participants we decided to choose a less conservative approach where two standard deviations (Leys et al., 2013) from the mean were used as cut-off. Using this approach three outliers could be detected which were removed from the observations of the system usability (see Figure 3.6).

The participants from the expert's evaluation are mainly vocational school teacher's. The evaluation was conducted unsupervised and online. In total 14 teachers were contacted, and 8 people completed the questionnaire. The questionnaire consisted of four parts: i) demographic and professional-related questions, ii) general questions about learning analytics tools, iii) tasks on the learning analytics platform, and iv) post questions about the system. All teachers were familiar with the serious game sCool, but an additional video was provided within the survey. They did not get any introduction to the learning analytics platform to avoid any advantages regarding the system's usage.

All data from both questionnaires was analyzed using the Python programming language and the Data Analysis library Pandas. All categorical data was coded and analyzed using basic statistic techniques such as calculating mean and standard deviation. Numerical or textual data from the tasks was manually evaluated and assigned to Boolean values (true/false) depending on the provided answer. All qualitative data such as open-ended questions were evaluated manually and grouped into different categories.

3.2.5 Results and Discussion

Usability. The usability evaluation conducted with the SUS showed a well-rated usability. The SUS value is 79.02 (SD=10.69) which is rated as A- (Lewis & Sauro, 2018) and thus, means well satisfying usability. Figure 3.6 depicts the SUS results, including even three outliers (P25, P28, and P29) which are not considered in the calculation of the overall SUS value. The mean Raw NASA-TLX (RTLX) (S. Miller, 2001; Moroney et al., 1992) score is 2.38 (SD=0.46) on a scale of 1 to 5 (very low to very high), implying low to moderate perceived workload.

When reviewing the open-ended questions the participants mentioned that the system gives comprehensive functionalities, but they had issues dealing with this complexity. They desire a simpler and more intuitive user interface that is self-explaining. This evaluation is similar to the results of the teacher's evaluation. However, study 2 did not focus on usability aspects in particular, but using the open-ended questions the majority of the participants had some issues solving tasks due to usability reasons.

Further, the evaluations showed that the user experience and usability highly depend on the considered group and the tasks (Morris & Dillon, 1997; Renz et al., 2014). Study 1 was conducted with participants that are highly familiar with interactive systems and data science. They seemed to be more familiar with the systems architecture. On the other hand teachers (study 2) commented that the system is rather complex and confusing at first glance. Another reason for the different evaluation results could have to do with the design of the tasks. While the first study's aim was to evaluate the system's usability, we focused on learning analytics aspects on the expert's evaluation. However, since many teachers had issues with the tasks (see Table 3.8) the user interface and system design should be reconsidered in future versions.

Analysis and Visualization. Each teacher responds that he or she wants to use a learning analytics tool with a learner-oriented approach. They want to keep track of the students learning progress and possible issues. Table 3.8 gives an overview of the results of the evaluation. The number of correct and wrong tasks revealed that some sections of the application are understandable and can be used for a meaningful analysis. This mainly concerns the tasks related to the student's sections (Tasks T2a/b, T3, and T4). Due to some ambiguous wrong answers the system might provide too many possibilities for analytics. The fact that just one person was able to complete Task 5 (and the high rating of the difficulty) showed, that the custom data visualization needs to be optimized in terms of the system's usability and user interface.

3.2.6 Conclusion and Future Work

In this paper, we presented an approach for a learning analytics environment for a serious game in computer science education. The goal was to design and evaluate the prototype. The results showed that there is a high demand for learning analytics platforms to support both students and teachers (Chaudy et al., 2014;

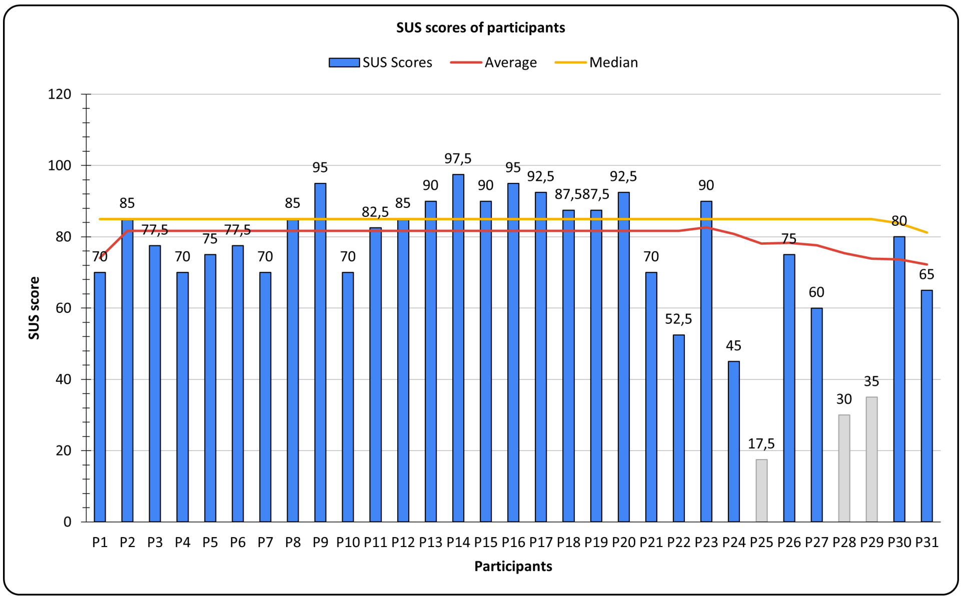

Figure 3.6: In total 31 participants (P) respond to the System Usability Scale. The mean system usability scale value is 79.02 (SD=10.69), which is considered to be a good grade in terms of usability. It is in the percentile 85 - 89 which is graded as A- within the scope of SUS Lewis and Sauro, 2018. The participants P25, P28, and P29 were identified as outliers and therefore not included in the calculation.

Hauge et al., 2014; Romero & Ventura, 2020). The design of such a platform has to fit the educators' needs. We encountered several aspects that are important in the design of a platform:

Lightweight Design. One key element of a learning analytics platform is a lightweight design that is well structured and easy to use. The tool should have a flat hierarchy that can be easily navigated. The provided information should just compromise the most important data.

Easy Usage. Strongly linked to a lightweight design is a simple usage of the system. Teachers want to have easy navigation and easily receive all data. The system should have a good balance between comprehensive data analysis and intuitive usage. The teachers responded, that they enjoyed using the system after they understood its depth. An on-boarding via an interactive tutorial or expressive labels/tooltips can improve usability and can have considerable benefits over explanation videos or manuals (Davis, 1989; S. et al., 2018; Schneider, 2016).

Learning Progress. Obviously, the tool should provide meaningful data over the students learning experience and progress. The system should be open for both students and educators in order to make the results as transparent as possible. Custom data analysis and visualization are perceived as a good possibility for individual support measures. In addition, the system should also help teachers to grade. Therefore, a flexible assessment and evaluation should be provided that can be exported easily.

Privacy. Teachers are hardly concerned about their student's privacy. They like to use systems that are already used in other educational institutions and are GDPR conform. Since data about children are classified as particularly worthy of protection (Slade & Prinsloo, 2013), the transmitted data should contain as little personal data as possible. A good approach would be the usage of pseudonyms where just teachers can identify certain students (especially for individual guidance) (Freire et al., 2016).

Future Work. For further improvements we want to provide player's data on their learning progress. Learners should get access to a dashboard where they get comprehensive information about their performance within the learning environment. On the one hand, this should cover game-related information such as passed levels or points and on the other hand, they should see which concepts and skills they already acquired and which are left. Predicting student's knowledge gain, classifying students based on similar behavior and requirements, and the ability to analyze a large number of students with easy-to-understand reporting could also be considered (Hauge et al., 2014; Verbert et al., 2013).

The results showed, that teacher's demand for usability improvements. After the improvement of the mentioned usability issues, further studies should be conducted on a larger population. The focus of this further observation will be the interaction with the system to identify workflows and features. Therefore, web analytics tools such as Matomo will be used to track user's behaviors.

3.3 Article 3: Collaborative Learning in a Serious Game

Motivation

Multiplayer game modes are important in serious games because they foster collaboration, communication, and social interaction among players, which enhances the learning experience. By engaging in multiplayer interactions, players can practice problem-solving skills, teamwork, and decision-making in a dynamic and realistic environment, making the serious game more effective in achieving its educational objectives (Malliarakis et al., 2013; Wendel et al., 2013b). In previous studies with the serious game sCool, students provided the feedback that a multiplayer game type might be considered to work on problems and learn together collaboratively.

Contribution

Collaborative learning approaches such as pair programming are frequently used in computer science education. Various studies in pair programming indicate that it leads to more success, higher satisfaction, and enjoyment (Simon et al., 2019). Due to the positive aspects of collaborative learning, a concept for a multiplayer game mode was developed and implemented. Within a study, three interventions have been conducted in both formal school settings and informal online contexts. The study aimed to observe how students interact and communicate by just using game components. Since the students collaboratively worked on different programming tasks the team's performance was analyzed additionally.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - Several studies emphasize the benefits of collaborative learning, especially in programming education. The existing sCool game does not support a multiplayer feature. In addition, students respond in previous interventions, that a collaborative effort in sCool could increase their engagement and motivation.
- **Theoretical Foundation** - A literature survey was conducted to review the current state-of-the-art literature in the field of collaborative multiplayer learning in game-based approaches and also regarding collaboration in programming education. The goal of the literature survey was to investigate design principles for implementing a collaborative scenario for programming.
- **Theoretical Concept** - The findings from the literature review led to the development of a conceptual model.
- **Implementation** - Based on this model a multiplayer game type was implemented. This game type was integrated into the existing game. A central aspect of this study was to create a meaningful learning context where players

have to work collaboratively on a problem.

- **Evaluation** - The multiplayer game type and the learning approach were evaluated in a formal school context but also in an informal online context.
- **Validation and Best Practices** - Based on the experiences from both evaluation contexts, interesting conclusions could be made. These findings helped to draw implications on how collaborative learning can be implemented within a serious game.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Formal analysis, Writing - Original Draft, Writing - Review, Editing
- **David Eckhard:** Software
- **Julius Dreveny:** Investigation, Resources, Writing - Original Draft
- **Christian Gütl:** Supervision

Steinmaurer A., Eckhard D., Dreveny J. & Gütl C. (2022). Developing and Evaluating a Multiplayer Game Mode in a Programming Learning Environment. In: Proceedings of the 8th International Conference of the Immersive Learning Research Network (iLRN), 2022, pp. 1-8.

Developing and Evaluating a Multiplayer Game Mode in a Programming Learning Environment

Alexander Steinmaurer, David Eckhard, Julius Dreveny & Christian Gütl

Abstract

Serious games have an already well-known positive impact on students' motivation and their learning experience. In computer science education a majority of games and approaches exist, that provide engaging environments for students. On the one hand, these can be games related to learning algorithmic thinking but on the other hand, games to learn coding. Besides single-player games, there are also multiplayer games where players compete against each other or collaboratively work on programming tasks. However, many existing multiplayer games offer pre-defined levels where teachers have limited flexibility and individuality for their students. Additionally, many existing game environments use competitive over collaborative approaches. In our project, we extended an existing game by a multiplayer mode where players work together on coding tasks. The game elements support many coding-related concepts but also computational skills such as sequencing, conditionals, loops, and also advanced topics such as concurrency or dependencies using meaningful levels and tasks. However, we conducted an evaluation including 41 participants in three workshops: two in-class activities with secondary school students and an online activity with computer science students. Within these workshops, the students collaboratively worked on coding tasks within the game environment. Thereby, we observed the communication between the students while working on the tasks. Additionally, we evaluated the students' attitudes towards collaborative learning. We found out that the in-game chat is barely used while in-class activities, especially when the tasks require only low coordination between players. We also found out, that students learn from each other and are more motivated when working together. With our approach, we want to provide educators with a flexible game environment where students can collaboratively improve their coding skills while solving engaging tasks.

3.3.1 Introduction

In educational and professional settings a large number of collaborative approaches for coding exists such as programming workshops, Hackathons, or pair programming. These methods can increase the skills, creativity, productivity, and diversity of development teams. Specifically, collaborative programming is beneficial for beginners but also for experts to share experience and get insights about other perspectives. Working together has a long and important tradition in human culture. When working collaboratively people are more interested in solving tasks, achieving a better performance, and are more persistent in challenging tasks (Carr & Walton, 2014). Social aspects such as motivation, or common goals are also relevant in computer games. First multiplayer games appeared already in the 1970s, as arcade games were played on arcade machines. The way people play computer games with each other has changed significantly over the past decades. The evolution of multiplayer goes from playing simultaneously on the same computer or console via a split-screen, or asynchronous in a *hot seat* mode to games over networks where each player has their own device. In multiplayer games, a basic distinction is made between competitive (two or more players compete against each other) and cooperative (two or more players aim to reach a goal together) (R. T. Johnson et al., 1986).

In addition to the entertainment factor, computer games are also used in education as game-based learning. Various serious games in many fields try to engage and promote students in the corresponding topics. Similar to entertaining games, a large number of multiplayer games exist in the area of education as well. Even though cooperative methods tend to have a greater effect on learning achievements (Creighton & Szymkowiak, 2014) and the motivation (W. Peng & Hsieh, 2012), a large number of tools exist that follow a competitive approach. Many authors point out the promising opportunities of collaborative educational games (Wendel et al., 2013a), not only to teach subject-related topics but also to increase the player's social skills.

In this research, we introduce a collaborative learning approach for programming tasks and challenges in a game-based environment. Therefore, an existing game was extended by a multiplayer game mode. Prior research showed that there is a need for different educational contexts within the environment such as in-person or online modes. Therefore, we focused on a simple interaction between players in the system. To gain a better understanding of the game-based approach we conducted an evaluation with three groups. The following research objectives were defined in the scope of this study:

- How do students communicate in teams while collaboratively solving a programming task in the multiplayer game mode?
- What are the opportunities and challenges related to the player's interactions

within the game?

- How does a multiplayer game affect the student's attitude towards collaborative learning?

The structure of all evaluations was equal, but the group of participants was different. Two evaluations took place as supervised workshops in school classes (3rd and 7th grade) and one was an unsupervised online study with computer science students. This research aims to evaluate a game-based learning approach and show its possibilities in education. Within this project, a central contribution is the design and implementation of the multiplayer game mode for the existing sCool environment. Additionally, we want to illustrate possibilities for an engaging approach to learn programming collaboratively.

This paper is structured as follows: Section 3.3 gives an overview of engaging collaborative learning approaches and tools. In Section 3.3 the sCool game environment is presented. Chapter 4 covers the three evaluations that were conducted within the project and in chapter 5 the results are presented and discussed. Finally, chapter 6 concludes the entire project.

3.3.2 Background and Related Work

Learning to program is a complex process for beginners. It requires learning not just a language but also specific ways of thinking. Over the last decades, many different educational approaches have been developed to help novices. These aspects of programming learning are covered extensively in literature (Luxton-Reilly et al., 2018; Vihavainen et al., 2014). The related research covers pedagogical models, discussions about programming languages for beginners, engaging tools and learning environments, and approaches of content delivery.

One central theory in the context of collaborative learning is the social cognitive theory (Bandura et al., 1999). According to Bandura people learn based on their own experiences and by observing the experience of other persons. When other people are observed a person tends to reproduce the learned behaviour. This type of collaborative learning is also relevant in introductory courses for programming (Azmi et al., 2015). When learning to program collaborative learning also indicates several advantages covering increased motivation, a better understanding of content knowledge, and improving soft skills. There are different approaches for collaborative programming learning such as peer review, pair programming or group activities (Luxton-Reilly et al., 2018). These approaches also include the usage of engaging technologies such as serious games. Therefore, tools such as

LightBot or *Robocode* are applied in education. These games are generally intended to be single-player games. Students can collaboratively work together on-site on tasks within the games in a pair programming setting (Silva et al., 2020a).

There are also games where the game mechanic is dedicated to a multiplayer setting. An educational game where two players collaboratively work on a problem is *Pyrus* (Shi et al., 2019). The system's web interface displays a problem and provides users with an editor to submit code. The game is round-based which means that each player has a turn where he or she can contribute to the solution. The evaluation showed that the participants spent most of the time on planning and organizing than on coding. Furthermore, the players fail on effective collaboration, since they are not familiar with the process of pair programming.

An important element in computer-supported collaborative learning is the communication between team members. Fast-paced communication technology such as chats promotes instant communication. Related to this aspect an explicit goal is vital to have a clear aim for all team members (Knutas, 2016). Text messages in a chat could have an impact on programming tasks, since both use text. Therefore, an audio support using a microphone can be used instead (Silva et al., 2020b). Another key aspect in collaborative learning environments is a well-structured educational concept. Educators should provide students with clear goals, appropriate resources and feedback (Azmi et al., 2015).

Games such as *CodeWars* are intended to be competitive games where players work on programming tasks simultaneously. While solving the tasks there is no interaction between the users. A similar platform is *CodinGame* where players are challenged to solve tasks. Again, there is no interaction between players during the tasks, but the platform provides an asynchronous communication where players can discuss solutions after a challenge (Heller & Mader, 2021).

Over the last years the number of research in the field of immersive learning environments in computer science education has increased. Pirker et al. (Pirker et al., 2020) conducted a literature survey which gives an overview of various immersive approaches in computer science education. They also name different advantages related to immersive learning such as meaningful visualization to make complex topics (such as algorithms) more understandable, or social experiences related to collaboratively learning to program. One immersive video game is *FunPlogs* (Horst et al., 2019) where players create and solve puzzle-like programming tasks. The game consists of two scenes, a building and a scripting scene. Within the building scene students are collaboratively creating levels together, where one person is using a Desktop view and the other person is wearing a VR device.

Another game-based approach related to programming is the serious game *sCool* (Steinmaurer et al., 2019). The game's purpose is to engage students in coding and to increase their computational thinking skills. sCool consists of a web platform where educators can manage courses and players and the game itself. The web tool provided teachers with many features such as defining tasks, managing players, creating game maps, preparing in-game questionnaires, or doing learning analytics. All data from the web tool is transmitted to the video game, which provides a flexible and individual learning experience. The game is divided into a concept-learning part and a practical part. The concept-learning part introduces basic concepts of coding and computer science. Within the practical part, the players have to apply these concepts in coding tasks using the Python programming language. Since this game did not provide a dedicated multiplayer mode so far, the next chapter describes its design and development.

3.3.3 sCool Multiplayer Game Mode

While sCool provided the player with different gaming possibilities for the single-player mode, there was no way to play and solve puzzles together. Therefore, we designed and developed a multiplayer mode for the practical programming part of the game - also known as robot missions. Here, players are given the task to navigate a robot through a map and reach a certain goal while avoiding obstacles. This has to be achieved by writing code in the Python programming language (Steinmaurer et al., 2019). Based on previous experiences and insights of the literature survey, the following requirements have been defined to provide a multiplayer mode:

- Integrate game elements to provide collaboration possibilities
- Allow educators to create multiplayer maps
- Provide real-time communication for players
- Creating and joining lobbies

Collaborative Game Elements

To encourage collaboration between players, new elements have been included in the existing game. With these new elements, it is possible for course designers to create challenges in which players rely on each other to advance in the level.

Doors and Triggers: For the first mechanic, a simple door was created. To open this door, a trigger has to be activated. This trigger is represented as a platform that can be located on an empty field on the map and must be activated by moving over it. The door and trigger have the same color as well as a line between them, to better understand the connection of these elements (see Fig. 3.7). This feature is a great asset for multiplayer and can also be used in single-player mode. In the online custom map creator, doors and triggers can be placed individually. When placing either of them, a number indicating the index will also show up. Doors and triggers with the same index belong together (see Fig. 3.8).

Wait and Signal: For the second collaboration element, the concept of threads in computer science was taken into account. Just like threads, the robots are able to work through their tasks concurrently. There might be a point where one thread has to be sure that another thread has completed a certain task, before moving on. A thread can therefore wait for a certain signal before resuming work. The same behaviour has been implemented for the robot missions. Players are able to wait for a certain signal until they resume their execution. Right now, there are two use cases for this: A certain path might lead both robots to get to the same field at the same time. This would lead to an undesired crash and a reset. One player would have to wait while the other can safely pass and then send a signal to wake the other up. The second use case would be to use it in combination with the door and trigger. These elements can be placed on the map in such a way, that requires one player to wait until the other robot activates the trigger and sends a signal that it is safe to move on (see Fig. 3.7).

The existing web interface allows educators to create individual maps by placing all available game elements on a 2D grid. The number of players is implicitly defined by the number of robots that are placed on the map.

Communication

Communication and interaction between players are vital parts of a collaborative multiplayer environment. The design of the gameplay requires communication between players to discuss strategies to solve the given puzzle. These strategies cover plans about dividing the tasks within a map, deciding on routes to avoid crashes, or which signals to use to share resources. Furthermore, communication can be used to ask for help and give guidance at a certain level. Players are also able to just chat with course colleagues using plain text messages. While all of these actions can be achieved through one channel, we decided to provide different channels. The first includes the whole course. Every player within a course can use it and will receive messages sent through it. The other channel is lobby-specific. Only players within a mission lobby can send and receive messages. A more private situation like this may encourage players to communicate with their partners in the

Figure 3.7: This map shows the field for a two-player level. This mission requires coordination between the two players to activate the triggers to open doors.

lobby instead of all the participants in the course.

Architecture and Functionality

Fig. 3.9 shows the system's architecture with all relevant components. On the one side, there is the game client. This allows a user to start the game which will communicate with the server. On the other side is the server. It will respond to requests made by the game. The server itself is running two containers - one for the relational database and the other for the Laravel applications. While the MySQL database provides persistent storage for data, the applications provide the functionality and logic to access, transform and return data. They do so by defining certain endpoints which can be accessed.

An usual approach for a distributed architecture is that the client sends a request and receives a response from the server (which is utilized in the general application on the server), but we decided to use WebSockets for the multiplayer communication. Once a WebSocket connection is established, it allows for bidirectional data transfer without the overhead of continuously sending requests. This means that the client and the server can send messages without the other explicitly requesting it. This allows providing a scalable and lightweight system.

Each course has its own endpoint and consists of several missions. For each mission, users are able to create lobbies on demand. A lobby holds information on current

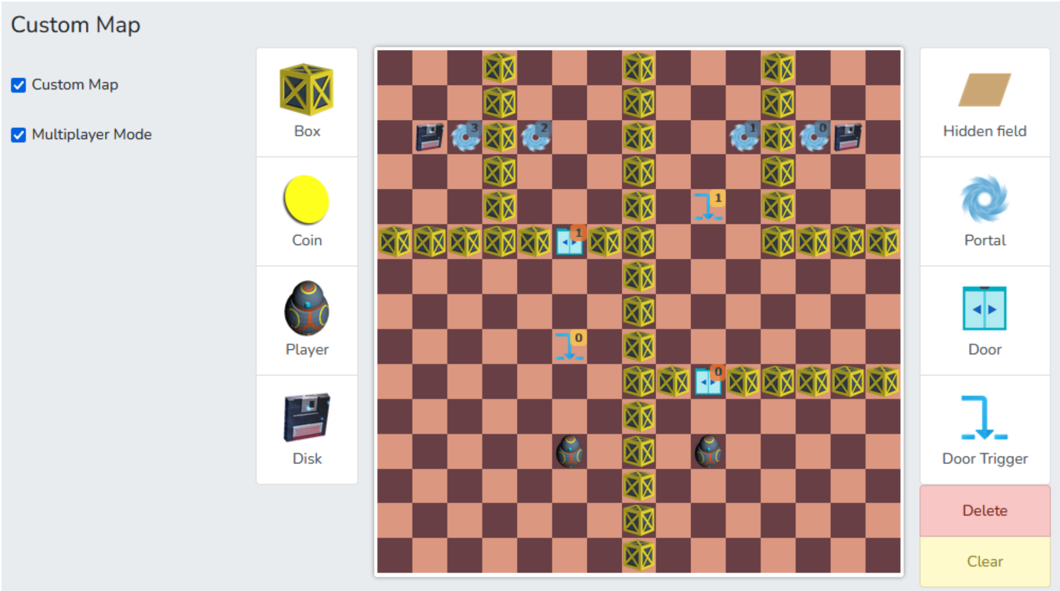

Figure 3.8: Example for a map creation in the teacher's web app.

players, their code, and spectators. A user may choose to join as a player controlling a robot or as a spectator, simply watching others and being able to chat with them. When a player is done writing code for a mission, the code will be run locally and checked for errors. If there are no errors, the code is sent to the server and broadcast to other players and spectators in the same lobby, after having received all code snippets. This will then trigger the game to execute the code of each robot and start the movement.

All of the communication functionality for multiplayer mode was done with different message types such as joining or leaving lobbies, code updates, or regular chat messages. This was realized with a so-called *Publisher-Subscriber pattern*. Here, the incoming messages from the server (publisher) are distributed to all the components (subscribers), which are interested in getting updates about these messages. As an example, when a player joins a lobby the chat component will display a message, while the component for the robot mission will initialize the robot. All of these actions happen independent of each other. Integrating new features and/or new message types may also be done without interfering with the current functionality.

Exploring Possible Learning Experiences

sCool's multiplayer mode opens up additional learning expenses. Using the web tool's map generator, the missions can have a high level of freedom. The game elements are expanded by interactable elements such as triggers, portals, or coins

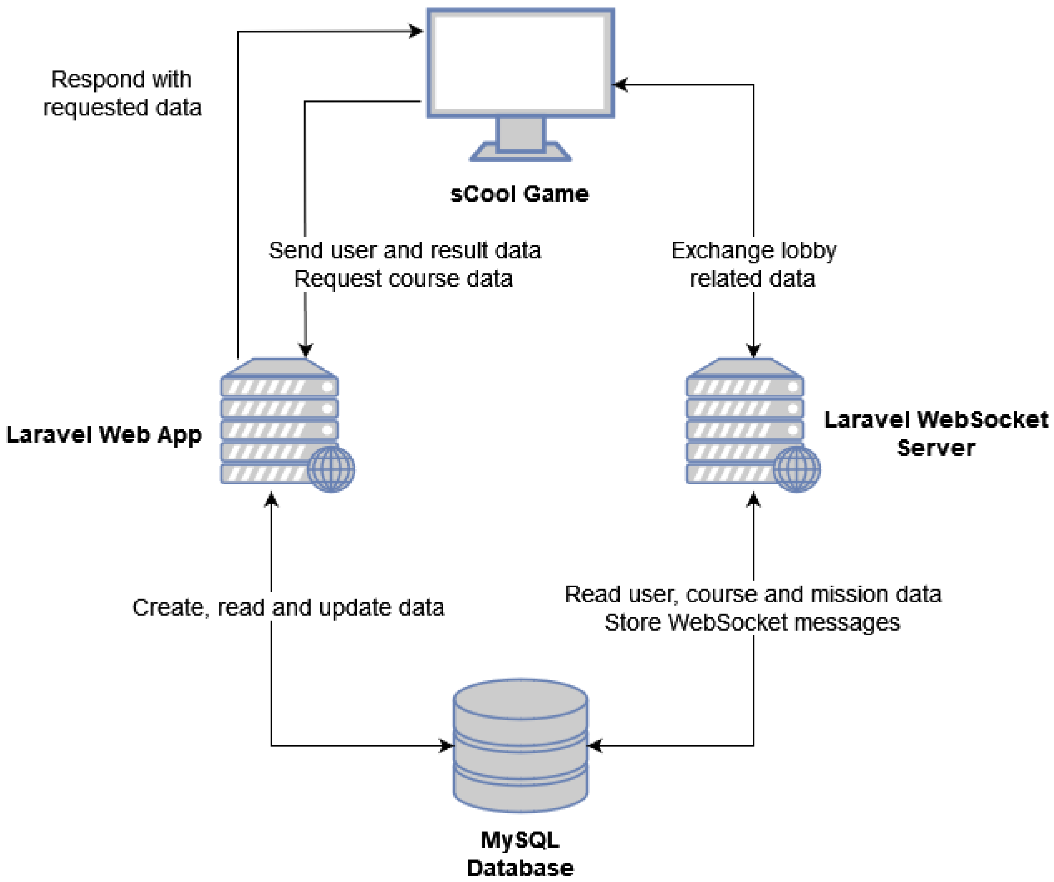

Figure 3.9: This figure shows the architecture of the sCool environment.

but also non-interactable objects such as doors or boxes. Certain learning objectives can be realized quite easily through appropriate level design. By adding elements such as doors and triggers, or portals, the players can be encouraged to highly collaborate to solve a level. They have to coordinate themselves and their team members as well. This also requires a good understanding of the programming environment and increases computational thinking skills (see Fig. 3.10).

Fig. 3.7 illustrates a possible level design where collaboration between both players is absolutely required to reach the goal. Each player has to use the trigger (floor plate) to open the door and let the other player pass. Additionally, they have to coordinate about the correct timing, since pushing the trigger depends on an open door. Therefore, the players can apply the wait/signal concept. The idea of this

Figure 3.10: The map creation allows educators to define levels with different levels of complexity which opens up a wide range of educational possibilities.

feature is to illustrate synchronization between players.

3.3.4 Evaluation

To evaluate the multiplayer game mode we conducted three evaluations. Two evaluations took place within a school class and the third as an online activity due to the COVID-19 pandemic situation with computer science students at the university. We were interested in evaluating the interaction and communication between the players and the student's attitude towards collaborative learning of the game type.

Participants and Recruiting

Overall 41 people participated in this study. To receive a wide range of opinions the same evaluation was conducted with three different groups: a 3rd-grade secondary school (group A), a 7th-grade secondary school (group B) and university students studying computer science (group C).

With reference to diverse feedback, the project team attached importance to the selection of the participant groups. For the in-class activities, two partner schools of the research group were invited to participate in the study. One of the school classes (group A) is from a rural area, while the other class is an urban school (group B). The recruiting of the participants for the online study (group C) happened within the scope of a computer science introductory class. An email was sent to all students of the course and 9 have agreed to participate in the study.

Group A consisted of 19 students (4 female, 15 male) with an age range between 12 and 15 (M=13.36, Sd=0.76). In group B 13 students (7 female, 6 male) attended the evaluation, aging between 17 and 19 (M=17.61, Sd=0.77). Group C consisted of 9 students (all male) in the field of computer science (Bachelor, Master, and PhD students). They were between 21 and 32 years old (M=26.77, Sd=3.38).

Materials and Methods

At the end of the evaluation, the participants were asked to fill out a survey with 24 items in total. The answers were collected using an in-game questionnaire that is integrated into the game's user interface. This approach allows asking questions without interrupting the player by changing the system.

The survey consisted of four parts: i) personal information (age and gender), ii) collaborative learning, iii) system interaction, and iv) general feedback. The questions related to collaborative learning are a modified version of Driver's questionnaire (Driver, 2002) including 11 questions (see Table 3.9). The items' overall reliability was assessed by calculating Cronbach's alpha (α=0.79). Thus, the value indicates a satisfactory internal consistency. The questions related to the system interaction are six statements on a Likert scale ranging from 1 (strongly agree) to 5 (strongly disagree). Due to the specific classroom situation of this study, the six questions of this questionnaire have been created by the authors to receive insights about the student's interaction with each other. When testing for internal consistency for this questionnaire the Cronbach's alpha has a value of α=0.6. Finally, the participants could answer three open-ended questions as general feedback covering aspects they perceived as positive and negative and give room for suggestions.

When starting the sCool video game the participants were informed that their data will be collected and analyzed for a scientific purpose. The players can decide if they want to confirm. In case they refuse the data collection and analysis they can play the game without any restrictions. Besides the participant's age and gender, no personal information was collected. For this reason, it is not possible to link an answer to a participant which means all data is anonymous. One person of the project team who was not involved in the data collection was responsible for the data analysis to keep this process as objective as possible. The data was analyzed using the R programming environment.

Table 3.9: Questions Related to Interaction (I1 - I6) and to Collaboration (C1 - C11).

#	Question	Mean	SD
I1	I mostly used the chat for communicating with others.	3.05	1.43
I2	I also communicated with my partners by talking.	2.14	1.36
I3	The communication via the chat was easy.	2.27	0.83
I4	It was easy to discuss possible solutions via the chat.	2.77	0.92
I5	The handling of the chat was easy.	2.73	0.94
I6	It was fun being in a group with people, I would usually not be together.	2.14	1.08
C1	I communicated often with my partners.	2.41	1.22
C2	I always felt like being a team with my partners.	2.36	1.18
C3	I often discussed possible solutions with my partners.	3.00	1.31
C4	Exchanging possible solutions with my partners was fun.	2.23	1.02
C5	I had the impression that the players of a course saw themselves as a team.	2.82	1.14
C6	The discussions I had with my partners was important for understanding the solutions.	2.64	1.22
C7	I learned a lot from others during the game.	2.73	1.20
C8	I see a benefit in what I have learned.	2.59	1.01
C9	I believe that I know my partners better now.	3.41	1.26
C10	Solving tasks in a collaborative way is more motivating than solving tasks alone.	1.86	0.77
C11	Generally, I prefer to solve tasks alone.	2.91	1.06

Procedure

Even though the mode (in-class and online) of the workshops slightly differed from each other, the procedure was the same for all of them. All interventions took a total time of maximum 50 minutes. First, all participants received an introduction about the evaluation and the game. In both in-class activities, this explanation was held by a member of the project team. At the online workshop, the participants received all necessary information via a YouTube playlist that was prepared by the project team again.

After the software installation, all remaining activities had to be done in the game which took about 40 minutes. After starting the game all participants had to go through the onboarding level to make themselves familiar with the user interface and the gameplay. Afterward, nine levels were provided: four basic levels with less collaborative purpose, two intermediate missions that required collaboration between the players and finally three advanced levels where a comprehensive coordination between the players was required. Besides the task numbers and categories Table 3.10 contains a list of all game elements that have been enabled for the specific levels. The column *Intensity of Interaction* defines to what extend interaction between the players was necessary to solve a task. The range is from *None*, which means that no interaction is necessary, over *Little* and *Medium* which means the players have to slightly coordinate their actions (within a few messages) to *High* which hardly requires a comprehensive strategy and cooperation to solve a task. All levels were designed for two players. The game mechanism allows to create a new game lobby or join one. As soon as two players are present in a lobby the game starts. The players are not aware of their partners in a game and get randomly assigned to a person to force them to mainly use the game chat for communication.

The players were asked to play at least the four basic and two intermediate levels. Afterward, the participants filled out the questionnaire about interaction and collaboration that was integrated into the game to encourage students to complete the questionnaire without being distracted by a change of platform.

Table 3.10: Game Missions and Game Elements of the Study

Task	Category	Game Elements	Intensity of Interaction
#1	Basic	Boxes, Coins, Disks	None
#2	Basic	Boxes, Coins, Disks, Portals	Little
#3	Basic	Boxes, Coins, Disks	None
#4	Basic	Boxes, Doors/Triggers, Disks	Little
#5	Intermediate	Boxes, Disks, Wait/Signal	Medium
#6	Intermediate	Boxes, Disks, Wait/Signal, Doors/Triggers, Portals	High
#7	Advanced	Boxes, Coins, Disks, Wait/Signal, Doors/Triggers, Portals	High
#8	Advanced	Boxes, Coins, Disks, Wait/Signal, Doors/Triggers, Portals	High
#9	Advanced	Boxes, Coins, Disks, Wait/Signal, Doors/Triggers, Portals	High

3.3.5 Findings and Discussion

Collaboration

The aim of this study was to let teams of two players work collaboratively on programming tasks. Three different groups of levels were offered, whereas the necessity for coordination between the players is increasing. The response of question C1 illustrates that communication was hardly required for the levels (M=2.41, Sd=1.22).

Obviously, the majority of the participants enjoyed the multiplayer mode, since they mainly agreed, that this is more motivating than working alone (M=1.86, Sd=0.77). In the open-ended questions, 23 out of 41 participants (56%) additionally mentioned that they consider the multiplayer mode to be motivating and promising: *"[I liked] the possibility to exchange within the chat [and] that I had to trust the other blindly if he did not answer in the chat. That was exciting!"* (group A). When observing the suggestions for improvements participants mentioned, that waiting for partners might take a while: *"you have to wait till the partner also executes the code"* (group A).

When it comes to collaboratively working on a problem, there is a noticeable correlation (d=0.75, p=0.000) between variables C1 (*"I communicated often with my partners"*) and C2 (*"I always felt like being a team with my partners"*). This indicates that participants that regularly communicated with each other during a mission felt stronger as part of the team. Another noticeable correlation (d=0.70, p=0.000) is between the items C2 and C9 (*"I believe that I know my partners better now"*) which also emphasized on the positive effect of communicating within a team. The feeling of being part of a team (C5) has also a strong positive correlation (d=0.61, p=0.003) with fun when exchanging solutions in a team (C4).

When looking at feedback regarding the improvements, the responses from the school interventions (groups A and B) mainly suggest including game elements such as skins, items, or enemies. The results from the online evaluation (group C) compromises usability features (hide chat or faster onboarding level) and additional functionality (speed mode, voice chat, or code completion).

Chat Interaction

During all three evaluations, 370 messages were sent within the sCool in-game chat. The participants of groups A and B were located in the same room since it was planned as an in-class activity. The instructors told them that they can use the chat, but also communicate within the classroom. The game is structured in a way that players got randomly assigned to another partner in the next round since players do not see which player is in which game. This makes in-class interaction more inconvenient. In both groups the players used the chat mainly for entertaining purposes: 130 messages were sent in total and most of the conversation was not related to game activities. Since the communication in group C was mainly limited to an online mode it was used more often. The correlation (d=0.73, p=0.000) between the items C1 and I3 (*"The communication via the chat was easy"*) also shows that participants are more likely to communicate with their partners via chat when they are able to handle the chat better. Overall 240 messages were transmitted in this group. The different usage of the chat also shows when comparing the mean values of question I2 (*"I also communicated with my partners by talking"*) for each group. Due to the in-class setting of groups A (M=2.57, Sd=1.55) and B (M=1.54, Sd=0.52) a lot of interaction took place in class, while in group C (M=3.00, Sd=1.73) more interaction took place using the chat. But the results of group C also shows that there was probably an additional channel for communication such as two fellow students working on the task together. The following chat snippet shows an exemplary conversation between two players:

- Player A: *"You go first, and I will wait."*
- Player A: *"And you are sending me a signal than?"*
- Player B: *"I try..."*
- Player A: *"Which signal name? ESP? :D"*
- Player B: *"Amen"*
- Player B: *"alright"*
- Player A: *"Nice :D"*
- Player B: *"well done :D"*

When observing the suggestions for improvements some participants mentioned, that their partners did not respond on chat messages: *"that the probability of getting an answer from my team partner in the chat was 50/50"* (group B).

One aspect related to chat and interaction is the approach that students are randomly assigned to other players. The idea was to shuffle team members to have diversity in the group constellations. Players should benefit from various opinions and solutions. Question I6 was related to the students' acceptability about random groups. They respond that it was fun for them to play with people they would usually not work together (M=2.14, Sd=1.08). Remarkably, the online group, in particular, noted that they sometimes had difficulties with the operation of the chat window, as it sometimes overlaps the game. Since approximately 65% (240 out of 370 messages) of the total chat messages came from Group C, there is more feedback from this group related to the chat. Although the chat was noted almost equally positively and negatively, some of the participants indicated in the feedback that they would like to see a voice chat. Since the coding activities are already text-based a chat might have a negative impact on the collaboration. Silvia et al. (Silva et al., 2020a) made similar observations and suggest microphone support as well.

The design of the levels hardly influences the need for communication. The first half of the levels were mainly intended to have no direct need for communication, and the second half required more coordination of the participants.

3.3.6 Conclusion and Future Work

In this paper, we have introduced a multiplayer mode for the serious game sCool. The game mode's objective is to combine the advantages of collaborative multiplayer gaming with an engaging learning environment to increase the students' engagement and skills in programming. Furthermore, we have conducted evaluations with 41 participants in three groups, two in-class activities with school students and one online activity with computer science students. The results showed that the collaborative learning approach of the game is motivating. The in-game interaction between the players, especially in in-class settings, shows room for improvement. Due to the game mechanics and the duration of each level a dedicated voice communication seems to be inappropriate.

Limitations

The selection of the participants was based on a diverse group of people, to receive various feedback. Each group consisted of a number between 9 and 19 participants, whereby the mode of the intervention was also different (in-class and online activity). In this way, it is not possible to make significant assumptions related to group-specific features such as performance, interaction, etc. Therefore, we plan to run further evaluations with a larger population of each group and similar conditions. This allows having a study design where comparisons between groups can be measured to gain further insights into the learning experience of the groups.

Future Work

In a future version of the game, we will also consider the participants' feedback for an improved game. The chat interaction will be optimized to have on the one side a lightweight user interface and on the other side more expressive interaction for instance arrows to point to specific areas within the game. We will also include further game elements that emphasize collaborative learning such as pressure plates, or additional signals.

The focus of this evaluation was to evaluate the novel game mode regarding interaction and the participants' attitude towards collaborative learning. In the next iteration, we plan to run evaluations that go beyond the scope of single workshops, to investigate long-term effects. This extended study design should give further insights into the students' performance related to the collaborative game mode.

3.4 Article 4: An Online Platform for Security Awareness

Motivation
The DigiSkill platform is based on several conversations with CS teachers about the lack of engaging online learning platforms to teach digital competencies. The idea was to create a concept for a learning environment where educators receive pre-defined learning scenarios but also have the ability to modify existing ones. The platform should be highly modular to easily include other topics and functionalities as well. A seamless integration of all modules into an entire course using a story-based approach was emphasized. Additionally, the platform should deal with the students' different performances and enable individual learning experiences.

Contribution
The fundamental idea behind the *DigiSkill* platform was to provide educators with a flexible web-based system for teaching digital competencies. Therefore, a modular system was developed that emphasizes the development of additional modules related to digital skills. Exemplary three modules were implemented, including spreadsheets, data literacy, and security awareness. The platform follows a story-based approach using real-life examples and a non-linear learning path for an individual learning process. For this reason, an educational concept was developed and delivered using DigiSkill in the field of security awareness. An in-class intervention with 52 secondary school students was conducted. This study aimed to observe the effectiveness of the tool but also how this approach can engage students in security awareness topics.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - A research gap was discovered based on various conversations with computer science teachers. They mentioned, that there is a need for an engaging learning approach to cover digital competencies in computer science education.
- **Theoretical Foundation** - A literature survey was conducted to identify related work, approaches, and concepts of how the learning platform can be developed to cover all relevant requirements.
- **Theoretical Concept** - The findings of the literature survey and the teachers' input formed the theoretical concept for the DigiSkill tool.
- **Implementation** - The software was developed and in parallel, learning approaches have been designed for in-class usage of the tool within a meaningful educational context. The focus of this study was to include the topic of IT security awareness.

- **Evaluation** - Within school interventions, several aspects regarding the tool and the learning scenario, such as engagement, interest, and performance, were evaluated.
- **Validation and Best Practices** - The results of this evaluation were the base for further conclusions regarding the design of learning platforms for digital competencies but also how they can be integrated into educational scenarios.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Formal analysis, Writing - Original Draft, Writing - Review, Editing
- **Azra Bajramovic:** Writing - Original Draft, Validation
- **Daniel Pollhammer:** Software
- **Christian Gütl:** Supervision

Steinmaurer A., Bajramovic A., Pollhammer D. & Gütl C. (2022). *Learning Security Awareness in Email Communication Using a Platform for Digital Skill Teaching*. In: Proceedings of 2022 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE).

Learning Security Awareness in Email Communication Using a Platform for Digital Skill Teaching

Alexander Steinmaurer, Azra Bajramovic, Daniel Pollhammer & Christian Gütl

Abstract

Despite emails having been a standard technology since the 1970s, they are still indispensable in daily communication and the number of sent emails continues to rise year by year. This technology is not limited to business, it is also used in education, entertainment, or personal correspondence. Emails are also present in the everyday life of children: they are required to join websites, social networks or to participate in school - especially in homeschooling. However, emails are also associated with disadvantages such as spam, phishing, or malware. To engage children in security awareness we developed the DigiSkill platform, which provides interactive educational experiences. It consists of modules such as email or browser simulations, to increase the student's digital skills. In this research, we evaluated the tool with 52 secondary school students to observe their knowledge on security awareness in the context of email usage and to help gain a better understanding. Therefore, we conducted an in-class evaluation with a tool intervention and related pre- and post-activities. We observed different characteristics in emails that students use for classification and identified common misconceptions. We found that students follow an intensely visual approach, which means the visual representation of an email has a strong impact on the level of suspicion students have. We aim to provide educators with a tool to improve the security awareness skills of their students and to discuss educational possibilities in this field with our research findings.

3.4.1 Introduction

The first email was sent about 50 years ago in 1971. At the time, no one could have imagined that this technology would have such a tremendous impact on digital communications. According to J. Johnson (2021a) 293 billion emails were sent and received on a daily basis in 2019. Statistica further reported that 28.5% of the overall mail traffic is spam, according to Kaspersky the number in 2019 was even higher and spam accounts for more than half of the all mail traffic (56.51%) (Vergelis et al., 2020). The dangers posed by spam emails vary and can range from unwanted junk mail or identity theft to compromised systems.

In 2018 Adobe (Abramovich, 2019) conducted a consumer emails survey in the United States with 1,001 participants, which revealed, that people in the age range between 18-24 check personal and work emails every few hours. These numbers show that email is by no means an outdated technology. Email addresses are already required for children to receive information or register for web services, and of course in school as well. Due to the school closures during the COVID-19 pandemic, the demand for digital communication increased even more. Students were suddenly forced to communicate with their teachers using digital channels instead of face-to-face interaction.

Even though children highly participate in digital communication the awareness about threats and risks that technologies bring in general, and email interaction specifically, are covered subsidiary. Children and teenagers are especially vulnerable and are exposed to Internet threats. Cyber criminals see easy victims in children in particular, as they often lack the necessary experience in digital communication or are more curious. This behavior can be exploited to gather information by means of phishing in order to manipulate targeted data. Websites that are mainly used by children are also potential targets for attackers. In this way, they can find potential victims and use scams like fake prices or lotteries to find out sensitive data (Kaspersky, 2017). It was in this context that we created the DigiSkill platform, a web application that aims to provide teachers and students with an engaging learning environment for digital skill teaching. The system's objective is to offer a flexible environment where students can train skills, for example, security awareness, coding, or data visualization. Teachers can create individual courses consisting of modules such as email composers and inboxes, browsers, web documents, or quizzes.

This research focuses on security awareness in email communication, specifically, whether a tool-based intervention can mitigate the issues mentioned above. We defined the following research objectives:

- RO1: What is the impact of the DigiSkill tool on the learning outcome of participants in the IT security context?
- RO2: What features in emails are crucial for students to classify them as trustful or suspicious?
- RO3: What is the students attitude towards our tool-based approach for IT security education?

Based on these objectives, we conducted a study in a secondary school to evaluate the platform in an educational context. This research aims to introduce an engaging learning tool for the subject of secure email communication and to evaluate it in an educational context. Another central contribution of our work is to point out and discuss students' common misconceptions in this area.

The paper is structured as follows: Section 2 covers frameworks and approaches for security awareness in schools. In section 3 we introduce the DigiSkill platform, in particular, its security awareness modules. Section 4 outlines the conducted evaluation in secondary school and section 5 presents and discusses the findings. Finally, section 6 concludes this paper.

3.4.2 Background and Related Work

As of January 2021, the digital population was approximately 4.66 billion active Internet users (J. Johnson, 2021b). Apart from searching for information, staying up to date with the latest news and events, or watching videos on various topics, one of the most prevalent usages of the Internet is for social media. Out of all Internet users, 4.2 billion are active social media users (J. Johnson, 2021b), posting sensitive personal information on a nearly daily basis, which raises the question of proper protection of this data. Cybersecurity is the practice of protecting critical systems and sensitive information from cyberattacks ("What is cybersecurity?" 2022). As a consequence it is of importance that today's Internet users comprehend the full extent of its meaning. Many companies today take the time to train their employees in cybersecurity awareness with a 53% rise in corporate security training since 2019 (Security, 2021). However, adults are not the only users that have to deal with cyberattacks, with one in every three Internet users being a child under the age of 18 years (Unicef, 2019). Children are starting to use technology at a young age and many parents are not aware of the things their children are exposed to on the Internet. This was supported by a survey conducted in 2016 by Kaspersky Lab that showed that 70% of the surveyed parents were unaware of illegal activities and cyberbullying their children were exposed to (Lab, 2016). In a study by Lastdrager et al. (2017) about anti-phishing training for children, the researchers state that the field of cybersecurity should not solely focus on adults, but that materials for children should be developed as well. This is further supported by a European Commission qualitative study in which the findings implied that younger children are already vulnerable to certain online risks and suggest an improvement in the educational material on online safety (Chaudron, 2015).

The importance of security awareness in children's education is also urged by the Computer Science Teachers Association (CSTA). The organization states different standards that a K-12 curriculum should fulfill when teaching cybersecurity to children at all levels (Seehorn et al., 2011). The CSTA K-12 standard suggests covering security-related topics already at kindergarten level, where children are aware of the need for strong and secure passwords. However, when looking at cybersecurity curricula, many of them are focused on the correct use of social media and being aware of predators that lurk on passwords, as well as the dangers of cyberbullying (Orlando, 2019). This is also supported by a study in which the secondary school curricula from Germany, Austria, England, United States, and Canada are compared to show the representation of the categories of Cybersecurity Body of Knowledge (CyBOK) ("The Cyber Security Body Of Knowledge," 2020) areas in the syllabus. The findings show that the knowledge area relating to "social engineering" and "security awareness" is highly underrepresented in many curricula.

Besides the CSTA K-12 standards, other organizations are aiming for higher integration of cybersecurity topics into computer science education as well. The European Strategy for a Better Internet for Children also raises the issue of the lack of quality in cybersecurity education in schools across Europe. They state that improving the teaching practices of online safety in schools is one of the desired actions in the Digital Agenda for Europe ("A European Strategy for a better Internet for our children," 2012). The mentioned standards also comply with European Commission's DigComp framework (Digital Competence Framework for citizens). According to the framework being digitally competent means to have competencies in all areas of DigComp (Commission et al., 2018). Safety is one of the five DigComp competencies which addresses the protection of devices and personal information as well as protecting oneself from threats while using digital technologies (Commission et al., 2018).

In 2016, the email marketing agency Adestra (2019) conducted a consumer survey and showed that the majority of participating teenagers used emails on a daily basis. This is not surprising considering that emails are needed for creating accounts on most social media platforms that are in use today. They are also used to subscribe to newsletters from companies, as well as for education, especially since the COVID-19 pandemic has brought a higher demand for online classes.

This raises the concern of protecting the younger population from scams that occur on the Internet, which can be done only through adequate education. The percentage of the younger population that is able to recognize phishing emails is quite low; Nicholson et al. conducted a study in 2020 where 83 children aged between 12-17 were tested on their ability to recognize phishing emails and the overall success rate for detecting phishing emails was 59% (Nicholson et al., 2020). Additionally, in a study conducted by Kumaraguru et al. an anti-phishing education system, as well as an anti-phishing game, were tested for their efficacy with 4,517 participants and the findings reveal that participants under the age of 18 performed worse in detecting phishing attacks, while other age groups had no considerable difference in performance (Kumaraguru et al., 2010).

The importance of integrating cybersecurity into early education is evident, however, the speed at which this can be done is influenced greatly by the knowledge level of teachers, lack of expertise, funding and resources (Rahman et al., 2020). The use of gamified approaches and educational platforms is a great way to circumvent factors that slow down the integration of a better cybersecurity curriculum. One such platform is Google's free interactive game Interland ("Interland - Be Internet Awesome," 2017). In one of the game levels with the title "Don't Fall for Fake" players answer questions that deal with specific situations in which a wrong step could lead to serious safety and security consequences. The questions touch upon phishing, sharing credit card information, reliability of sources and validity of emails. Apart from that, the PhishGuru embedded training system (Kumaraguru et al., 2009) is specified for anti-phishing training. It was developed as a response to the evident lack of anti-phishing training platforms in 2007. The system was carefully designed to take into consideration the decision-making process of users paired with different instructional design principles to ensure effective education (Kumaraguru, 2009). Although this tool showed to be an effective solution for better education of Internet users (Kumaraguru et al., 2009), it was developed with the average Internet user in mind and thus still leaves the younger generation's cybersecurity education problem unsolved.

The lack of a solution and of comprehensive research in the area of cybersecurity education for children inspired the development of the DigiSkill learning platform. The idea behind the platform is to support educators in implementing a flexible cybersecurity curriculum. Furthermore, the use of the tool can provide insight into children's understanding of cybersecurity, as well as what type of support educators would most benefit from in this field.

3.4.3 DigiSkill Tool and Learning Experience

Requirements

As outlined above, IT security is an omnipresent topic that concerns every Internet user. Due to the lack of learning technologies in this area, we developed the DigiSkill platform[11]. This open-source web application allows teachers to easily create courses and content that depict real-world scenarios. It aims to prepare students to recognize phishing emails and sharpen their security awareness skills. The requirements for the platform were defined within the project team since the team consists of experts in learning technologies, education and teaching:

- **Learning management system**: The system should provide common features of a learning management system such as user management, course management, and basic learning analytics.
- **Modularity and flexibility**: The system should be flexible and easy to extend by modules.
- **Sandbox environment**: The modules covering security awareness should provide users with a sandbox environment where realistic scenarios can be trained securely, without any impact on the user's system.
- **Individual learning experience**: The system should enable non-linear learning paths where the performance on a task (correct, improvable, or incorrect) has an influence on the following tasks.
- **Feedback**: Students should receive an immediate response on their score after each module and receive information on their performance.

System design

The DigiSkill platform is built as a web application that utilizes the frameworks Laravel and Vue.js. Teachers can create courses using a backend view and every course consists of a number of tasks (see Fig. 3.11). Each task again contains an intro, a type, and an outro. Both intro and outro can either be a text or a multimedia element such as a video, YouTube link, or a figure. This allows multiple tasks to be connected into a complex story that runs throughout an entire course. To provide a flexible system with a wide range of educational options, different modules are supported. A module defines the type of a certain task and the possibilities and functionalities a student has in it. The system provides modules related to security

[11] http://digiskill.codislabgraz.org/

awareness such as an email module, or a browser module, but also other modules such as a quiz, or a coding module. A module is thus a blueprint and can be customized by the teacher for each task.

On the other hand students can enroll for specific courses by getting assigned or using a join code. The students take the course in the pre-defined level path which was defined by the teachers. Different to the teacher's view the students see a simulated Browser or Mail client with all specified information. Depending on the users interaction with the module the task will end correct or incorrect (see Fig. 3.13).

Security awareness modules

The modules related to security awareness are an email and a browser module. In both of these, teachers define transitions that determine if the answer or behaviour was correct, improvable, or incorrect. The platform provides a multitude of possible transitions.

Email module
The objective of this module is to display emails realistically to provide a safe and authentic educational environment. Teachers compose a message by defining both the header (sender, subject, date, etc.) and the message body using the tool's user interface (see Figure 3.11). A WYSIWYG editor can be used to compose a complex, multimedia email. The editor makes it also possible to use the source code of existing emails to easily include real-world examples into a sandbox environment. All emails created for a specific task will be displayed to the students in an inbox showing sender, date, and subject in the overview list. This enables the creation of spam emails that contain specific suspicious characteristics. A large number of categories and features have already been introduced in the research area of spam detection methods. In the DigiSkill tool features of spam message can be referred to one of the following five categories: i) header, ii) subject, iii) payload (body), iv) attachments, v) or URLs (Tran et al., 2013).

Teachers can define what action is correct, improvable, or incorrect for each email in the inbox. Such actions would be closing a suspicious email, clicking on a link, or opening an attachment.

Browser module
The browser module simulates a web browser by rendering a single page. Again, the web page can be created using a WYSIWYG editor, including HTML, CSS, and

JavaScript elements. Additionally, the module provides a link and form generator. This allows evaluation of the users' behaviours, for example when clicking a link or entering a password in an input field. The JavaScript support gives comprehensive possibilities regarding dynamic web pages such as input forms or surveys.

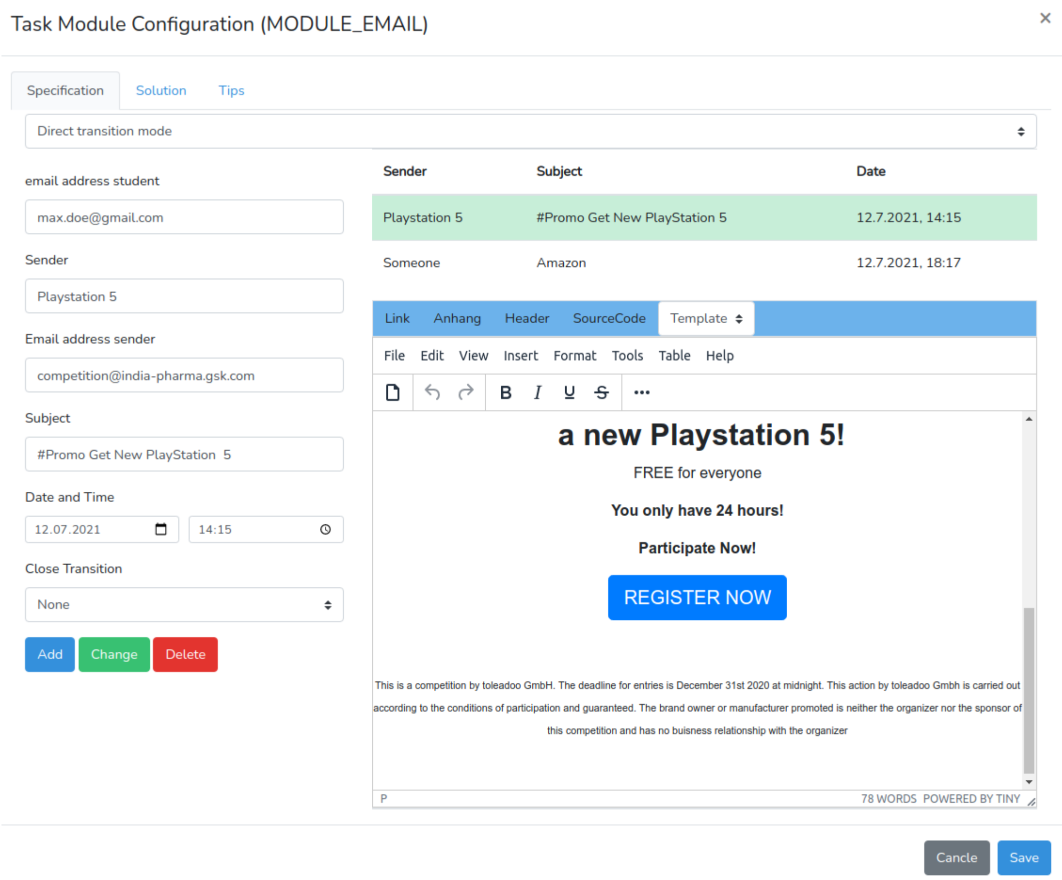

Figure 3.11: This figure shows the teachers view of the email module.

3.4.4 Case Study

The goal of this study was to conduct evaluations to get insights on issues students face in the context of email usage and how the proposed approach and tool can mitigate these issues. The study was designed as an in-class workshop and was held in four different groups. The activities involved quizzes, practical tasks, and a DigiSkill intervention.

Participants

Overall 52 students attended the workshop, of these 34 were female (65.38%) and 18 (34.62%) were male. The age range of all participants was between 12-17, the mean

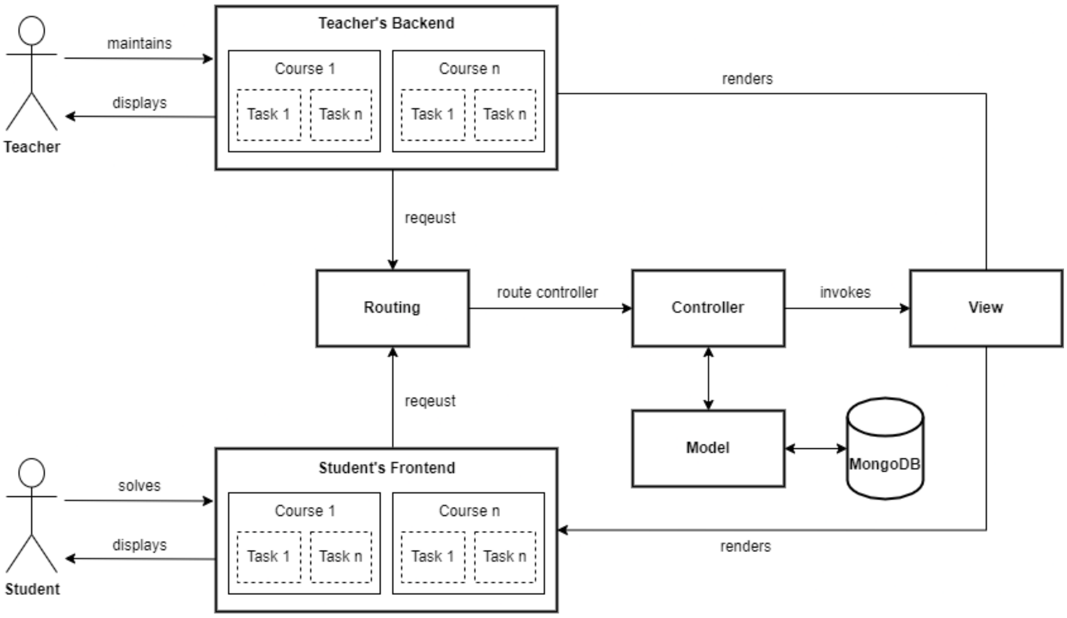

Figure 3.12: This figure gives an architecture overview of DigiSkill.

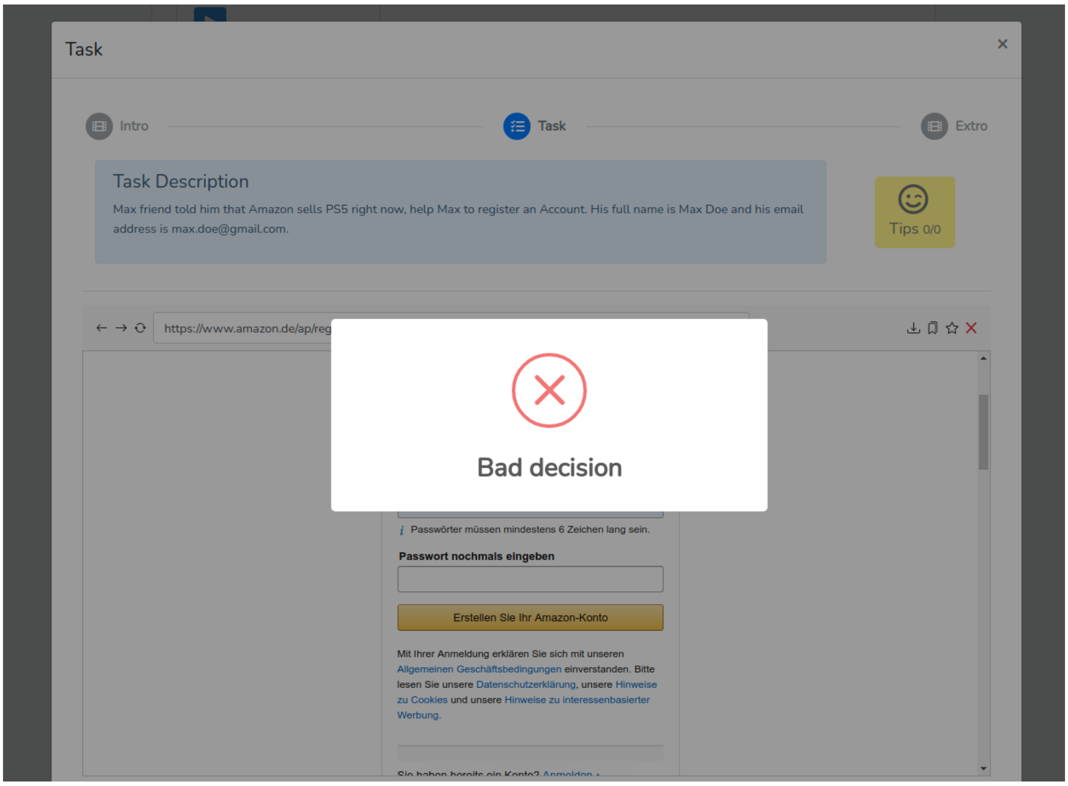

Figure 3.13: This figure shows the student view of the browser module after the student made an incorrect decision.

Table 3.11: Four groups with a total of 52 students attended this workshop.

Group	Grade	Age	Female	Male	Total
A	5th	14.53	14	3	17
B	5th	14.37	15	4	19
C	3rd	12.57	2	5	7
D	7th	16.44	3	6	9
Total		**14.54**	**34**	**18**	**52**

Table 3.12: Ten emails were presented to the participants that were to be classified as either ham or spam.

	Sender	Category	Typ	Features	# Fraud
A	N/A	money scam	spam	sender, links, salutation, opportunity	42
B	FX-trading	lottery scam	spam	spelling, links, subject, opportunity	19
C	Netflix	blocked account	spam	sender, links, urgency	10
D	Dropbox	account upgrade	ham	-	13
E	Conrad	newsletter	ham	-	5
F	Amazon	fraud detection	spam	sender, links, urgency	31
G	Netflix	password recovery	ham	-	7
H	Netflix	fraud detection	spam	sender, subject, attachment	7
I	Amazon	lottery scam	spam	sender, links, opportunity, urgency	29
J	Kurier	lottery	ham	-	5

age was 14.54 years (SD=1.21). Table 3.11 shows further information regarding the participant groups. None of the students had any prior knowledge about IT security from school education.

Material and methods

The evaluation's activities consisted of a pre-questionnaire, two quizzes, two email classification tasks, and a post-questionnaire. In the pre-questionnaire, the students answered questions regarding their email behaviour, experience with emails, phishing, malware, and data security. In addition, the students were asked three self-evaluation questions rating their familiarity in i) computers in general, ii) IT security, and iii) emails. Both quizzes involved 15 questions in the field of IT security (see Table 3.13). To investigate the effect of the tool the same quiz questions were used in the pre- and the post-phase.

One central aspect of this study was to evaluate whether the students identify emails as spam. The authors thus prepared ten different emails from various areas of daily life. Six emails were intended to be fraudulent and four emails to be trustworthy[12]. The examples involved mails from Netflix, Amazon, Dropbox, local newspapers, or stores. Since the categories of the emails were diverse, the suspicious features they contained offered a wide degree of difficulty ranging from newsletters or lottery to money scams. Table 3.12 lists all emails used in the study. The column Sender refers to the company that appears to the participants as the author of the email. In some cases, the sender was spoofed (f.i. mismatch between displayed sender name and address). A number of specific suspicious features were defined for each spam email. These features include URL spoofing (manipulation of links), spelling errors, misleading subjects, opportunities (price winning), or urgency (immediate call for action). After this activity, the students should also rate how confident they felt about their classifications on a Likert scale from 1 (very confident) to 5 (not confident).

All user data was collected via Google Forms. The temporary email address served as an identifier throughout all activities, to ensure that all activities were completely anonymous. The R programming environment was used for further data analysis.

Procedure

Overall four evaluations were conducted in computer science classes of an Austrian secondary school and took 100 minutes per group, which is the equivalent of two lessons. The school workshop was split into three parts: i) a pre-phase, ii) the DigiSkill intervention, and iii) a post-phase. All tasks were accomplished in a web browser, whereas the tasks in the second phase was completed on the DigiSkill platform. At the beginning of each class, the teacher introduced the workshop and then left the class to promote a less stressful and more motivating atmosphere during the whole activity period. Each workshop was held by a member of the project team, an experienced lecturer with a background in computer science teaching.

During the pre-phase all participants first answered the pre-questionnaire. Following on from this, each student was provided with two temporary email addresses that were only available within the context of this workshop. One account was used in the pre-phase and the second one in the post-phase. Each email account was comprised of an inbox with five emails - three spam and two non-spam (see

[12]We refer to non-spam emails as ham, a term used in the literature as the opposite of spam (SpamBayes-Development-Team, 2002).

Table 3.12). The participants were asked to individually go through each email and classify them as trustful or suspicious. After this, all students received a quiz covering 15 questions about topics on IT security (see Table 3.13). The final activity of this phase was a collaborative brainstorming session to discuss and identify suspicious features in emails.

During the next phase - the DigiSkill intervention - the students worked on the prepared course in the tool for 30 minutes. This course is accompanied by an embedded narrative covering 13 different tasks. Based on the story, a fictive person had the intention to purchase a new Playstation 5 and registered her or himself with different newsletters to receive the latest information. Throughout the different tasks, the students were faced with spam emails aiming to phish for personal information or compromise the user's system. The tasks in the course include four exercises in which the users have access to an email inbox with different messages, three exercises involving suspicious websites and six quizzes asking questions about the activities. The email and website samples together with the quiz questions are intended to illustrate prototypical dangerous situations in digital communication.

The post-phase included three activities: a second email classification task, the quiz on IT security, and a post-questionnaire. Once again, the participants received an email inbox with five emails and all of these had to be classified as either trustful or suspicious. The 15 quiz questions were equal to the quiz in the pre-phase to measure if the tool's intervention had an impact. Again, a member of the project team supervised a final brainstorming session about email security.

3.4.5 Findings and Discussion

The pre-questionnaire consisted of questions about the experience the participants had in the topic. All 52 participants answered that they have at least one email address, while 40 students stated that they had two or more addresses. Overall, 16 people answered that they knew what phishing emails are and all of them were able to give a correct definition of the term.

The three self-evaluation questions were observed by a factor analysis on these three variables. This shows that the loadings for this item were non-qualified for a common factor. The Spearman rank correlation between the two ordinal variables (v1) students knowledge about computers and (v2) their perceived security skills shows a moderate positive correlation of $\rho = 0.462$ ($p < 0.001$). There is another positive correlation between the knowledge about computers (v1) and the self-rated skills regarding emails (v3) $\rho = 0.512$ ($p < 0.001$). Notably, there is just a very weak correlation between security awareness (v2) and experience with emails (v3) $\rho = 0.23$ ($p < 0.16$).

Results of Quizzes

The performance of the students in both quizzes was measured by summing up all the correctly answered questions (see Table 3.13). Of all 52 students, 36 participants completed both quizzes. A Shapiro-Wilk test on the paired samples was performed to further investigate the impact of the interventions on performance, which showed that the data does not show a normal distribution. We thus decided to use a non-parametric method for testing the effect, to be more precise a Wilcox signed-rank test was applied. The test showed a considerable improvement through the intervention, but no statistical significance (p=.3946).

Table 3.13: The study participants answered 15 questions before and after the intervention. 36 participants completed both quizzes.

	Question	# Pre	# Post
1	You should click on the link in an email to figure out if the email is real?	36	35
2	You should be suspicious if an email uses a link shortener service?	35	34
3	Before you click on a link, hover over it to see the destination address.	32	36
4	A serious company sends emails from a public email address.	26	24
5	A serious company does not request personal data by e-mail.	29	33
6	Phishing emails often contain links to the website they are trying to imitate.	33	32
7	Spoofing is a phishing attack in which the sender address is faked.	33	35
8	It is safe to open unwanted attachments.	34	35
9	You can get a computer virus or malware by opening an attachment.	33	33
10	The majority of companies use security mechanisms against email spoofing.	34	35
11	Attackers exploit human behavior.	34	35
12	You should use the unsubscribe link in fraudulent emails.	32	26
13	Communication via HTTP is not encrypted and can be intercepted by an attacker?	22	26
14	It is not a problem to reuse your password.	34	35
15	The attackers use personal information to their advantage.	35	35

The students already showed a good performance on the first quiz. The mean value for correct answers is 13.10 points (SD=1.18, min=10, max=15). After the intervention, the mean value slightly increased to 13.55 points (SD=1.05, min=11, max=15). The results in Table 3.13 points out that the participants intuitively answered correctly in many areas. The differences between pre- and post-quiz show that there are slight improvements in the answers. Students that received a good number of points in the first quiz had similar results in the second one. This means, in regard to RO1, that no statistical significance is given, but at least a considerable improvement.

A discrepancy emerges when comparing the quiz results with the observations of the email classification tasks. Although many students gave correct answers in the quiz, only a small proportion of the participants could draw a line between theory and the practical application of these concepts. 32 participants (88.88%) answered that they would check a link by hovering over it before they click on it (Question 3). In the practical activities (tool intervention and email tasks) only a few students did this, because it is very closely related to email C (10 correct classifications only). A similar picture emerges with unwanted attachments: Question 8 deals with the behaviour of email attachments, where 34 students (94.44%) answered that this is unsafe. A spam email from Netflix in the email classification task (email H) contained such an attachment, but only 7 people (19.44%) supposed fraud.

Email Classification

In the pre-questionnaire the students were asked what is characteristic for spam emails. The three most frequently stated associations are 1) suspicious senders 2) suspicious links, and 3) spelling errors. The students also mentioned that they get curious if they receive emails they do not expect.

Overall 48 students out of 52 finished both email activities. Table 3.12 shows the number of emails that were categorized as suspicious.

After both classification tasks, the students rated how confident they felt with this activity. The mean level after the first activity was M=3.326 (SD=1.01) whereas after the second activity confidence decreased to M=3.109 (SD=1.23). The three self-evaluation variables and the performance of both classification tasks show a very low degree of correlation for each variable. This shows that the initial estimation the participating students had of their skills is not significantly related to the performance.

When identifying similarities in the choices of the classified emails, it seems that the students followed a strong visual approach. Three emails (A, F, and I) were recognized as fraudulent by most of the participants (>60%). The commonality between these three emails is that the graphical representation of the message is very simple (mainly text and logo) and the sender's address is conspicuous: accounts@mazon.com, or relevea@ca-anjou-maine.fr. This indicates that if the layout of an email was not appealing it was more likely to be classified as spam. The participants were precise when it came to spelling or encoding issues in a message. Email D had an encoding problem with a character which was conspicuous for some students. Another considerable observation is, that even though the emails C and H are spam, only a quarter of the students identified them correctly. These emails are originally from Netflix and had been slightly modified, thus giving them an authentic appearance. In the final brainstorming sessions, the students responded that the appearance of the messages and content had not made a dangerous impression, but they have not thought about hovering over the link. In email H an XLSM (spreadsheet file that supports macros) was attached, but the participants noted, that they were not aware of the potential danger. This shows in an exemplary fashion, however, that students have a good theoretical understanding coupled with an intuition for security-related topics and dangerous features in emails. The problem here, would appear to be applying the skills in real-world situations, which in turn would seem to be a challenge for education (RO2). Additionally, the students were also asked to answer open-ended questions regarding their experience with the tool. Some students mentioned that the system should have a faster response (n=3), to make it more usable. One student suggested adding videos or animations to explain certain topics. Another student said the spam emails should be made even more realistic, to provide harder tasks.

Finally, the participants answered two evaluation questions about the DigiSkill tool ranging from 1 (fully agree) to 5 (fully disagree) regarding RO3. First, they answered how meaningful and realistic the tool was when it comes to visualization and explanation of email and websites. The response (M=2.5, SD=1.18) shows, that the majority of the students were satisfied with this purpose of the system. The second question covered the opinions the students had about the usefulness of the system compared to explanations in traditional email clients or browsers. Again, the evaluation given by the students showed that they see a benefit in the system (M=2.11, SD=1.32). Examples in the literature (Kumaraguru et al., 2010) also show that teaching security awareness to students is an important but also difficult task. Educational systems can help to make this more engaging and successful.

3.4.6 Conclusion

In this paper, we defined three research objectives covering the impact of the DigiSkill platform on the students' security awareness skills, identifying features in emails that are relevant for students to classify them as suspicious or trustful and finally, evaluating the students opinion on our approach. We conducted a study with four groups and 52 participants in total. The results showed the intervention of the tool did not have a statistical significance, but it did have a considerable influence on the security awareness skills of the participants. Apart from these issues, students tend to have a good intuition on security topics but they have some problems with real-life applications.

Threats to Validity

Due to the COVID-19 restrictions, school workshops proved difficult to arrange. The number of participants was limited as a result. This limitation influences the meaningfulness of the data and the statistical approaches applied to them. Because of the brief workshop duration of 100 minutes, it was not possible to conduct a longer study to measure the impact of the tool over a longer period of time. The effect of this intervention could have been determined much more meaningfully had there been a larger sample size and an A/B test.

Future Work

In the current phase of the project, our main objective was to use the DigiSkill tool to evaluate it in a school workshop for obtaining improved insights into the experience secondary school students have in email technology. In further research, our aim will be to achieve an evaluation with groups from different schools over a longer time period. An experimental group and a control group will thus be observed as a means of measuring the effect the tool intervention has. This should help not only in focusing on the declarative knowledge of the students, but also on their improvement in the context of security awareness skills. Besides the perspective of the students we also want to include more teachers in future studies. This should help to improve the tool and to set a specific focus on problems related to security awareness in education. In addition to this, we also plan to include other email related topics in our research, such as mail agent settings or availability of email encryption.

3.5 Article 5: Social Media Awareness Training

Motivation

Technology and society are inevitably linked in today's world. For this reason, the combination of these fields is present in many areas of K-12 education. Since social media is omnipresent in students' daily lives, this topic is highly interesting to many students. One approach to explain social media phenomena is the so-called *Dagstuhl Triangle*, which connects three perspectives: technological, user-oriented, and socio-cultural (Brinda et al., 2016). In contrast to the presented three-dimensional approach of the Dagstuhl Triangle, the topic of social media is often taught in only one dimension, depending on the respective school subject. For this reason, social media awareness training (SMAwT) was designed, developed, and evaluated as training for secondary school computer science education.

Contribution

SMAwT was developed within an interdisciplinary team of computer science teachers, CSEd researchers, instructional designers, and software developers. This made it possible to develop a research-driven concept for social media education within the context of computer science classes. This training is a phenomenon-based learning environment focusing on different perspectives on social media awareness. The findings of the development and evaluation give valuable insights into students' skills in social media awareness and conclude how training can be implemented and embedded into education.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - The starting point for the training was the lack of technologies and approaches for social media awareness in computer science classes.
- **Theoretical Foundation** - A literature survey was conducted to review state-of-the-art research regarding social media awareness. This survey revealed some valuable models within the scope of this study.
- **Theoretical Concept** - An educational and technical concept was developed based on the models that were discovered within the literature survey. The concept proposed a training that is highly aligned with the learning outcomes, assessment, and teaching and learning activities.
- **Implementation** - The implementation followed an agile development process. Within regular workshops between computer science teachers, CSEd researchers, and web developers the platform was developed but also a learning concept including educational resources.

- **Evaluation** - The training was evaluated in a qualitative study with teachers and a quantitative study including 216 students in secondary schools. The combination of both interventions gave insights into the social media usage of students and also how social media education can be taught engagingly.
- **Validation and Best Practices** - The mixed-method design of the evaluation revealed interesting findings on the students' social media usage. This showed how engaging experiences can be designed in a way to have an affect on the students' performance.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Formal analysis, Writing - Original Draft, Writing - Review, Editing
- **Andreas Dengel:** Methodology, Resources, Investigation, Writing - Original Draft, Writing - Review, Editing
- **Michael Umfahrer:** Software
- **Kerstin Zöhrer:** Software, Resources, Writing - Original Draft
- **Paul Kogler:** Software, Resources, Writing - Original Draft
- **Christian Gütl:** Supervision

Steinmaurer A., Dengel A., Umfahrer. M., Zöhrer, K., Kogler P., & Gütl, C. (2023). *A SMAwT Approach for Raising Social Media Awareness in Secondary CS Education - Maybe a Little Bit Too Much?*. Manuscript submitted for publication.

A SMAwT Approach for Raising Social Media Awareness in Secondary CS Education - Maybe a Little Bit Too Much?

Alexander Steinmaurer, Andreas Dengel, Michael Umfahrer, Kerstin Zöhrer, Paul Kogler & Christian Gütl

Abstract

Social media platforms have transformed the way people communicate, share information, and interact with each other. While studies have shown that children and teenagers use social media platforms on a daily basis, they are also exposed to the negative aspects of social media such as fake news, hate speech, or the fear of missing out. It is therefore imperative for schools to address social media education to equip students with the necessary skills to navigate the online world. This paper presents SMAwT (Social Media Awareness Training), an innovative learning tool designed to raise social media awareness among students in computer science education from an interdisciplinary approach combining technological, socio-cultural, and user-oriented perspectives on phenomena occurring in social networks. We explore how the tool can be integrated into CS classes by conducting a study involving 216 students aged 10-17 and seven secondary school teachers. Pre- and post-intervention, the students completed a social media competence quiz and answered questions about their social media usage, self-efficacy, and task-related interest. Semi-structured interviews were conducted with the teachers to gain deeper insights into their experience with the tool and its use in computer science classes. The results of the quiz were analyzed alongside the qualitative content analyses from the teachers' interviews, highlighting the effects of self-efficacy, social media usage, and interest. The results show, that the tool is efficient to increase the students' social media skills from a technical perspective. However, the quantitative data indicates, that the students' socio-cultural awareness decreases after using the tool. By combining the quantitative with the qualitative data it gets clear, that students develop a critical attitude when using social media.

3.5.1 Introduction

Social media are a huge part of teenagers' everyday life. According to the JIM study (Feierabend et al., 2022) 99% of all asked teenagers between 12 and 19 years have a smartphone and the majority of them are using social media for daily communication. The connected phenomena should therefore be part of their education. Especially topics such as cybersecurity (Quayyum, 2020) as well as data privacy, data security and databases (Grillenberger & Romeike, 2014) are part of existing traditional computer science curricula in K-12 education (e.g. Guo and Ottenbreit-Leftwich (2020), Hörmann et al. (2022), Javidi and Sheybani (2018), and Lamprou et al. (2017)). Besides technological topics, there is also a high social impact. The raise of social media has led to the emergence of issues like cyberbullying (Hamm et al., 2015), online grooming (Cano et al., 2014), and FOMO (fear of missing out) (Abel et al., 2016). All of these topics can be found in media education subjects, but also in new, interdisciplinary subjects such as "Medien & Informatik" (*Media and Computer Science*) (Lamprou et al., 2017), in Switzerland or "Digitale Grundbildung" (*Digital Basic Education*) in Austria (Hörmann et al., 2022). Such innovative subjects combine computer science education, user-oriented application skills, and socio-cultural questions regarding the reciprocal effects of phenomena in the digital world, as shown in the Dagstuhl Triangle for education in the digital world (Brinda et al., 2016).

There are several digital tools that focus on social media education such as Google's *Be Internet Awesome* (Google, 2023), *Social Media TestDrive* (DiFranzo et al., 2023), or *InstaHub* (Dorn, 2019) that simulate platforms and cover different topics in the context of media education. However, existing tools mainly address just a few aspects, either from the technological side (function) or the psychological/socio-cultural side (effect) of the topic of Social Media. Additionally, these approaches provide a sophisticated learning environment, and students can learn individually, which makes it hard to perfectly include them into school classes. Further, these tools are usually seeing social media education from the lens of pure media education and not in the context of computer science. A web-based *Social Media Awareness Training* (SMAwT) has been developed within a team of teachers, researchers from Computer Science Education and Media Educations, together with web developers to provide comprehensive interdisciplinary education through phenomena-based learning. SMAwT is designed for the seamless integration of media literacy into computer science classes. The tool was evaluated in a mixed research design approach together with six schools, including quantitative data from 216 students and qualitative data from seven teacher interviews. To observe the efficiency of the tool and its educational possibilities the following research questions have been stated:

- **RQ1** - Is the SMAwT approach effective in fostering students' digital literacy?
- **RQ2** - What factors influence learning outcomes and performance in a social

media environment?

- **RQ3** - How do teachers and students perceive SMAwT in regard to the learning goals to be achieved?

The main contributions of this project include:

- Developing a web-based tool for social media awareness in CS classes based on the different perspectives of the Dagstuhl Triangle.
- Evaluating the tool in a formal educational setting including six CS classes in secondary schools with 216 students completing the course and seven CS teachers.
- Highly increasing the students' awareness regarding socio-cultural aspects such as hate speech or posting appropriate content.
- Providing teachers with a story-based and interactive tool that assists to visualize or explain specific topics related to social networks. The tool can be integrated into various phases and educational scenarios in CS classes.

3.5.2 Background and Related Work

The Role of Social Media in Computer Science Education

Tretinjak and Andelió argue that teachers need to become familiar with the digital competence areas of *information*, *communication*, *safety on the internet*, *problem-solving*, and *content creation*. This enables learning and teaching knowledge, skills, and attitudes to achieve critical and creative use of ICT and digital media for the purpose of achieving goals related to work, learning, or leisure. The competence area *information* comprises the ability to identify, locate, retrieve, store, organize, and analyze digital information. *communication* summarizes skills such as communicating through online tools, together with aspects like privacy, safety, and netiquette. The area *safety on the internet* includes properly managing personal protection, data protection, and digital identity protection. *Problem-solving* means identifying digital needs and resources, solving conceptual problems through digital means, creatively using technologies, and solving technical problems. *Content creation* emphasizes competencies related to creating and editing new content, integrating and re-elaborating previous knowledge and content as well as applying intellectual property rights and licenses (Tretinjak & Andelic, 2016). We can apply these competencies to Social Media, focusing on specific aspects regarding

- received and shared *information* through messages, posts, profiles, etc.
- mutual *communication* within these networks with other users, but also with

programs,
- *safety aspects* regarding the potential risks in social networks
- *problem-solving skills* focused on issues occurring through using social networks, and
- *content creation* through text, images, videos, etc.

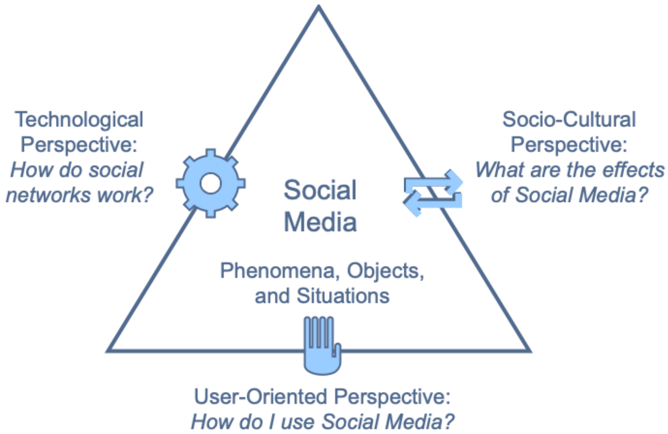

Figure 3.14: Technological, socio-cultural, and user-oriented perspectives on Social Media.

All of these competencies are linked to different disciplines including computer science education, media pedagogy, and several application uses. Interdisciplinary approaches such as the Dagstuhl Triangle structure these different perspectives on phenomena of the digitally connected world from an educational perspective (Brinda et al., 2016). In this theoretical framework, the technological perspective focuses on the question "How does it work?", related to an object, situation, or phenomenon of the digital world. The user-oriented perspective asks "How can I use it?" and comprises all application-related skills, e.g. using software and hardware. From the socio-cultural perspective, the effects (uni- and bidirectional) of these objects, situations, and phenomena are taken into account (Brinda et al., 2016). In later iterations, a design perspective was added to the Dagstuhl Triangle, resulting in the Frankfurt Triangle (Brinda et al., 2019), which also accounted for interdisciplinary connections. As this study wants to distinguish as clearly as possible between different perspectives for evaluation purposes, this paper follows the idea of the Dagstuhl Triangle and relates its perspectives to Social Media (see Fig. 3.14).

Approaches towards Social Media Literacy

Göbl et al. argue that transparency and knowledge regarding underlying principles of social networks are important steps towards empowering adolescent users. They developed and evaluated a serious game approach for fostering social media literacy (Göbl et al., 2019).

Social Media TestDrive is an interactive social media simulation that combines experiential learning in a realistic and safe social media environment (DiFranzo et al., 2023). The approach includes educator-facilitated classroom lessons. The evaluation of the tool could show that TestDrive induced high engagement. Students reported a fostered understanding of topics such as digital citizenship. Educators observed students engaging in meaningful classroom conversations. DiFranzo et al. further suggest involving multiple stakeholder groups, such as researchers, youth, educators, and curriculum developers, when designing educational technology (DiFranzo et al., 2019). Social Media TestDrive, therefore, focuses on application skills but also critical perspectives on the effects of social networks.

Google's program *Be Internet Awesome* (Google, 2023) comprises four components: the Internet Code of Awesome (the Code), Be Internet Awesome Pledge (the Pledge), Be Internet Awesome Curriculum, and a browser-based game called *Interland*. The program has five central tenets: "Share with Care", "Don't Fall for Fake", "Secure your Secrets", "It's Cool to be Kind", and "When in Doubt Talk it out". Each of the first four of these tenets is connected to a level in the game. Seale and Schoenberger investigated how the program conceptualizes and presents safety threats on the internet. They argue that Be Internet Awesome is well designed and that it addresses common themes related to internet safety, but that the promised rise in childrens' awareness gain is not comprehensive. They criticize that Be Internet Awesome fails to consider information use past a surface level and that it ignores elements outside of the user's control. Further, it creates the image of Google as an authoritative and benevolent Internet expert (Seale & Schoenberger, 2018). This program focuses especially on the effects of and criticism in social networks.

InstaHub (Dorn, 2019) is a social network used in the classroom to teach the topics of databases, social networking, and privacy. It has an integrated SQL interface that can be used to formulate queries for the database of the social network. The tables of the database are ads, comments, users, likes, photos, follows, tags, password resets, and analytics for brands. Following these learning goals, InstaHub focuses in particular on the technological perspective of the Dagstuhl Triangle.

These tools often stand for themselves and are therefore difficult to integrate into the classroom. Further, each of these innovative approaches to fostering social media literacy and computer science-related competencies focuses on just one or two perspectives of the Dagstuhl Triangle. However, since the topic of social media must be viewed from several dimensions to be able to understand related phenomena accordingly, it is important to include these perspectives. For this reason, a game-based approach was developed, that can easily be integrated into different phases of CS classes. This tool emphasizes technological, socio-cultural, and user-oriented perspectives, for an interdisciplinary and comprehensive view of social media education.

3.5.3 Social Media Awareness Training

Educational Concept

Since social media education gets more and more relevant also in school curricula, an educational concept for social media education in CS classes was developed. The starting point for this concept was to state the cognitive learning goals of each student that should be covered in the tool. All activities within SMAwT are based on Anderson and Krathwohl's taxonomy of learning (Krathwohl, 2002). All learning objectives are covered in three so-called chapters. These chapters are i) *Dive into the world of Social Media*, ii) *Social Media and you*, and iii) *The dark sides of Social Media*. Table 3.14 illustrates the assignment of each learning objective to its corresponding chapter and the perspective of the Dagstuhl Triangle.

To provide an engaging learning experience, a story-based approach was applied. The story of Simon, a 18-years old travel influencer, is the common thread throughout the whole training. The protagonist shares his experience when starting with social media and explains some general phenomena of social media by demonstrative examples that he went through. The narrative takes place on the fictitious social media platform *Pointer*.

All three chapters have a similar structure: The beginning of each section is a brief whiteboard animation-style video where a third-person selective narrator explains a specific problem that happened to Simon. Next, the virtual avatar of Simon (see Figure 3.16) guides the player through the chapters by explaining and giving hints. The communication with the avatar can happen linearly, which means he explains specific aspects of the lessons or non-linearly in form of dialogues where he responds based on the students' input. Further, each chapter consists of different modules, such as quizzes, puzzles, or sorting games to have a high level of interaction between the system and the player. Overall, a well-balanced mix of multimedia content should engage the students during the course. All content was created by members of the project team, including the platform, images, and

Table 3.14: Learning objectives of the modules in SMAwT from the technological, socio-cultural, and user-oriented perspective on Social Media

chapter	*technological perspective*	*socio-cultural perspective*	*user-oriented perspective*
Dive into the world of Social Media	The students can create a safe password and assess different passwords' security.	The students can explain why they need to differentiate between friend requests on Social Media and argue their assessment of various friend requests.	The students can differentiate between personal data and non-personal data. They can argue about what to post and which information should be considered personal data.
Social Media and you	The students can detect a hacker's attack on their accounts. They can evaluate their situation and argue what to do if a profile has been compromised.	The students can identify problematic content in pictures and posts and can justify their statement why certain content can be problematic.	The students can point out the differences between appropriate posts and inappropriate ones.
The dark sides of Social Media	-	The students can point out differences between hate speech and criticism and explain their effect on readers.	The students can identify inappropriate messages. The students can analyze situations on different social media websites and argue for a specific response/behavior.

videos.

In the following subsections, the three chapters are explained in detail.

Dive into the world of Social Media.

The first chapter covers topics such as social media in general and digital identity in detail. The chapter starts with a video of Simon who tells about his experience on social media, by an example where one of his followers got access to his *Pointer* profile. The students playfully learn about secure passwords, personal data, and how networks with social media are structured. A graph representation with nodes and clusters should visualize these connections and introduce an example of graph theory in the students' daily lives (see Fig. 3.15).

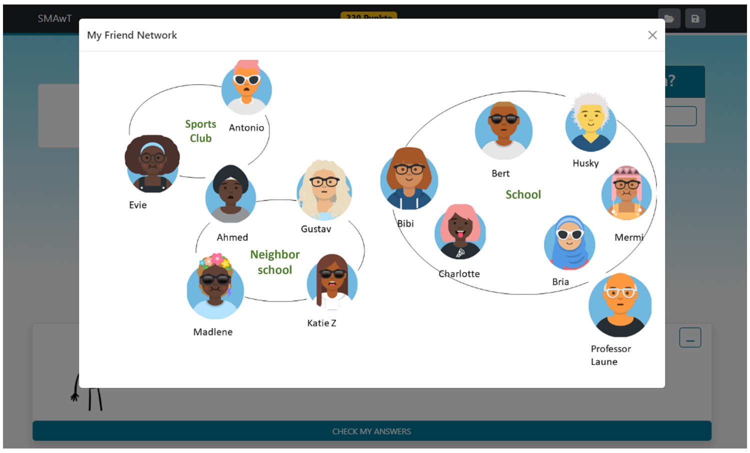

Figure 3.15: A friend network can be visualized with nodes and clusters, to bring the concept of graphs into the context of (social) networks.

Social Media and you. The focus of this chapter is the creation of content on social media. It further explains social networks, the Internet, and posting. Since the story is based on a hacker's attack on Simon's *Pointer* account, the characteristics of a hacker attack and countermeasures are explained within the introductory video. In addition, it is pointed out that posting personal information (i.e. in pictures) can cause severe security issues. Therefore, the reporting of posts is also pointed out, to explain this important feature of social media platforms. The chapter contains several mini-games where students receive images from postings and have to decide if they contain personal information or if they would probably report the postings. Within these games, all postings and photos appear in the style of social media platforms, to mimic a realistic scenario.

The dark sides of Social Media. Based on the JIM study a high number of social media users between 12 and 19 were already confronted with cybermobbing, cyber grooming, fake news, hate speech, and the fear of missing out (FOMO). Therefore, the third chapter addresses these negative aspects of social media. Teachers reported that these topics are hard to integrate into class since some students can hardly discuss these highly personal topics with teachers. In this way, SMAwT is designed as a possible way to communicate these sensitive issues to students in a subliminal way. The third chapter includes two videos: one which is explaining these phrases to the players and the second one deals with strategies against them.

Another central part of the third chapter is disinformation. Therefore, the so-called *CRAAP Test* (Fielding, 2019; Hanz & Kingsland, 2020) is used where the students are encouraged to think critically about what they read on social media. CRAAP is an acronym for Currency, Relevance, Authority, Accuracy, and Purpose of the given sources.

Again, the students will learn about these phenomena through examples and mini-games, where they are confronted with comments and messages including hate speech, mobbing, or grooming. Based on their responses Simon, the virtual avatar, gives them feedback for a better understanding.

Development of SMAwT

SMAwT (Social Media Awareness Training) is a client-side web application developed using the Javascript framework Vue.js. The progressive web framework allows the creation of fast and modern web applications. To accelerate the development process and to ensure data protection, the implementation of a backend was intentionally omitted. No data collected by the tool is stored or processed on any server, to have a safe and data-sensible learning environment.

The tool can be used from any device that provides a browser. This means that users are not tied to a specific operating system or device and can easily access the application. Already during the conception phase, much emphasis was placed on a consistent design, to provide the users with an easy-to-use interface. The application is divided into three logic components covered below.

Registration
In order to use the application, creating a fictitious account is first required. However, this is different from a traditional account creation, which requires an email address and password. Rather, *SMAwT* mimics a simple social network. To create a user profile, the user must create an avatar and provide some basic information. After registration, the user can start with the actual chapters.

Tasks

SMAwT tells the story of the imaginary influencer Simon, who shares his experiences with social media. The story is told in a multimodal way, including short videos, figures, and dialogues. Simon accompanies the user as an avatar guide through the chapters and acts as a central point of contact for explanations, questions, and problems.

A chapter consists of several tasks. Between the individual tasks, the user interacts with the avatar Simon, who always provides assistance and feedback. There are different types of tasks that are implemented in the form of modules. The multitude of modules makes it possible to implement exciting and varied stories. There is a plethora of interactive modules such as the chat in Fig. 3.16 and non-interactive modules such as videos.

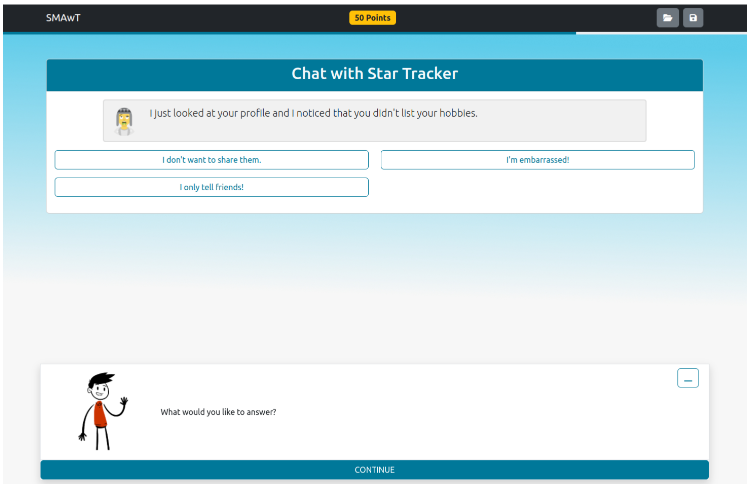

Figure 3.16: Chat module where the social media phenomenon of grooming is simulated.

The following is a list of all modules currently available in SMAwT.

1. HTML Viewer
   This module enables the integration of HTML files that can contain both text and images. The formatting options are extensive and use the full range of functions offered by HTML and CSS. Given the extensive creative freedom it offers, this module presents an effective means of producing diverse and captivating content for educators.
2. Video Viewer
   This module enables the easy integration of videos. It also ensures that the videos play automatically and cannot be skipped.

3. Password Strength Checker
   This module makes it possible to determine the strength of passwords. The calculation takes place locally in the browser and private data is not shared with external services. The lightweight password estimator *zxcvbn* is used to calculate the password strength (Wheeler, 2016).
4. Quiz
   This module enables the integration of knowledge quizzes that support both single-choice and multiple-choice question formats. A unique aspect of this quiz module is the integration of social media elements. Beyond just displaying the question, the content to which the question relates can also be displayed. This module makes it possible to integrate posts that are very similar in appearance to posts from other popular social networks.
5. Sorting Game
   This module is a game-based activity where students are presented with a series of statements that may be either true or false. The objective of the game is to sort the statements into two categories - "True" and "False" - by dragging and dropping them into the appropriate boxes.
6. Image Scanner
   The objective of this module is to help students understand the importance of responsible image sharing and to identify potentially problematic content in images.
7. Puzzle
   This module involves a puzzle game activity in which students are required to assemble the pieces of a puzzle. Once the puzzle is fully completed, the information presented in it can be comprehensively read and understood.
8. Information Chat
   The chat module is designed to store information entered by students and use it for evaluation purposes. It mimics a chat platform and offers the student several predefined response options to choose from.

No programming knowledge is required to create tasks, as *SMAwT* pursues a configuration-driven approach. For this purpose, a special, easy-to-understand data structure has been designed with the aim of creating tasks quickly and easily. The data structure is represented in the JSON format, comprising a series of sequential steps. Each step specifies the modules to be loaded and the corresponding information to be displayed on the screen. The subsequent step to be displayed is determined by the user's actions.

Assessment and Feedback

In the course of the story, the influencer Simon attempts to obtain personal data from the user, which is subsequently used for the evaluation. After completing the entire story the user is presented with this data, highlighting how social networks collect and analyze user data. This can be seen in Fig. 3.17. The aim of this exercise is to encourage users to think about how easily their personal data can be collected and how it can be used to build a comprehensive profile.

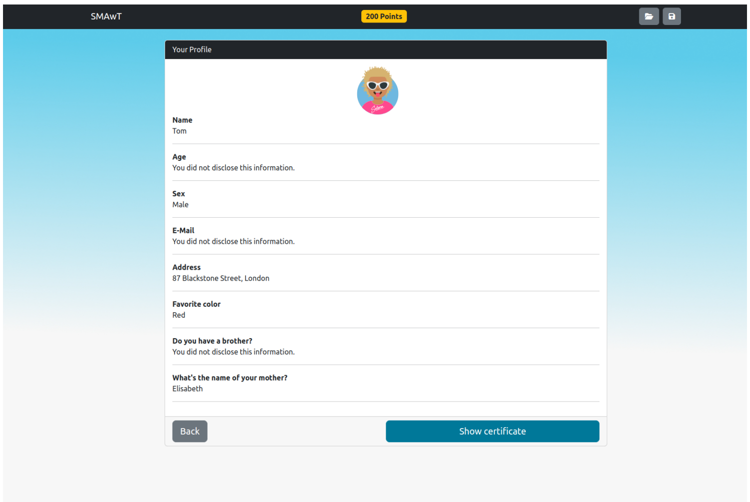

Figure 3.17: The self-assessment screen of SMAwT, which is available after completing all chapters.

3.5.4 Methods

Participants and Recruiting

For the purpose of this study, six secondary schools were invited for the experiment. Schools were selected to include a heterogeneous group of students with a mix of urban and rural schools in Austria and Germany. Within these six schools, seven computer science teachers used the SMAwT platform in their CS classes. The background and experience of the teachers also varied from one year of professional experience to over 20 years. Overall 436 school students (227 male, 201 female, and 8 diverse) attended the SMAwT intervention. All of them at least started the pre-test at the beginning of the lessons. Finally, 216 students (105 male, 108 female, and 3 diverse) completed both, the pre-and post-test. The students that completed all activities are between 10 and 17 years old (M=12.97, Sd=1.15) whereas 97 students (44.91%) are attending an Academic School (Gymnasium) and 118 students (54.63%) are attending a Lower Secondary School (Middle School). Depending on the particular school and its lesson plans, the level of formal knowledge in Social Media Education was highly different. The different knowledge was intended to observe various educational contexts for the tool.

Materials and Methods

Social Media Self-efficacy
For measuring social media self-efficacy a questionnaire with 11 items was created. The scale has a five-scale range from *strongly agree* to *strongly disagree*. First, the Kaiser-Meyer-Olkin (KMO) criterion was applied with a KMO=0.69 indicating that a factor analysis might be conducted (mediocre) (Hair et al., 2006; Kaiser, 1974). According to Kaiser's Rule (eigenvalue criterion) four factors come into consideration. Bartlett's Test of Sphericity was statistically significant ($\chi^2(55) = 434.947$, p ¡ 0.005). Table 3.15 shows question items and the results of the exploratory factor analysis.

Items 3, 7, and 8 build the factor *Social Media Usage*. The items are related to a skill set that is required to maintain a social media profile, communicate with others and create digital content. Items 9, 10, and 11 are building the factor *Social Media Empathy* which can be seen as a reflected and empathetic usage of social media by reflecting on others users' feelings when interacting with them. Factor *Social Media Self-Perception* covers items 1 and 2 that determine how a person perceives their own skills on social media. Finally, items 4, 5, and 6 build the factor *Social Media Literacy* which indicates skills in critical social media usage.

Table 3.15: Results of the exploratory factor analysis with the resulting four factors

No.	Item Description	I	II	III	IV	h^2	μ	δ
	Factor I: Social Media Usage ($\alpha = 0.715$)							
3	I can create and manage my personal profile on social media.	**0.83**	0.01	-0.06	0.05	0.67	0.33	1.0
7	I can create and post text, image, or video content for social networks.	**0.73**	0.02	0.25	-0.25	0.71	0.29	1.5
8	I can communicate with other users through a social network.	**0.81**	-0.11	-0.19	0.12	0.64	0.36	1.2
	Factor II: Social Media Empathy ($\alpha = 0.644$)							
9	When I write comments on a social network, I think beforehand if it may hurt other people.	-0.02	**0.80**	-0.02	0.04	0.66	0.34	1.0
10	Before posting anything on a social network, I consider what other people might think about my post.	0.14	**0.75**	-0.06	-0.19	0.54	0.46	1.2
11	If other people are insulted on a social network, I empathize with that person and try to help them.	-0.20	**0.72**	0.03	0.07	0.59	0.41	1.2
	Factor III: Social Media Self-Perception ($\alpha = 0.655$)							
1	I am familiar with social media.	0.08	-0.05	**0.79**	0.10	0.71	0.29	1.1
2	Compared to my friends, I am more familiar with social media.	-0.11	0.00	**0.91**	-0.04	0.76	0.24	1.0
	Factor IV: Social Media Literacy ($\alpha = 0.507$)							
4	I can assess the consequences that arise when I disclose my data on social media.	-0.02	-0.16	0.06	**0.83**	0.67	0.33	1.1
5	When using social media, I am critical and frequently question content.	-0.03	0.13	-0.05	**0.76**	0.61	0.39	1.1
6	I can recognize inappropriate content on social networks and know what to do with it.	0.27	0.24	0.14	**0.32**	0.37	0.63	3.3

Situational Interest of Tasks

We used an 11 items questionnaire to assess the students' task-related situational interest (Baumgartner, 2014). The questions are answered on a five-scaled Likert scale ranging from *fully agree* to *fully disagree*. The questionnaire measures two constructs: i) task interest and ii) subjective importance. Seven out of eleven items target the task-related interest with a $\alpha = 0.84$. Related studies using this scale show α values between 0.81 and 0.92 (Geißler, 2008). The remaining four items are related to subjective importance. The Cronbach's Alpha in this sub-scale is $\alpha = 0.73$. This sub-scale observes how students rate the relevance of the tasks and their impact on their daily life and society.

Social Media Knowledge Quiz

During the pre and post-test a social media knowledge quiz was conducted. Both quizzes consist of 14 questions that were divided into three categories (technological, socio-cultural, and user-oriented). Each question can therefore be assigned to one perspective of the Dagstuhl Triangle. Table 3.16 shows all questions and the min-max normalized test scores. The questions and answers of both tests are not the same (such as other different choices for password security or other postings when considering if they are valid to post).

Teacher's Interviews

The semi-structured interviews were held with the CS teachers. Ten questions were the base for the interview, depending on the interview partner some additional questions were asked. The pre-defined questions were asked all participants to have a better comparison between all teachers.

1. How long have you been working as a CS teacher?
2. What is your education to teach CS?
3. What was your experience with the SMAwT tool in your lessons?
4. The tool SMAwT is based on the Dagstuhl Triangle (Brinda et al., 2016; Diethelm, 2022), which has three perspectives on digital education.
   a) What do you think the students learned from the tool in the *technological perspective*?
   b) What do you think the students learned from the tool in the *socio-cultural perspective*?
   c) What do you think the students learned from the tool in the *user-oriented perspective*?
5. Do you think the SMAwT tool is suitable for teaching the central concepts

Table 3.16: All questions from pre- and post-test

No	Question Description	Pre-Test (Mean)	Post-Test (Mean)
	Technological Perspective		
1	Which of the following passwords would you consider safe? [Password security]	.823	.920
2	Which indicated that you have been hacked? [Identity theft]	.714	.752
3	What would you do if you think that you got hacked? [Identity theft]	.604	.750
4	Which of the following types of data should not been shared with others? [Data protection]	.754	.954
	User-oriented Perspective		
5	You receive the following request on a social network from a user you do not know. Would you accept it? [Networks]	.847	.509
6	Which of the following statements are true about the following posting? [Tagging]	.723	.713
7	Which of the following statements are true about the following posting? [Hashtags]	.847	.811
8	You receive a chat message from a friend who encourages you to click a link to win a PlayStation. Which statements are true? [Phishing]	.921	.853
9	You deleted messages (photos) in a chat app. Which of the statements is true? [Delete Messages]	.839	.830
10	A friend is going on a journey and would like to post a photo. Do you think he can post this image? [Content Creation]	1	.671
	Socio-cultural Perspective		
11	You are following a YouTube channel and read the following comment. Which statements are true? [Hate Speech or Criticism]	.869	.529
12	A classmate offends another classmate in a class group. What would you do? [Cybermobbing]	.671	.737
13	Read the communication between two users. What should the user do? [Cyber grooming]	.791	.781
14	You read the following post on social media. How would you verify if this is real or fake? [Fake News]	.771	.796

about social media literacy?

6. Would you further use the tool in your classes? In which phase of your classes would you use the tool?
7. Reflecting on the use of the tool in your teaching, what were the positive experiences of the students and in which areas did they face difficulties?
8. What is the ideal target (age) group for the use of the tool?
9. How has the topic of social media literacy been addressed in your classes so far and what are the usual methods of conveying this topic?
10. Are you aware of other tools or methods that are commonly used in this area?

Procedure

The SMAwT study is planned as a mixed methods research design, combining a qualitative part and a quantitative part. For both parts, the SMAwT tool was used in a formal in-class learning activity in CS classes supervised by the CS teachers. The teachers were initially instructed by the project team and the teachers conducted the intervention. The quantitative part of the project was an intervention including pre-and post-test for the students. The qualitative part of the study is focusing on the teacher's perspective regarding *Social Media Literacy* and the *SMAwT tool*. Therefore, semi-structured interviews were held with the involved teachers.

Quantitative Study: School Intervention Two weeks before the actual intervention took place, the teachers received instructions from the project team. It was suggested to spend 2-3 school lessons (á 50 minutes) for all activities including pre-test, SMAwT intervention, and post-test. Further, it was recommended that all three chapters within the tool should be completed during school classes and not as homework. The entire intervention was designed as a single-person work, therefore teachers should provide an environment where each student can work individually. Since the tool includes different types of multimedia all the participants used headphones to not disturb each other. The instructions did not include any guidelines on how the tool should be used in class, to observe different educational scenarios. This means, the teachers could use the tool in any phase they want, but pre-and post-test have to be conducted within the class and the tool should be used at least for 60 minutes. The tests were integrated right into the tool with buttons that link to Microsoft Forms surveys.

Qualitative Study: Teachers' Interviews One week after the school intervention was conducted by the teachers, all involved educators were invited to participate in a semi-structured interview using Cisco Webex. At the beginning of the interview, the teachers were informed, that the interviews will be recorded. The questions in Section 3.5 were the base for the interview. The interviews took between 25 and 35 minutes in total.

Data Analysis The data collection was conducted using Microsoft Forms. Careful handling of student data was the top priority, which is why only a few pieces of information were requested in the surveys. Only gender and age were collected to establish possible correlations between these variables and other factors. The data analysis was performed using the R 4.0.2 Programming Language with Jupyter Notebooks. All interviews were held and recorded by a single interviewer using Cisco Webex. The recordings were manually transcribed and a Qualitative Content Analysis after Mayring (2000) was performed. Two people coded the interviews following an inductive categorization approach for the intended learning objectives mentioned by the teachers as well as the reported learning effects mentioned by the students. Following the process model of the inductive category formation presented by Mayring (2000), research question 3 framed the category definitions ”mentioned learning objectives” for the teacher interviews and ”mentioned learning outcomes” for the student open responses. Single phrases or words were defined as coding units with a medium level of abstraction for the teacher responses and a high level of abstraction for the student responses (as especially students often only gave very short answers). After scanning approximately 30% of the material and formulating the categories, the two raters consolidated each other and revised the categories together. After revising and subsuming the categories again after working through the material, the technological, socio-cultural, and user-centered perspectives were used as the main categories where each inductive category was allocated. As most student responses were single words, the interrater reliability was only conducted for the categorization of the interviews. A moderate Cohen’s Kappa value of .43 (acceptable according to Landis and Koch (1977)) resulted from disagreements between the raters for categories of the socio-cultural perspective and the user-oriented perspective (such as privacy settings in profiles).

3.5.5 Results

RQ1: Improvement of digital literacy

A pressing question after developing a novel tool is its efficiency in a learning context. Before and after the intervention a social media knowledge quiz involving 14 questions was conducted by the students (see Section 3.5). The goal of this quiz was to evaluate the students’ prior knowledge and observe their learning progress after using SMAwT. Even though the question format was the same for both tests, the tests were not entirely equal (i.e. different answers regarding passwords or personal data between the quizzes). To receive comparable data a min-max normalization was applied for each category, which results in normalized points between 0 and 1.

Fig. 3.18 shows a boxplot diagram indicating that the mean values for both tests are almost the same. However, a general conclusion on the total points might draw misleading conclusions. Therefore, the results in regard to their category of the Dagstuhl Triangle were considered. First, a Shapiro-Wilk normality test for the three categories was performed. *Technological Perspective* (Pre-test: $W = 0.964, p < 0.001$, Post-test: $W = 0.928, p < 0.001$), *User-oriented Perspective* (Pre-test: $W = 0.881, p < 0.001$, Post-test: $W = 0.928, p < 0.001$), and *Socio-cultural Perspective* (Pre-test: $W = 0.954, p < 0.001$, Post-test: $W = 0.980, p < 0.001$) are not normally distributed, which requires a non-parametric test. A Wilcoxon rank sum test was used to prove if the results between pre- and post-test differ significantly. The test statistics show statistical significance for the technological perspective ($Z = 590, p < 0.001$), which indicates that the students improved from M=0.738 to M=0.870. However, the categories of user-oriented perspective (from M=0.832 to M=0.801) and socio-cultural perspective (from M=0.805 to 0.686) show no improvement.

Within the pre-test the students were asked i) which social media they know (open-ended question), and ii) which services they are actively using (given choices). To get a broader picture of the actual media usage the data from all participants (n=436) is considered. The results from i) show, that Snapchat (n=256), TikTok (n=231), Instagram (n=208), WhatsApp (n=195), and YouTube (n=193) are the most known social media. The least mentioned platforms are Discord (n=33), Google (n=17), Facebook (n=5), Microsoft Teams (n=5), and Netflix (n=5). In ii) participants responded that they are mainly using WhatsApp (n=390), YouTube (n=376), Snapchat (n=309), and TikTok (n=251). The least used social media platforms are Signal (n=40), Twitter (n=31), and Facebook (n=9).

RQ2: Influence on Learning Outcome

Linear regression was calculated to predict the learning outcome (points of the post-test). A significant regression equation was found ($F(7, 208) = 26.03, p < 0.001$) with an $R^2 = 0.45$. It was found that the results of the pre-test (*pre_test*) have a high significance on the results of the post-test ($\beta = .619, p < .001$). Further, social media empathy ($\beta = .214, p < .001$), social media self-perception ($\beta = .135, p < .05$), social media usage ($\beta = -.112, p < .05$), and subjective importance ($\beta = -.104, p < .1$) have a significant influence as well. Interestingly there is

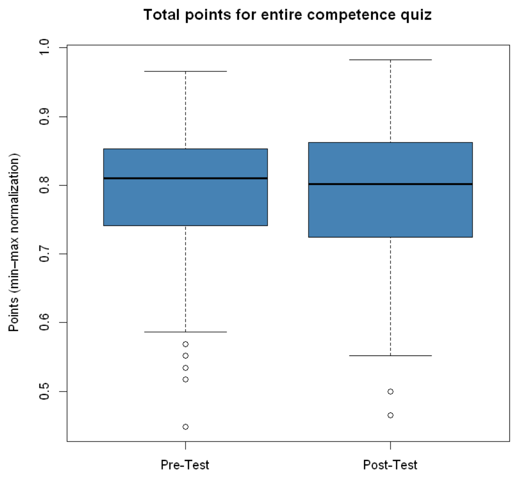

Figure 3.18: Comparison of normalized points between pre- and post-test for the entire quiz. The mean of the pre-test is 0.787 (Sd=0.088) and the mean for the post-test is 0.788 (Sd=0.095).

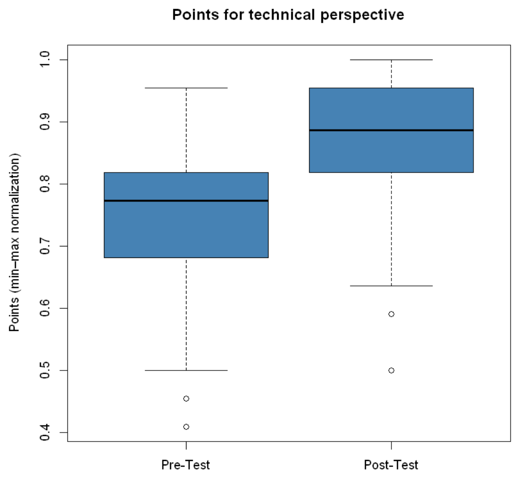

Figure 3.19: Comparing the results between pre- and post-test for the questions of the category *Technological Perspective*. The mean of the pre-test is 0.738 (Sd=0.113) and the mean of the post-test is 0.870 (Sd=0.090).

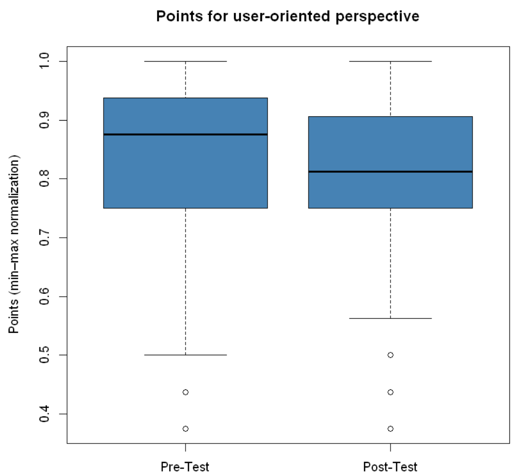

Figure 3.20: Comparing the results between pre- and post-test for the questions of the category *User-oriented Perspective*. The mean of the pre-test is 0.832 (Sd=0.135) and the mean of the post-test is 0.801 (Sd=0.144).

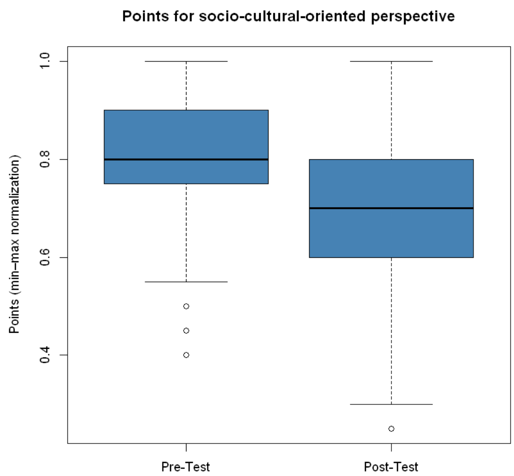

Figure 3.21: Comparing the results between pre- and post-test for the questions of the category *Socio-cultural Perspective*. The mean of the pre-test is 0.805 (Sd=0.125) and the mean of the post-test is 0.686 (Sd=0.147).

no correlation between learning performance and the student's age ($r(214) = -0.196, p = 0.003$), gender ($r(214) = -0.027, p = 0.694$) or daily time spent on social media ($r(214) = -0.054, p = 0.425$) (Hinkle et al., 2003).

RQ3: Learning Objectives and SMAwT

As part of the interviews, the teachers were asked to identify the learning objectives of the SMAwT tool according to the three perspectives of the Dagstuhl Triangle. Even though the teachers were asked one separate question for each perspective, the transcriptions were categorized across the whole part of the interviews addressing the learning objectives (some teachers had difficulties in distinguishing the perspectives). From the interviews, the following teachers' learning objectives (T) of SMAwT could be identified:

1. Technological Perspective
   a) The students can explain algorithmic aspects and underlying systems of social networks.
2. Socio-Cultural Perspective
   a) The students can name potential risks occurring in social networks.
   b) The students can explain the potential consequences of sharing personal data in social networks.
3. User-oriented Perspective
   a) The students can critically use social networks.
   b) The students can name different means of assistance for occurring problems and can name people they can talk to if they encounter difficult situations in social networks.
   c) The students can use common profile functions in social networks.

Not everyone of the teachers identified all of these learning objectives in the tool (see Table 3.17). Only two of the teachers identified learning objective T.1.a that could be allocated to the technological perspective of the Dagstuhl Triangle. All teachers mentioned phrases that could be summarized as learning objective T.2.a.

Table 3.17: Learning objectives identified by the teachers and their allocation to the perspectives of the Dagstuhl Triangle

Teacher (anonymized)	1a	2a	2b	3a	3b	3c	Technological	Socio-cultural	User-oriented
Laila		x	x		x	x		x	x
Jonathan		x	x	x	x			x	x
Samira	x	x		x			x	x	x
Joseph		x		x	x			x	x
Christian	x	x	x	x	x	x	x	x	x

At the end of the student questionnaires, the students were asked what they remember from the activities that they did not know already. Even though the post-test was supposed to give first indications of the learning outcomes, it was intended to observe the students' impression of what topics actually remained in their memory. From the open answers, the following student learning outcomes (S) could be identified:

1. Technological Perspective
   a) I know that a complex password is needed.
2. Socio-Cultural Perspective
   a) I know potential risks occurring in social networks.
   b) I know that I should be careful on social networks.
   c) I know that social media can influence my mental health.
   d) I know that I should not share any personal data on social networks.
3. User-oriented Perspective
   a) I know how I should react when problems occur.
   b) I know how to use profile functions.
   c) I know how I can evaluate the trustworthiness of information on the world wide web.

When trying to map the learning objectives identified by the teachers and the learning outcomes mentioned by the students, it appears that the students remember learning objectives as "rules" while teachers have a rather "abstract" view of the learning objectives (e.g. S2d *"I know that I should not share personal information"* vs. T2b *"The students can explain potential consequences of sharing personal data in social networks"*). The same applies when comparing the teachers' learning objectives and the students' reported learning outcomes to the intended learning objectives of the tool presented in section 3. While the learning objectives defined for the SMAwT modules were competence-oriented and align rather well with the teachers' learning objectives, again, the students' reported learning outcomes seem to rely more on learned rules and expected behavior without further reflection.

Further, not all reported objectives and outcomes can be mapped. For example, *"I know that I need a complex password"* (S1a) from the technological perspective can only be partly interpreted as *"The students can explain algorithmic aspects and underlying systems of social networks"* (T1a) and important concepts seem to be missing here.

Another question that was analyzed was about the teaching phase, in which the teachers would use the tool, for example as an introduction or motivation, for the elaboration of the new learning content, or for exercising/securing topics that they have already learned. All teachers agreed to use it either as a motivational tool or as a learning activity for the elaboration of new content.

3.5.6 Discussion and Limitations

Discussion

The SMAwT tool was developed to foster the students' social media awareness and digital literacy (RQ1). The study's quantitative results show from an overall perspective, that there is almost no improvement. However, the tool's effect on the students gets clearer when looking at the three perspectives of the Dagstuhl Triangle. There is a statistically significant improvement from the technological perspective. This means, that the tool seems to be efficient when it comes to topics such as cyber-security or data protection. On the other hand, there is also a significant decrease in the socio-cultural perspective, which raises one question: Why is there such a high loss of these skills? This can just be answered when looking at the qualitative data in addition.

When relating the results from comparing the teachers' identified learning objectives vs. students' reported learning outcomes to the levels of learning taxonomies, one could argue that students achieved only the first cognitive level of "knowing" certain strict rules instead of achieving the from the teachers desired level of "understanding". This could partly explain the apparently worse scores of the students in the post-test than in the pre-test for the socio-cultural or parts of the user-oriented level: Instead of thinking about and reasoning why they should not, for example, share personal data, the students learned the "rules" to not share anything. This became apparent in the tasks where they should judge whether they would post a photo with a passport or a holiday picture of a city. While many students said in the pre-test that they would post a photo with a passport, most students said in the post-test that they would not even post a holiday picture with the city, meaning that they might have become too critical.

When observing which variables have an influence on the learning outcome (RQ2), it is evident that especially the results of the pre-test are a high indicator for the results of the post-test. Another high impact is on the factors of social media empathy and the students' self-perception. This indicates, that students that reflect on their behavior on social media and critically question their interaction with others, tend to have better results. Interestingly, there is no statistically significant correlation between the students' performance on the post-test and their gender or age. There is also no significance on the students' time spent on social media, which means, that being active on social media does not automatically increase the skills in the three perspectives of the Dagstuhl Triangle.

The learning objectives to be achieved through SMAwT were discussed during interviews with teachers (RQ3). All teachers agree that the ideal age for the tool is between 12–15. Students in this age range have already experience with social media since they are usually actively using different platforms. The teachers identified especially that the tool covers learning objectives within the socio-cultural and user-oriented perspective well. Almost all teachers suggested using the tool at the beginning of social media education. Therefore, SMAwT seems to be a great possibility to let students explore social media phenomena, which can be the base of further discussion in class.

Limitations

Besides its educational possibilities in CS classes, one central aim of the SMAwT tool was to observe its effect on the learning outcome. Since the study was planned within one group with slightly different questions (and points on the quizzes) this had an impact on the comparability of the results. By conducting an A/B test with an experimental and a control group, the effect of the tool could be measured more appropriately.

All school interventions were conducted within the same two weeks. Therefore, the tool was not always ideally integrated into the current curriculum.

In addition, the selection of participating teachers was somewhat biased, as this study primarily involved interested and motivated teachers.

3.5.7 Implications for Computer Science Classroom Integration

Interdisciplinary approaches to Computer Science Education such as the school subject "Medien & Informatik" (Lamprou et al., 2017), in Switzerland or "Digitale Grundbildung" in Austria (Hörmann et al., 2022) show that phenomena of the digital world, such as those occurring in social media, can be taught from different perspectives in single subjects. Our approach was to combine a technological, a socio-cultural, and a user-centered perspective in a tool for teaching about social media. As the results imply that SMAwT increased the students' wariness too much, the tool should not be used as a standalone treatment but should be accompanied by class discussions and reflection phases with the teacher. This aligns with the popular quote of Chris Dede, that *"[l]earning applications are not like fire, a wonderful technology that provides a benefit from merely standing in its vicinity. In education, technologies achieve their power indirectly, as catalysts for deeper content, more engaging activities, more* *active forms of learning and instruction, and richer types of assessment."* (Dede, 2010).

Following the teachers' judgment, the tool should be used either as a motivation or as an elaboration tool during the first hours of teaching the topic of Social Media. Doing so, the tool can be either used as a structure for the topic by completing the levels one by one with complementing activities together with the teacher, or it can be completed as a whole and used as a starting point for the topic, followed by deeper insights through other methods and media. Especially when connecting the tool with other topics of Computer Science Education, such as databases and data security, the interdisciplinary character can be strengthened.

3.5.8 Conclusion and Future Work

This paper presented the development and evaluation of the Social Media Awareness Tool (SMAwT). Different aspects of social media awareness have been observed by conducting a mixed methods design study with students and teachers from six Austrian and German secondary schools.

The tool was designed as a flexible and accessible cross-platform web application. Using a story-based approach the users should be motivated by an engaging real-life story. The tool's modularity allows for composing chapters that contain different mini-games.

While the quantitative evaluation showed inconsistent findings regarding the learning outcomes, the qualitative results provided possible insights into why these findings might have occurred. The performance tests showed significant learning outcomes from a technological perspective, but it seemed that the students got warier than intended regarding all aspects concerning all information found and shared on the world wide web. In this study, the mixed methods approach could enhance the understanding of the quantitative data through the qualitative results. When combining both results, it is suggested to use the SMAwT tool either as an introduction to the topic or as a structuring tool for the elaboration phase of the class.

The study's results indicate, that there is an increase in the area of the technological perspective on media education. However, students' (too) critical attitude after using SMAwT toward the socio-cultural perspective of social media raises the question: *How can students get aware of socio-cultural aspects of social media, with an appropriate level of criticism?*

The role of social media education in the context of computer science classes varies from country to country. Future research will also include other countries as well, to receive a broader picture. Another way to evaluate the effectiveness of the tool is to enable a simulation module, where students can apply their acquired skills in a simulated social network. By observing student performance in this environment with additional interviews, the student perspective can be explored more fully.

3.6 Article 6: Behavioral Pattern Analysis of Programming MOOC

Motivation
Programming courses in higher education open up many possibilities for scientific questions. The course *Learning object-oriented Programming* (LOOP) from the Technical University of Munich is a large-scale course that is held as a Massive Open Online Course (MOOC) using the edX platform. An interesting approach in MOOCs is the learners' engagement during the entire course. Engagement in MOOCs is related to higher completion rates and better academic achievement (Deng et al., 2020). The data from the LOOP MOOC are the base for further analysis regarding behavioral patterns and user interaction within an introductory programming MOOC in higher education.

Contribution
The LOOP MOOC can be seen as a pars pro toto for introductory programming MOOCs in higher education. A heterogeneous group of students from diverse backgrounds attend this course. For this reason, the analysis and findings can be transferred to similar courses. The main contribution of this study was the analysis of the students' behavior and derive concepts for an engaging design of further MOOCs.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - In 2019 TU Munich conducted the LOOP MOOC, a large-scale open online course with approx. 2,500 students enrolled. Similar to related MOOCs this course was characterized by a high attrition rate. This raised the question of how students can be engaged in such courses.
- **Theoretical Foundation** - A literature survey has been conducted regarding MOOC design, drop-out prediction, and behavioral analysis. Related approaches have been identified where data analysis approaches have been applied to analyze the students' learning process.
- **Theoretical Concept** - Based on the literature survey a concept for the data analysis process has been created. The concept has planned that both the behavioral patterns of the students will be analyzed.
- **Implementation** - Within the implementation phase a classifier has been developed to analyze the features that were defined in the concept. The data analysis also considered the educational approach of the MOOC regarding instructional strategies and assessment.
- **Evaluation** - The purpose of the evaluation phase was to validate the find-

ings. By analyzing the findings and the educational context of the MOOC interesting phenomena could be detected.

- **Validation and Best Practices** - The results from the evaluation showed several interesting findings regarding the engaging design of MOOCs. Phenomena such as a *cliffhanger effect* could have been detected.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Validation, Writing - Original Draft, Writing - Review, Editing
- **Christoph Schatz:** Methodology, Software, Formal Analysis, Writing - Original, Draft
- **Johannes Krugel:** Conceptualization, Methodology, Resources, Supervision
- **Christian Gütl:** Supervision

Steinmaurer A., Schatz C., Krugel J. & Gütl C. (2022). *Analyzing Behavioral Patterns in an Introductory Programming MOOC at University Level*. In: Proceedings of the 8th IEEE Learning with MOOCS (LWMOOCS).

Analyzing Behavioral Patterns in an Introductory Programming MOOC at University Level

Alexander Steinmaurer, Christoph Schatz, Johannes Krugel & Christian Gütl

Abstract

Massive open online courses (MOOCs) are an indispensable component in university education today. In large introductory courses especially, MOOCs can promote the efficiency of online teaching tremendously, since a large and heterogeneous group of students can be prepared for further courses and learn self-paced and self-directed. However, MOOCs are also characterized by high dropout rates and with a small group of people only completing the course. In this paper, we analyzed the *Learning Object Oriented Programming* MOOC from Technical University of Munich, an edX course that is dedicated to first-year students in different fields. The course run of 2019 with 2,489 enrolled users is analyzed for this purpose. The dropouts (89 %) were analyzed to better understand future course design. Interaction in the MOOC was considered in this context as a means of detecting behavioral patterns and predicting early dropouts. We found that the interaction with certain MOOC elements such as videos or the problem tool had a major impact on course success. These results may be useful for earlier dropout predictions and the design of future courses to provide an engaging environment with fewer students quitting the course.

3.6.1 Introduction

Massive open online courses (MOOCs) have become a relevant position in online learning today (Kannadhasan et al., 2020). These are not limited to school or university education, there are many platforms such as Coursera, Udemy, or edX that provide courses from a wide range of disciplines. They give users the opportunity to learn wherever and whenever they want. MOOCs can be used to reach a broad audience with defined learning content, but they give also the possibility for individual online interaction such as chats or forums. MOOCs are thus a promising approach for introductory courses at universities, since they are often characterized by a high number of participants with a heterogeneous level of prior knowledge.

There exist a large number of introductory programming MOOCs adressing different goals together with related research at university level is already available. The Technical University of Munich was also faced with the challenge to provide a diverse group of beginners to the same level and created therefore an introductory MOOC (*Learning object-oriented programming* – LOOP) in which students participate before the beginning of the semester.

In this paper, we have analyzed the MOOC interaction data to obtain a better understanding of students' interaction with the MOOC elements and the characteristics of the students involved. We thus defined the following research objectives:

- RO1: Which concepts in the MOOC are causing potential problems and are disengaging?
- RO2: Which features can help to identify students who tend to drop out of the course?
- RO3: What are differences in user behavioral patterns among the various MOOC groups?

The main contributions of this work are to identify problematic features in MOOCs that will influence the student's performance and to analyze the behavioral patterns of the participants. With this work we aim to present concepts to create more engaging MOOCs and to get a better understanding about the MOOCs participants.

This paper has the following structure: Section 2 covers brief research on MOOCs, MOOC analysis, and drop-out prediction in general and also introduces the LOOP MOOC. Section 3 outlines the study, to be more precise, the participants, materials, and the procedure. In Section 4 the study's results will be discussed and finally, Section V concludes the work and gives an outlook on further research.[13]

3.6.2 Related Work

Online learning is gaining increased interest and has been widely adopted in a variety of subjects and settings by introducing the concept of massive open online courses (MOOCs). Movements towards carbon neutrality and the COVID-19 pandemic have been further boosted for this situation. Earliest MOOC examples known includes initiatives by David Wiley in 2007 and George Siemens and Stephen Downes in 2008 (Hernandez Rizzardini et al., 2014). Among the advantages that stand out, literature and practical experiences reveal inclusiveness with fewer social

[13] The section numbering in the original paper is represented as Roman numerals. Due to a consistent number scheme, they have been changed to an Arabic numeral format.

and cultural barriers, free or more affordable education, and student focus on selective topics in a self-regulated learning setting. On the negative side, the most prominent issue is the very high dropout rate as well as the feeling of isolation and disconnectedness many students have (Hernandez Rizzardini et al., 2014; Pelletier et al., 2022).

Given the high dropout rate, research is paying much attention to related aspects. For example, the literature showed a notable discrepancy between usual high numbers of enrollments and the actual lower numbers of active students in MOOCs. Reported completion rates widely vary from 4 out of ten to even as low as under one percent, depending on course layout, subject and the audience addressed. Reasons for dropping out include lack of time as well as inappropriate digital learning skills, insufficient prior knowledge and support (Hernandez Rizzardini et al., 2014; Onah et al., 2014). Based on this situation, the authors in Gütl et al. (2014) have proposed an attrition model for MOOC analysis and group the online learners into three groups: (1) the 'persistence group' which subsumes the 'completers' of the course, (2) the 'unhealthy group' of 'disengagers' who want but struggle to finish, and (3) the 'healthy attrition group' of people who enroll for interest or to access the learning content but do not intend to finish the course. In an orthogonal view, Henderikx et al. (2017) present a topology from the users' viewpoint for determining success and dropout of MOOCs based on their intentions and behaviors. The authors report in their studies of two MOOCs and compare the success rate applying the traditional approach of 6.5 and 5.6% compared to their suggested approach of 59 and 70%.

The situation stated above has motivated many researchers to specifically focus on dropout prediction. Interest includes early detection of dropout candidates, getting insights on behavioral differences between the persistence group and the dropout group, and to identify useful features for monitoring the MOOC users (Gütl et al., 2014; Hernandez Rizzardini et al., 2014). Nagrecha et al. (2017) focused on explainability of dropout prediction models based on existing MOOC dropout prediction pipelines. They conduct a study with click-stream data of a statistics course in edX. In their approach they used, trained and evaluated several models each for i-1 weeks to build the model for week i. Decision tree models have been finally used to highlight the most important features (longitudinal explainers) to predict the probability of completing the course. Vitiello et al. (2018) carried out a similar study applying a boosted decision tree and investigated prediction accuracy and the most important features in this context. Unlike weeks, the authors focused on the prediction performance for each of the first seven active days of the users and certain percentages of overall completion. Prediction performance are comparable and ranges from 65 to over 90% depending on the completion rate. Panagiotakopoulos et al. (2021) report on early dropout prediction after one week of activity through supervised learning and hyperparameter optimization. LightGBM, a gradient boosting decision tree, showed the highest accuracy of 95.6%.

Due to the high demand from businesses and industry but also the increasing importance of digital skills and coding (Pelletier et al., 2022), MOOC courses addressing several aspects from beginners' courses for school kids, to preparation courses for university freshmen to vocational training. Dalipi et al. (2018) administered a review on MOOC dropout predictions and showed that logistic regression, support vector machine and decision trees are frequently used. Among other issues, click-stream data standardization over platforms and generalization of prediction models for real time usage provides room for further research. Vihavainen et al. (2012) report about a beginners Java programming course including common concepts such as expressions, terminal input and output, basic control structures, classes, objects, arrays and strings as well as object-oriented programming features. Design of the course builds on the pedagogical method of Extreme Apprenticeship (XA) and scaffolding of students' tasks with an assessment and feedback system. From 417 registered users, 405 started programming tasks and finally 70 participants finished over 90% of the programming tasks. In a similar initiative, Spyropoulou et al. (2015) designed an introduction MOOC into the programming language C.

Focusing on pedagogic aspects and components applied in coding MOOCS, the follow basically best practices and established knowledge from MOOC research. Inspired by cMOOCs and collaborative approaches, xMOOCs and individual assignments as well as hybrid forms (hMOOCs) offer programming courses in a schedule experience (synchronous MOOC) for several weeks or in a flexible form (asynchronous MOOC) allowing the learners the full control, and the extend of the MOOC can span entire subjects or selected parts (miniMOOC), and can address personalized learning (adaptive MOOC) (Hunter, 2021; Røynesdal et al., 2022). Theoretical frameworks mainly follow connectivism and (social-)constructivism, and learning process usually include (formal) assessment, feedback and guidance and interactive tasks (Stracke & Trisolini, 2021). Specifically, coding MOOCS may include interactive programming tasks and puzzles, complex programming exercises and code reviews (Chakraverty & Chakraborty, 2020; Feklistova et al., 2021; Spyropoulou et al., 2015).

The research community is also showing an increasing interest in understanding the reasons for dropout and behavior patterns in programming MOOCS. Oeda and

Hashimoto (2017) report their results on student dropout prediction and uncovering programming issues based on commands in the programming environment and how to interact with source code editing. They apply dynamic time warping and k-means clustering on an input of 39 programming beginners during a 90 minute lecture session. The results revealed 3 outlier clusters with fewer, more and increasingly rapid input of commands. Yan et al. (2017) conducted research in the context of the Gidget Programming Game and predicted the dropouts for each of the first five levels of the game. They applied a Gradient Boosting Decision Tree for dropout prediction with an average accuracy of 68. Inspired by the cliff-hanger effect in media consumption, C. Chen et al. (2020a) showed a similar effect by overshooting the dropout rates at the end of topical chapters.

3.6.3 Background

The research in this paper is based on collected data of the introductory programming MOOC called *"LOOP: Learning Object-Oriented Programming"*. The course was initially developed at the Technical University of Munich because computer science education in school varies strongly, which is why the prerequisite knowledge of freshmen at universities is very inhomogeneous (Krugel & Hubwieser, 2017). As students cannot be expected to be present at university before lecturing starts, MOOCs seem to represent potential solutions to compensate or reduce these differences. The primary target group of LOOP is comprised of prospective students of science or engineering who are due to attend CS lessons in their first terms. However, the course is freely available online via edX, and the content of this course is provided in German. Since the course is intended for beginners it can also be attended by a worldwide, German-speaking, audience with a more diverse background that is not limited to students of the Technical University of Munich.

The course follows a "strictly object first" design and consists of the following six chapters and 18 sections (the last chapter was added in 2019) (Krugel & Hubwieser, 2020):

1 Object-oriented modeling
- 1.1 Objects
- 1.2 Classes
- 1.3 Methods and parameters
- 1.4 Associations
- 1.5 States of objects.

2 Algorithms
- 2.1 The concept of algorithm

- 2.2 Structure of algorithms.
- 3 Classes in programming languages
  - 3.1 Class definition
  - 3.2 Methods
  - 3.3 Creation of objects.
- 4 Object-oriented programming
  - 4.1 Implementing algorithms
  - 4.2 Arrays.
- 5 Associations and references
  - 5.1 Aggregation and references
  - 5.2 Managing references
  - 5.3 Communication of objects
  - 5.4 Sequence charts.
- 6 Inheritance
  - 6.1 Specialization and Generalization
  - 6.2 Polymorphism

The MOOC is based on a constructivist pedagogy and consists of a series of short videos, quizzes, and interactive exercises in each chapter, and a concluding final exam. The course addresses different learning preferences and impairments by providing the learning content as visual (videos, graphics), textual, and audio presentations. A focus of the course design is the various interactive exercises for enabling the learners to experiment with the presented concepts. Such interactive elements are for example visualizations that illustrate the flow of a source code or an algorithm. Furthermore, the chapters contain programming exercises with constructive feedback for the learners using a web-based integrated development environment and additionally an automatic grading system. A detailed description of the course design and syllabus was published in Krugel and Hubwieser (2018).

A course run takes six weeks (one week for each chapter) and the targeted workload of the learners is 5-10 hours per week. The communication among the learners and with the instructors takes place entirely in the discussion forum. The main task of the instructor during the conduction of the course is to monitor the forum and to react accordingly, e.g., answer questions or help with technical problems.

The course runs are offered publicly as MOOCs on edX and are included in the global edX course catalog in the category *Computer sciences courses* and are freely available worldwide.

3.6.4 Study

Within the scope of this study, we have used the edX log data from the course run in 2019. This JSON-based data contains all user interactions during all course activities. This study aims to conduct a behavioral analysis on the users to identify groups of participants and specific features related to them. Therefore, a classifier was trained and tested on the given data. This research project should help to early detect patterns that are related to dropouts to support students during the MOOC.

Materials and Methods

An introductory online questionnaire was integrated into the course (with the title "course start survey" in the following), in which the participants were asked about their age, gender, major, and CS school education. Three options were offered for assessing the previous programming experience: i) *none*, ii) *little*, and iii) *advanced* (for a program written with more than 100 lines of code). The following four options were provided for assessing the intention regarding the course completion: *I want to take a look at the course. / I want to work on some topics relevant for me. / I want to work on most topics. / I want to complete the whole course.*

In a concluding questionnaire at the end of the course (referred to as "course end survey" in the following), the participants were asked for positive and negative textual feedback regarding the course; it consists of two text fields asking the learners for positive and negative aspects discovered during the course.

The main basis of the further analysis was the detailed edX log data which can be assessed by partnering institutions. The data is provided in JSON formatted text files and was pseudonymized during the preprocessing step.

The platform edX supports different tools that can be used to create an online course. Even though the MOOC had various tools, we analyzed the four primary used ones. The tool *Learning Management System* (LMS) includes all the course content and is essential for navigating with the other tools. The *Video* tool is used for the interaction with a video. The interactions provided are pausing and playing, changing speed and position and enabling or disabling transcriptions. The *Problem* tool is comprised of quizzes and also programming exercises. The tool is responsible for handing in submissions for the practical coding tasks. Finally, the *Forum* is used for the interaction between trainers and course participants.

In this work, some definitions about user groups were made. *Completers* are users that obtained at least 50% of the possible points in at least 14 of the 18 courses

Table 3.18: Summary of the MOOC's users, droputs and dropout rate considering only active users (considering enrollments in parentheses).

Enrollments	Active	Inactive	Completers	Dropouts	Dropout Rate
2,489	2,059	430	276	1,783 (2,213)	87% (89%)

units. *Dropouts* are the group of users that did not reach 50%. We further define *Problem Attempters* as users that had at least one interaction with the problem tool. *Active users* are users with more than just the enrollment action. For all further analysis enrollment is not considered when talking about actions or tools since it does not hold any information value. Therefore, *Inactive users* can be described as users without any actions. The numbers of the analyzed course run are given in Table 3.18.

To analyze which behavioral patterns separate the group of completers from the group of dropouts the term *session* was defined as a continuous flow of actions that take place without an interruption of more than 30 minute periods. Finally, the dropout rate is defined as number of dropouts divided by enrollments.

Participants

The course run 2019 of LOOP attracted 2,489 registrations (see Table 3.18). We received 936 responses for the course start survey (female: 260, male: 566, diverse: 6, no answer: 104) with a diverse study background (numerous different majors, including Computer Science, Management, Engineering, Mathematics and many more). The participants were mainly from Germany, but in total from more than 70 countries. The average age was 21,7 with a standard deviation of 6.3. In the programming context, 281 participants had no experience, 393 had basic knowledge, and 128 participants had already written a "bigger" program of at least 100 lines of code (no answer: 134 participants).

The Technical University of Munich announced the courses on its official Facebook page and informed all students in CS-related subjects about the course offerings by e-mail. Everyone was free to participate in the courses without any formal or content-specific prerequisites. The only requirement was German language proficiency since the course was offered in German. Participation was voluntary in all course runs and did not count towards a grade. edX issued verified certificates for successful participation (= obtaining in sum at least 50 % of the possible points of at least 14 of 18 course units, for further information see enumeration in Section II.B). Those certificates, however, had to be paid for, except in the case of persons affiliated with Technical University of Munich.

Procedure

The MOOC consists of various tools such as a forum, videos, and problem tool, and based on this a *transition* is a sequence of two actions where the user switches from one tool to another tool. All the transitions were added up for the group of completers and dropouts. In a first approach these values were normalized for the number of users in each class (per user average transition). In the second approach they were normalized for the number of transitions in each class (transition probability difference). The matrix of the dropouts was subsequently subtracted from the matrix of the completers for both approaches. This resulted in a matrix in which high values mean that completers interact more with the tool and negative values mean dropouts have higher interaction.

A set of features was defined to characterize each user. Features that have either been used heavily, or have been shown to differ in usage between completers and dropouts were used. The value of a feature was calculated by counting the number of times a user interacted with it. This set was expanded with time-based features describing the sessions. The first time-based feature is the average time-span between clicks. Active time is the sum of all sessions duration. The average session length is defined by active time divided by number of sessions. Another feature are daily requests which are the number of actions normalized by the number of days a user interacted. And lastly the average session actions, are the number of actions divided by number of sessions.

We looked at two approaches for early dropout prediction. One approach is to take the data of the first seven days of the MOOC into consideration, and the second is performed on the entire dataset. We wanted to learn if it is possible to make a good early prediction. We created a subset of data for each approach and then used it as input for the classifier, which uses a stratified shuffle split and a boosted decision tree for oversampling SMOTE (Synthetic Minority Oversampling Technique) (Chawla et al., 2002). An AUC-ROC curve was used to evaluate how well the classifier performs. The resulting weights show us which features are meaningful for early dropout prediction.

3.6.5 Results and Discussion

The dropout rate is the same for males and females with only a small subgroup not answering or choosing divers. Previous programming experience however has an impact on the dropout rate with it decreasing to 73% for people with programming

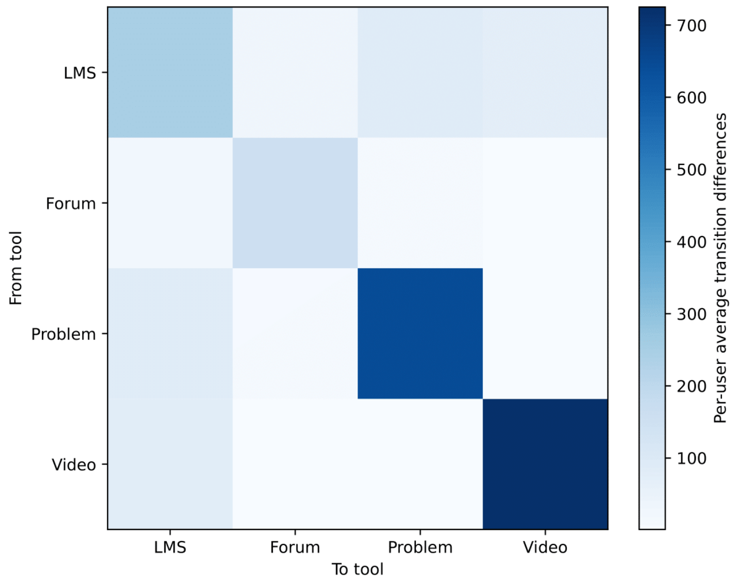

Figure 3.22: Per user (active users) average transition matrix, whereas high values indicate big difference between completers and dropouts.

Table 3.19: Overall feature usage of the course.

Forum	LMS	Problem	Video
396,827	215,516	216,263	67,020

experience. While users with no experience suffer a dropout rate of 86% and with basic knowledge 75%.

When analyzing behavioral patterns there is a strong correlation to the results found in Vitiello et al. (2018) where, the main differences are found with the LMS, video and problem tools. However, we also found a noticeable difference in transitions between completers and dropouts in the forum tool. As Table 3.19 shows, the forum tool is used the most in terms of absolute interactions (396,827 interactions).

Interestingly Fig. 3.22 shows that the video tool seems to be used more by the dropouts while Fig. 3.23 suggests the opposite. An explanation therefore is, that while completers interact with all tools more in terms of absolute numbers, dropouts use the video tool proportionally more frequently. When normalizing with the total interactions it seems likely that the dropouts engage more with the video tool than the completers, but they interact far less frequently with the other tools. Both figures show that the problem tool seems to be one of the most reliable indicators of whether someone is going to complete the course or not (RO1).

The data analysis regarding the behavioral patterns on the actions of all tools show,

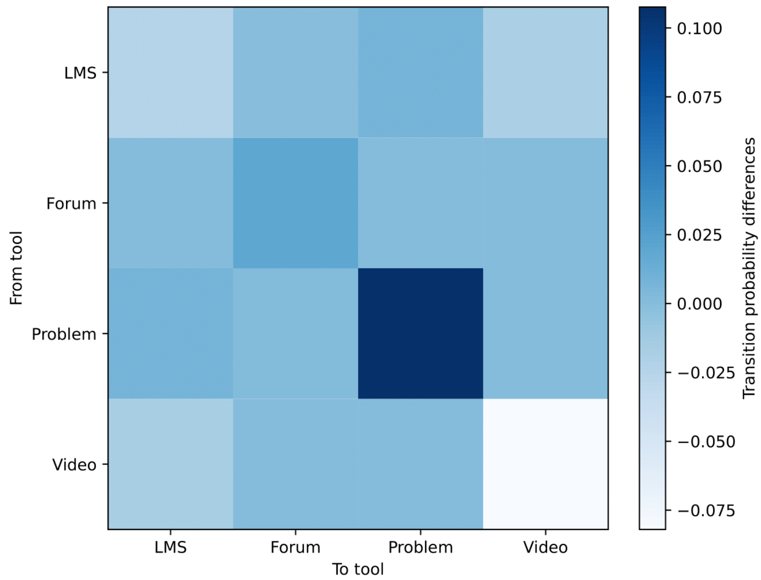

Figure 3.23: Transition probability differences matrix for active users.

that completers follow a more structured approach to MOOCs while the dropouts seem to interaction with the less important features, this is especially pronounced in the video tool.

A classifier using a boosted decision tree was implemented to predict potential dropouts early. The trained classifier already had a high accuracy with the given MOOC data with already little input. 75% accuracy was already achieved from the 3rd day and 85% on the 7th day of the course run. We consistently obtain a slightly higher accuracy when considering all users and not just active users.

Table 3.20: Weights of the different features used for the classifier.

	Days			Percentage		
Feature	**1**	**4**	**7**	**5**	**50**	**100**
Timespan Clicks	0.17	0.15	0.11	0.11	0.09	0.01
Delta Time	0.01	0.02	0.03	0.14	0.13	0.11
AVG Session Actions	0.19	0.17	0.15	0.01	0.06	0.03
Problem Submitted	0.20	0.22	0.12	0.11	0.28	0.36
Video Played	0.11	0.06	0.05	0.07	0.05	0.00

The weights of the classifier comply with the results of the behavioral patterns analysis, we find high weights for the problem features, especially the problem submitted feature is noticeably one the completers interact more with. This could be observed for both early predictions and later on in the course. We also see an interesting weight distribution for the time span clicks feature, which has a high

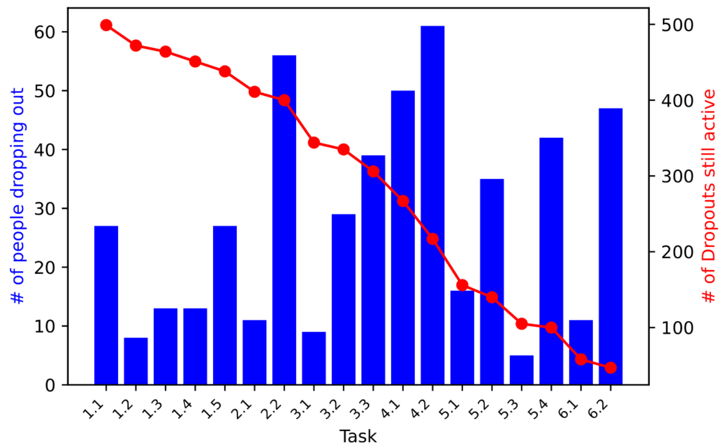

Figure 3.24: This figure shows the sections after which users drop out of the course.

weight when considering only little data, but decreasing, however when more data is being considered. This can be put down to the fact that when first enrolling in the course there seems to be a less structured approach from dropouts where they simply explore the course page with a great many of clicks, which leads to a small time span between the clicks. The same observation can be made on the average session actions. These two features are therefore especially interesting for early dropout prediction. The other session related features are not particularly consistent for predicting completion. The sum of the time the users spend with the course, referred to as delta time, has a high weight when using more of the data, however it does not appear to be very useful for early predictions (RO2).

Dropouts overall have noticeably less interactions than the completers even early in the course. This can allow us to identify this subgroup of users who do not interact a lot with the course early (RO3).

Fig. 3.24 shows after which course section users gave up on the course. When summing up for chapters it is clear that the fourth chapter (*object-oriented programming*) appears to be the most challenging one. The representation also shows a so called *Cliffhanger effect* (C. Chen et al., 2020b; Höfler et al., 2017; Yan et al., 2017) meaning that people tend to complete a whole chapter and then will not come back afterwards. This effect can also be seen at the transition of nearly all chapters (in the course sections 1.5, 2.2, 3.3, 4.2, and 5.4).

We also took the motivation of the users into consideration, and we re-run everything only taking into consideration those users who had intended to complete the course according to the survey that was conducted when it began. This proved to be useless, however, since we ended up with the same results for all plots when comparing to the run on all the active users. This method will thus not be included in the paper.

3.6.6 Conclusion and Future Work

In this work we have analyzed the MOOC data of an introductory programming course. We focused on analyzing drop-outs and behavioural patterns within the different user groups by analyzing various features.

The results of the data analysis showed that there are huge differences between completers and dropouts in the interaction they have with the different tools. Completers tend to use various tools while dropouts use the video tool proportionally more and make less use of other tools. One of the most reliable indicators for a successful course is the problem tool, since it is mainly used by completers already at an early stage.

An interesting observation is the transition between the different chapters where the number of dropouts have a peak. This phenomenon is already well documented already in the literature (*Cliffhanger effect*) (C. Chen et al., 2020b) and reveals room for improvement in MOOC design.

For future work we plan to collate our quantitative findings with qualitative findings of Krugel and Hubwieser (2020) and furthermore include an analysis on the submitted code, to obtain a better understanding on the participants gained coding skills. Besides the source code we also want to include features such as the forum and the postings. A time series analysis should therefore provide further insights into user groups and behaviour. This further data analysis should be applied to course re-runs of different years to compare the data and also to have a suitable model for dropout prediction in future years.

3.7 Article 7: Online Teaching in Introductory Programming Courses

Motivation
MOOCs are one possible format to deliver a university-level course about programming fundamentals. Another way to provide an online course is a blended learning course. The course *Introduction to Programming* at Graz University of Technology was designed and held as a fully online course. Figure 1.1 shows how instructional strategies, teaching and learning concepts, online learning platforms, and educational assessment can shape engaging learning experiences. Within this course, elements such as online streaming platforms, discussion forums, semi-automated assessments, and learning management systems are used to provide an engaging course.

Contribution
This course was re-designed and evaluated from an existing traditional university-level course. The main goal of restructuring the course was to increase the student's satisfaction and academic achievements. Different aspects of the course have been evaluated, such as the acceptance of an entire online course, the course satisfaction, and the students' attitude towards blended learning formats. The findings are the base for further implications for researchers and practitioners to design engaging blended course formats.

Research Methodology - Engaging Learning Tree

- **Initial Motivation** - The starting point of this study was whether a large-scale programming course with over 800 students could be delivered entirely online and simultaneously increase the students' satisfaction.
- **Theoretical Foundation** - A comprehensive literature survey has been conducted regarding instructional strategies, online platforms, assessment methods, and learning taxonomies.
- **Theoretical Concept** - The conceptual model comprises these perspectives and related approaches from other course formats. An educational concept was proposed combining several teaching and learning concepts, online learning platforms, instructional strategies, and educational assessment.
- **Implementation** - This concept was implemented in a course run in the winter semester of 2020 at Graz University of Technology.
- **Evaluation** - The entire course was evaluated over the whole semester using qualitative and quantitative data.
- **Validation and Best Practices** - The results of this study represent recommen-

dations for engaging online formats in higher education.

Authors

- **Alexander Steinmaurer:** Conceptualization, Methodology, Investigation, Visualization, Validation, Writing - Original Draft, Writing - Review, Editing
- **Christian Gütl:** Supervision

Steinmaurer A. & Gütl C. (2022). *Implementation and Experiences of a Flipped Lecture Hall – A Fully Online Introductory Programming Course*. In: Proceedings of the 25th International Conference on Interactive Collaborative Learning (ICL).

Implementation and Experiences of a Flipped Lecture Hall – A Fully Online Introductory Programming Course

Alexander Steinmaurer & Christian Gütl

Abstract

The course Introduction to Programming is one of the first and fundamental courses within any computer science-related study program. Traditionally, such introductory courses are characterized by a large group of students, whereas this group has a heterogeneous prior knowledge of the topic. These courses are usually taught in a traditional setting due to a high number of participants. However, the Covid-19 pandemic situation required to shift from traditional teaching to alternative approaches. In the winter semester 2020, a total of 636 students actively participated in the course at Graz University of Technology. Therefore, the course was revised to a fully online flipped classroom course using asynchronous elements such as pre-recorded videos and synchronous elements such as live streams on Twitch. In this paper, we show how we implemented a fully online course using the flipped classroom approach. We present approaches that engage students in active participation and encourage self-paced learning. We found that a high community-related interaction with students has a major impact on students satisfaction. This can be reached using lively communication and different communication channels. These results may be useful for researchers and lecturers that want to have insights into experiences in flipped classroom settings.

3.7.1 Introduction

The Covid-19 pandemic forced many schools and universities to reconsider traditional learning situations in just a short period. The range lasted from recordings out of an empty lecture hall to comprehensive e-learning courses. Within the scope of our course, we have decided to revise the structure of the existing course towards a flipped classroom setting. Therefore, we provided the students with short videos about a particular programming topic at the end of each week. We used the lecture recordings from the previous years and created small junks of programming concepts. Besides this asynchronous activity each week a live stream took place via the Twitch streaming platform. Within these streams, the lecturers presented further information and live coding examples. A key aspect of the synchronous sessions was a direct and immediate interaction between lecturers and students using the platform's chat. With this course design, we aimed to give the students a unique learning experience with both asynchronous elements to learn concepts and synchronous elements to interact with the lecturers.

- RQ1: What is the overall satisfaction of students attending the flipped classroom course?
- RQ2: What blended learning elements have an impact on the students' satisfaction with the flipped classroom course?
- RQ3: Which course elements are perceived as positive and negative for students in a fully online course?

The main contribution of this research paper is to present the design of a fully online introductory programming course. We want to share our experience to give researchers and lecturers insights into the experience the authors made during the course. By showing advantages and disadvantages based on the students' feedback we will present best-practice examples.

The paper is structured as follows: In chapter 2 we will give an overview of the background and related work. Chapter 3 will introduce the university course and will outline the structure of the lecture and the practical part. This chapter will also cover the used technologies for the online course. During the semester two evaluations have been conducted. This paper focuses on the evaluation aspect of online learning, which is covered in chapter 4. Chapter 5 presents the findings of the evaluation and discusses the results. Finally, we conclude the findings and show some limitations.

3.7.2 Background and Related Work

The concept of flipped (or inverted) classrooms shifts from a teacher-centered to a highly learner-centered form of learning. To flip the classroom means, that concepts are learned at home (instead of in the classroom) and practical tasks (such as examples or homework) becomes a part of the in-class activities (Bergmann & Sams, 2012). This gives the students the possibility to self-pace their learning progress, increase student-teacher interaction, and also to practice problem-solving skills. On the other side, this requires more responsibility for the students since they might have a higher workload at home. A flipped classroom concept also demands more preparation time for educators (Kovah, 2014). The concept of flipped classroom is a form of blended learning. The term blended learning means that traditional classroom methods are combined with online learning (Bonk et al., 2006).

There are various examples in different subjects of higher education where elements of blended learning are used. This ranges from a scale of a nearly full flipped classroom to courses with just some elements. Ng (2018) used the flipped classroom concept as part of the course *Information Technology in Education* for one learning concept (photograph editing techniques) with 73 participants. The students were provided with videos and lecture materials instead of step-by-step instructions during class. The students answered that they were positive about the flipped classroom. A finding was that more interaction is desired during online phases.

In 2019 Alammary (2019) conducted a systematic literature review on blended learning models in introductory programming courses. The results showed, that most studies report a positive impact on teaching and learning. Especially in programming courses elements such as online programming environments can be used to motivate and engage students, or interactive videos can help to spend more time with learning content. Even though many studies report the benefits of blended learning, there are also some challenges. Breimer et al. (2016) suggest that instructors should find the right balance between online content and face-to-face interaction, to support various types of learners. Especially in programming courses the skills in problem-solving should be trained. This can be improved using (interactive) videos to demonstrate steps and introduce concepts and also add face-to-face activities to train the application of these concepts. An important part of a flipped classroom is lecture resources. These resources can be slides, documents, videos, etc. The subject of learning videos is already well observed and gives researchers and lecturers extensive insights. The engagement in short videos (around 10 minutes) is much higher (Albrecht et al., 2018). Since more complex topics require more explanation the length of videos also tends to increase. To tackle drop-outs interactive videos can be created. These videos can include elements such as questions to engage students. A major advantage of videos is that students can re-watch the videos (especially for exam preparation) and work at their own pace. Videos can also help to modularize topics which often gives a clear structure (Clark et al., 2016).

3.7.3 Course Outline

In the previous years, this lecture was held traditionally in the lecture hall and the practical part was given by tutors with about 20 students per group. Each tutor held a weekly class where the concepts from the lecture were applied by live coding. The classes were organized decentrally which means every tutor prepared his or her teaching resources. The assignments were equal for all groups and graded by their own tutor. After each assignment so-called assignment reviews took place

Week		Lecture		Practical	
1		Kick-Off		Kick-Off	Evaluation 1
2		Compiler, Hello World, Variables		Kick-Off A0	
3		Operators I, Characters, Casts		Q/A A0	
4		Operators II, Contionals, Lopos		Kick-Off A1	
5		Functions, Recursion		Q/A A1	
6		Arrays, Strings		Deadline A1	
7	Twitch Stream	Pointer, Preprocessor	Twitch Stream		
8		Stack and Dynamic Memory		Kick-Off A2	
9		Heap, Structs, Enums		Q/A A2	
10		File/IO, CLI		Deadline A2	
11		Exam Preparation		Kick-Off A3	
12		Christmas Lecture			
13				Q/A A3 Deadline A3	
14				Grading	
15					Evaluation 2

Figure 3.25: This figure shows the structure of the course over the whole semester.

where students got asked about their submissions in a face-to-face meeting.

Due to the Covid-19 pandemic, the course team decided to fully revise the existing concept and implement a fully online, flipped classroom setting for the whole course. To tackle the challenges in online teaching we applied the following main components to the concept:

- Provide a central information platform for all students to share all relevant information and keep a steady information flow.
- Implement a flipped classroom to allow students a self-paced form of learning by integrating asynchronous elements (lecture videos) and synchronous elements (live streams).
- Enable a high level of interaction with the whole course team in both the synchronous and the asynchronous phases.

Structure of Lectures

At the beginning of the semester, the students were informed about the (flipped classroom) lecture mode. The lecture consists of asynchronous and synchronous phases. The students were provided with asynchronous content using the university's learning management system (LMS) Moodle. Within this system, short video sequences (from 5 to a maximum of 20 minutes) to a certain topic were unlocked at the end of each week. Since the lectures from the previous years were already recorded, video content was already available. These lectures were cut into small learning units. Additionally, other interesting resources (websites, code snippets, or videos) were uploaded into the LMS. The synchronous part of the lecture was a weekly live session that took place every Thursday. The lecturers decided to use a video streaming platform that enables a high degree of interaction with the participants (Pirker et al., 2021). Therefore, the video streaming platform Twitch[14] was used. During a 45 minute live lecture stream the topics from each week's asynchronous videos were summarized and selected aspects were presented and discussed.

To actively participate in the flipped classroom setting the students had to prepare for each live session by watching the videos in the LMS in advance. At the beginning of each live stream, the lecture team briefly summarized the most important concepts, usually by quizzes using an audience response system. The live sessions were held in a streaming studio in team teaching, which means that both lecturers were streaming together. One lecturer was in charge of presenting the content while the other was responsible for the communication and interaction with the audience; the roles alternated weekly.

At the end of the semester, the lecture part was completed with a final exam. The exam was divided into a code analysis part (five questions regarding a given source code) and a code production part (solving a modified Parson Puzzle (Parson & Haden, 2006) and writing a simple program). Within the last years, the exam mode was pen and paper. Due to the online setting, the exam was integrated into the LMS as a quiz activity. The exam was held in a closed-book mode and supervised by tutors in groups of 8-12 students each using Cisco Webex.

Structure of Exercises

The practical part of the course also started in week 1. The exercises aim is to apply the concepts that were introduced in the lecture. Therefore, the students work individually on four assignments. The scope and level of complexity increase for each assignment, with a maximum of 100 points. During the whole semester, an online stream was held by three tutors via Twitch, which was organized similar to the lecture stream. The stream covered organizational aspects such as an introduction

[14] https://www.twitch.tv/

for the assignments and Q/A or live coding sessions for practical topics.

The very first assignment (A0) was not graded and aimed to make students familiar with the submission system (GitLab) and the development environment. Assignment 1 (A1) was worth 27 points where the students had to implement a simple gross-net calculator. The related concepts included all topics that have been introduced until week 5 (variables, casts, control structures, and functions). Assignment 2 (A2) was a symmetric encryption algorithm (Playfair) with 33 points in total. Therefore, students had to apply all concepts from A1 and also arrays, strings, and pointers. The final assignment (A3) was to implement a game in the command line that was inspired by Pipe Mania. The required concepts covered all topics from A1 and A2 and dynamic memory management, structs, and file organization. This assignment was worth 40 points in total. After each assignment an assignment review took place where the students had to explain their code and make modifications on their programs.

Technologies

Live Streaming Platform. Both the lecture and the exercise were streamed via Twitch. Originally Twitch is a platform that is closely related to gaming. It is especially known for a lively conversation with the community and a high level of interaction between streamers and the audience. Due to Twitch's interactive and open character and the high number of participating students, the course team decided to use this well-known and established platform.

In each stream (lecture and exercise) at least 3–4 moderators (members of the course team) and one of the streamers were actively participating in the chat moderation. They tried to encourage the students to join discussions and ask questions. Unfortunately, not all users are intended to promote a constructive and friendly interaction. Many streamers have also reported toxicity, harassment, and spam (Pirker et al., 2021). To tackle these issues clear rules were defined at the beginning of the semester. Behavior that is contrary to these rules will be sanctioned by the moderators by blocking or banning the users. Twitch provides various built-in functionalities to interact with the audience (f.i. chat), but it can also be extended with plugins. The poll functionality (Staw Poll[15]) provides additional interactive elements to the chat. Streamers can create polls and show the results using the chat while users can vote in the poll.

Audience Repsonse System Even though the Twitch chat gives a lot of possibilities, it is also limited in the level of interaction. To circumvent this problem, the Audience

[15]https://www.strawpoll.me/

Response System (ARS) Mentimeter[16] was used. The tool allows to create interactive slides that can be displayed by the presenters via the browser or even integrated into PowerPoint. There are different interactive slide types such as quizzes (single- or multiple-choice answers), Q&A questions, scales, or rankings. Users can join a presentation by opening the website and entering the code. During a certain (interactive) slide (for example a single-choice quiz) is opened, the users can use their devices to answer. All answers are immediately displayed on the presenter's slide. This promotes a lively and interactive discussion between the audience and lecturers since responses can be discussed. Another benefit is to improve the students' self-reflection since they receive immediate feedback and get a feeling regarding their learning progress.

Learning Management System. A vital component of the course is its Learning Management System (LMS). The university provides a customized instance of the Moodle system. All relevant information was shared on the platform including all lecture slides, assignments, forums, lecture videos, video on demand (VoD) streams, and organizational details (grading, coding style guide, etc).

Another benefit of the LMS is the grading. The course team created grading reports for the students. All activities (assignments, assignment reviews, or exams) are added to the report including the corresponding points. This gives the student permanent information on their performance in the course since all points are published after the activities are graded.

Test System. The test system is a central component of the exercises since it helps students with their coding assignments. The students have a local instance of the testing system on their machine which they can use for testing. For each assignment, a certain number of public and private test cases are defined, which determine the number of points for an assignment. Using the test system the students can test their submissions against the public test cases to see how many points they would receive. Additionally, they receive a detailed report for each test case that should help the students to identify errors, warnings, and memory leaks.

3.7.4 Evaluation

The Covid-19 pandemic situation was a driver for digitalization in many areas. In the past years there were already some ideas to revise the course *Introduction to Programming* at Graz University of Technology, which was highly accelerated to the closing of universities. To get detailed insights into the revised concept, the online learning, and the students' perception of the course, two evaluations have been

[16] https://mentimeter.com/

conducted. This study aimed to get valuable feedback from the students voluntarily. The objectives of the research team were to find out the students' satisfaction related to flipped classroom and to identify elements that have an impact on the level of satisfaction. Additionally, it should be observed which course elements have a positive and a negative impact on the online course. This research should help to gain a better understanding of online learning and to further develop the course to provide an engaging learning environment.

Participants

In total 636 students actively participated in the course, which means they submitted at least one assignment and received a grade. Overall 188 students (26 female and 162 male) completed both evaluation activities. The course is mandatory for all students in the following study programs: Computer Science, Software Engineering and Management, Information and Computer Engineering, Electrical Engineering, and Teacher Training Programme Computer Science. Due to data privacy reasons, the students' ages could not have been collected. However, since the course is a compulsory subject in the first semester, the majority of the students are freshmen.

Materials and Methods

The students were asked to participate in two evaluations, the first one at the beginning of the semester (before starting the first assignment) and the second one at the end (after the final assignment). The first questionnaire mainly covered questions regarding the student's prior experience, their available learning resources, mindset, motivation, and learning strategies. The second questionnaire at the end of the semester contained questions related to collaborative learning, their self-evaluation of learning, and the e-learning context in particular. Participating in the study was fully voluntary, but as a reward, the students received bonus points.

Within the scope of this study, the results from the online-related questions have been analyzed. Table 3.21 gives an overview of the 18 items of the online questionnaire, which was created by the project team. The scales are from 1 (strongly disagree) to 6 (strongly agree). The questionnaire shows a high internal consistency with a Cronbach's α of 0.88.

Procedure

The introductory course started at the beginning of the semester in the first week of October and lasts until Mid of January, with 15 weeks in general (excluding Christmas break). Before the first lecture, the students were informed about the course's procedure via email. All relevant information was announced in the Moodle

Table 3.21: This table contains all 18 items from the questionnaire related to online communication.

#	Question	Mean	SD
1	Easy and quick exchange of information and knowledge (via e-mail, chat, forum, etc.) with other students.	4.94	1.18
2	Promotion of joint learning with other participants.	4.13	1.45
3	Personal contact with other students.	4.12	1.52
4	Self-determination of learning path, learning strategies and learning speed.	4.87	1.10
5	Opportunity to practice and work independently.	5.17	0.96
6	Opportunities to review one's own learning progress (e.g., through self-testing).	4.96	1.14
7	Support of own motivation (e.g. by teachers, through feedback, self-tests, etc.)	4.74	1.22
8	Multimedia-based, varied communication of learning content (via images, videos, animations, etc.)	4.54	1.33
9	The ability to decide for myself when and where I learn.	5.11	1.02
10	Clarity and clear structure of the course and learning materials.	5.38	0.97
11	Completeness and timeliness of learning materials.	5.38	0.96
12	Wide range of communication services for exchange with other participants (e.g. e-mail, chat, forum, etc.)	4.54	1.26
13	Rapid feedback from teachers (e.g. via e-mail, forum, etc.)	4.98	0.95
14	Learning supports and teacher guidance as appropriate.	4.71	1.04
15	A good and comprehensive introduction to the course (info about the learning platform, course organization, etc.)	4.94	1.09
16	Possibility of personal contact with teachers.	4.19	1.37
17	Good accessibility of teachers via e-mail, chat, forum, etc.	5.02	0.93
18	Extensive knowledge of teachers with regard to the design of media-based events.	5.01	0.97

system and in the first lecture stream. During the first lesson, the students got information regarding flipped classroom and the synchronous and asynchronous course elements.

The first evaluation was conducted at the beginning of the first semester and was available until the first assignment started. The second evaluation was conducted at the end of the semester after students submitted all assignments and received their total points. Both evaluations were voluntary, but they received bonus points in case both evaluations were filled out. All data was collected using the university's LimeSurvey service. The students were informed that the analysis has no impact on their grades and that the raw data is processed by a project partner to guarantee anonymity. Regarding data analysis, the R programming language was used on the pre-processed data. Besides quantitative data, quantitative data in form of open-ended feedback was collected.

3.7.5 Findings and Discussion

Within this section, the main findings will be reported and discussed. The course evaluation is a fundamental part of the quality management of many universities. On the one hand, the faculties receive feedback about courses and on the other hand lecturers get valuable insights into the students' perception of the course. The evaluation at Graz University of Technology is conducted in *Introduction to Programming* every year at the end of the semester anonymously using the universities course management system. Figure 3.26 shows all evaluation results beginning in the winter semester of 2016 to 2020 (5 years in total). One of the evaluation questions is *"How satisfied are you with the course in general?"*. The results show a noticeable improvement in course satisfaction for 2020. Whereas the evaluations mean was usually around neutral or dissatisfied in the last years, the level of satisfaction showed just a few negative evaluations and mainly positive feedback (M=4.66, Sd=1.46). This shows that the improvement is highly related to the shift from traditional teaching in the lecture halls to an interactive online experience (RQ1).

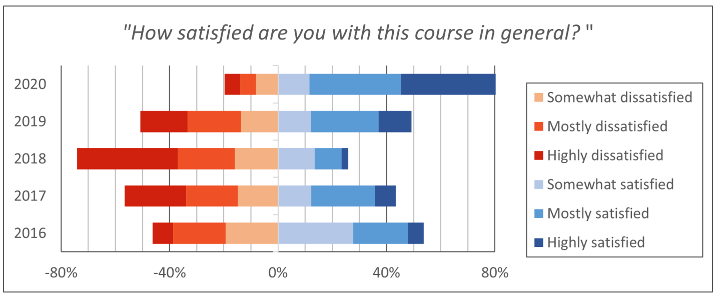

Figure 3.26: This plot contains all evaluation results from the five years between 2016 to 2020.

To get a better understanding of which course elements have an influence on satisfaction the positive and negative feedback from the open-ended question of the second evaluation has been taken into consideration. Therefore, all answers were analyzed and categorized into different categories. Table 3.22 contains the absolute number of mentions within the answers per group. Overall 148 (out of 188) students reported positive and 115 (out of 188) reported negative feedback. Approx. a third of all positive feedback was related to the synchronous streams on Twitch: *"Good preparation of the subject matter, information transfer worked great, maybe it was due to twitch but in my opinion the relationship between teachers and students was much more relaxed and pleasant than expected."* Another satisfying element is the asynchronous learning videos where students mentioned that the possibility to re-watch videos and pause them helped a lot. The students also mentioned that the videos helped them to keep orientation on the course. They also enjoyed the interactive elements in the videos to keep their attention. Another important aspect is interaction. Students mentioned that interaction during the Twitch streams (in the chat) and also asynchronous using the forum were valuable for them. On the negative side students mainly mentioned that the level of difficulty in the streams and assignments is too high. Since the students have different previous experiences on the topic, some were unchallenged while others were overwhelmed. Another aspect is explanations during lecture or exercise which were not clear for some students. Finally, some students respond that some streams had technical issues (such as sound problems at the beginning of the stream), which was irritating. Overall, this shows that students enjoy a mix of synchronous and asynchronous elements with a high level of interaction. Even though the level of interaction was already perceived as good, this could also help to tackle difficulties and difficulties in understanding explanations (RQ2).

Table 3.22: The table contains the categorized open-ended feedback from all students that participated the evaluation (n=188).

Positive	**#**	**Negative**	**#**
Streams (synchronous)	53	Difficulty	28
Learning Videos (asynchronous)	41	Explanations	16
Interaction	33	Technical Issues	12
Course Structure	15	Interaction	7
Slides	5	Workload	7

About 35% of the students watched all videos that were provided in the LMS. Figure 3.27 shows, that just a small group of the students did not watch any of the videos as preparation for the streams. The streams were highly visited over the whole semester since 44% stated that they have attended all streams. Table 3.21 shows that the majority of the students (M=5.17, Sd=0.96) appreciated the opportunity to practice and work independently (Item 5). It is also worth mentioning that the students attach importance to a clear structure and completeness of all learning materials (Items 9 and 10). However, the results also show that the personal contact with other students (M=4.12, Sd=1.52) seems need of improvement (Item 3). This shows that even with a lively interaction using the Twitch chat and the forum, face-to-face communication can not be replaced in an online setting (RQ3).

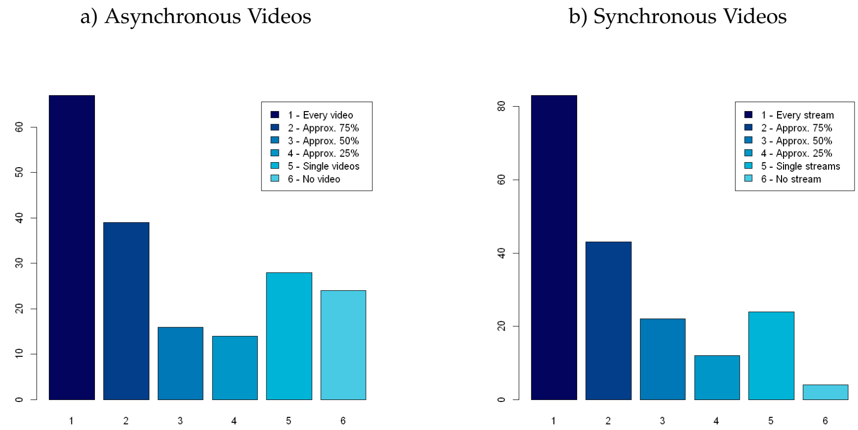

Figure 3.27: Both plots show how many of the participating students watched (a) the asynchronous videos and (b) the live streams.

3.7.6 Conclusion

The evaluation of the online programming course showed that the flipped classroom setting is beneficial for students in regard to satisfaction and learning performance. Especially beginners in programming can self-pace their learning progress. Following, we want to emphasize three main findings of the study.

Interactive Streams. The feedback from the evaluations showed that students appreciate a lively and interactive stream. This can be achieved by asking questions in the chat, motivating them for active participation using ARS, and promoting error-tolerant communication. A community-centered approach will bind the audience to the stream.

Availability of different learning resources. By providing different types of learning content such as preparation videos or slides, different forms of learning are enabled. Students respond that they enjoyed different explanations and perspectives on a specific topic.

Clear Structure. A clear structure helps students to stay on track and promotes self-paced learning. This can be reached by well-structured course information (f.i. using an LMS) and clearly defined rules.

Limitations. To encourage a high number of students to participate in the study bonus points were provided for all students that participate in both evaluations. Since some students already dropped out at the beginning of the semester valuable data is lost from these groups of students.

3.8 Cumulative Findings

Within this section, both research questions of this Ph.D. thesis will be answered based on the findings from the conducted studies.

> **Research Question 1**
>
> What design principles and strategies should be employed in developing learning technologies and educational scenarios to ensure an engaging learning experience in digital competencies and programming education?

Several learning technologies and educational scenarios have been designed, developed, and evaluated according to the *Engaging Learning Tree* model. Further, Figure 3.28 shows an abstraction of the model proposed in Figure 1.1, including five areas that influence engaging learning experiences: i) Teaching and Learning Concepts, ii) (Online) Learning Platforms, iii) Instructional Strategies, iv) Educational Assessment, and v) a solid foundation of the subject (computer science education).

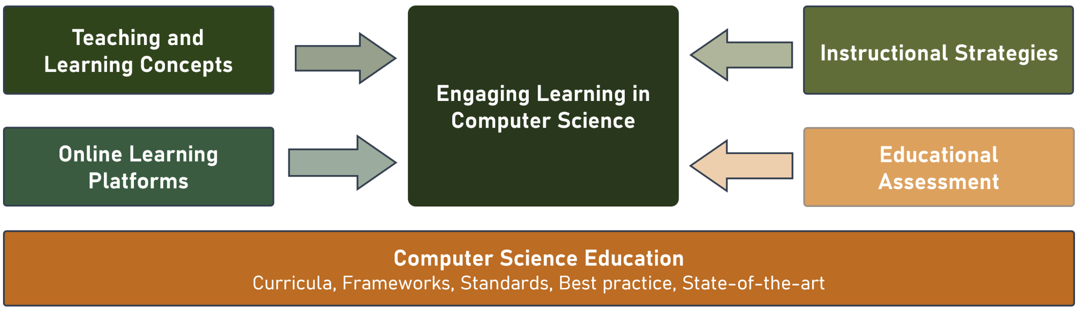

Figure 3.28: : This figure shows a simplified version of the FELCS model from Figure 1.1.

Within this dissertation, several **teaching and learning concepts** have been applied. A commonly used concept is Bloom's taxonomy (revised), which is also successfully implemented in computer science education. It provides teachers and educators with a framework for classifying educational objectives for effective teaching. In the SMAwT study (Article 5), for example, all activities are based on Anderson and Krathwohl's (2001) taxonomy to assign the required skills to different taxonomy levels. The desired learning outcomes are aligned with the corresponding taxonomy level to ensure the proper assessment strategy is used.

A meaningful learning experience and high academic achievement also require considering multiple perspectives on learning content. Only by applying the Dagstuhl Triangle in Article 5, the full potential of media education become apparent. Considering only one perspective (such as the user-oriented perspective) would result in incomplete findings of the learning process. This means certain aspects would be simplified or left out without these perspectives, impacting the learning outcomes.

Integrating learning technologies or educational scenarios into learning contexts is an important consideration in **instructional design**. A well-known model, for example, is the 5E model (Engage, Explore, Explain, Elaborate, Evaluate), which promotes student-centered active engagement within the learning process. Learning experiences can be placed in different educational phases according to their purpose. The studies in Article 3, Article 4, and Article 5 for example illustrate that the proposed interventions can be used when introducing a new topic, but also to make known concepts understandable. The learning technologies and scenarios have also been used within the assessment phase.

The findings from several studies (mainly Article 5 and Article 7) showed that learners are striving for a clear and transparent structure in course design. This is relevant in secondary but also in higher education. A structured design is highly prioritized, especially when using different elements of online and on-site teaching, but also synchronous and asynchronous elements. A clear structure helps learners focus on the content and not have to deal with organizational matters.

An important aspect of engagement is collaborative learning. The results from Article 3 show that the student's engagement is increased within a playful learning environment in a multiplayer context. However, the results also show that such platforms have to be used in a well-thought-out way and that there should be an appropriate educational concept behind them to maximize their benefits.

There is a large number of **online learning platforms** that can enrich the learning experience. These tools and platforms can be used partially (audience response systems, chats, forums, etc.) or entirely (learning management systems, streaming platforms, MOOCs, etc.). The studies from Article 5 and Article 7 show that the right balance between interaction and a mixture of multimedia content can help to engage students through a course. Aligning the right learning platform with instructional strategies and assessments is also important to increase student engagement. They need to be in line with the unique characteristics of large-scale teaching, for example, to decrease the attrition level (Article 6).

The learning process is highly affected by **educational assessment**. There are several ways to include assessments relating to the used platforms and instructional strategies. For example, audience response systems are tools that provide both teachers and students with feedback and assessment. Article 7 shows that these tools are helpful for students in terms of self-evaluation and immediate feedback. However, feedback instruments can also be integrated into game-based environments. Article 1 shows that gamification elements such as rewards for answering questions can positively impact student response rates. An important aspect is minimizing the distraction level of feedback and assessment due to context switches.

When it comes to assessment, educators and learners are concerned about security and privacy. Article 3 states that a trustworthy environment (such as a well-known platform) is the foundation for meaningful feedback. Another important aspect is that learning technologies should provide an error-tolerant environment for the students. This should encourage students to make their own experiences.

Research Question 2

How can the effectiveness of different types of engaging learning experiences and their characteristics be evaluated within the context of digital competencies and programming education?

Evaluating learning technologies and educational scenarios is central to teaching and learning. The *Engaging Learning Tree* model illustrates that the evaluation can consist of data- but also learner-centered evaluation methods. This means that the way how evaluations are conducted can be different. Data-centered approaches can be the analysis of interaction data or learning analytics. Learner-centered evaluation methods include user studies, surveys, feedback, and focus groups. Both data sources provide valuable insights into the learning process and educational characteristics.

An essential part of evaluations is to receive meaningful data. Learner-centered evaluation methods such as surveys require people to respond actively. Therefore, engaging approaches can help to motivate learners to attend them. The study presented in Article 1 proposes in-game questionnaires as an engaging and straightforward instrument. The benefits are that students get less distracted due to a context switch (which causes cognitive load).

A study with experts in Article 1 further investigates why students do not participate in surveys to improve the number of feedback. The most frequently mentioned reasons are time constraints, a lack of motivation, or previous negative experiences with feedback instruments. The study also shows that feedback and assessment are related to teachers' and students' security and privacy concerns. These concerns can be minimized by using a well-known and trustworthy environment.

When evaluating learning technologies and educational concepts, there are several methodologies. The study in Article 5 showed that clearly formulated learning objectives that are aligned to learning activities and assessment are an effective way to evaluate success (in terms of academic achievement). However, the limitations of several studies (Article 3, Article 4, and Article 5) suggest preferring a longitudinal study for making assumptions over the impact of a specific intervention instead.

Most of the conducted studies capture both perspectives students and teachers. This is important to consider both dimensions, teaching and learning. Findings of the studies conducted in Article 1, Article 3, and Article 5 highly emphasize on including both perspectives. Results from the SMAwT study even showed that just the alignment of the findings from the student with the teachers' evaluation provides an educational value.

Learning technologies and educational scenarios can be applied in different educational contexts and situations. The collaborative learning study (Article 3) with sCool was conducted in both a formal in-class scenario and an informal online scenario. Thus, the learning situation was the only variable that was different in the study design. The findings show huge differences between both interventions regarding performance, engagement, and collaborative learning.

The study of a learning analytics platform (Article 2) showed that a lightweight design is a crucial element for teachers to use a learning platform. These platforms should focus on displaying only the most relevant data using clear visualizations. Custom visualizations and data analysis add additional value for tracking and monitoring the students learning progress.

4 Conclusion and Future Work

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

Marie Curie

Engaging learning technologies and educational learning scenarios are influencing students' engagement and performance. The central question within this thesis was: *"How can computer science education be designed to be both engaging and enduring?"* Within this chapter, the central findings will be concluded, and implications will be drawn. In addition, this chapter will also cover the most considerable limitations. Finally, the chapter ends with an outlook and future work within this research field.

4.1 Conclusion

According to the *FELCS* model, engaging learning in computer science is influenced by the following factors: teaching and learning concepts, online learning platforms, instructional strategies, educational assessment, and the subject itself (computer science education). Within this thesis, several studies have been introduced where learning technologies and educational scenarios have been designed, developed, and evaluated. Each study followed the process stated in the *Engaging Learning Tree* model, aiming to increase students' engagement and learning performance.

Align engaging learning experiences, learning outcomes, and educational assessment. Aligning engaging learning experiences, learning outcomes, and assessment is vital for promoting engagement. Engaging learning experiences have an impact on students' interests and create an interactive and dynamic environment that fosters curiosity and exploration. These experiences should be designed with clear learning outcomes, ensuring students acquire the intended knowledge and skills. Furthermore, assessment strategies should align with learning experiences and outcomes to accurately measure students' progress and provide constructive feedback.

Combine learner- and data-centered evaluation methods. The conducted studies have impressively shown that there is often a discrepancy between a learning environment's intended and actual benefits. This underlines the high significance of meaningful evaluation methodologies and well-conceived evaluation instruments. In order to obtain an overall picture, it is therefore essential to include various aspects and stakeholders in the evaluation. Only by combining the experiences of students and teachers is it possible to assess the actual educational benefit of an intervention. Evaluations can be based on both learner-centered and data-centered approaches. Again, a more comprehensive evaluation can be reached if both sources are connected. The conducted studies also showed that one and the same intervention could be different in each setting. This can be due to learners' different backgrounds and experiences and to slightly modified learning contexts (online settings instead of on-site). For this reason, an evaluation should also consider various educational contexts.

Give orientation and structure. The findings of the studies showed that meaningful integration of engaging learning technologies and scenarios in an educational context is essential to ensure high acceptance and interest. These approaches are not the traditional teaching methods in both schools and universities. For this reason, it is crucial to provide appropriate orientation to learners. It is important to explain to the learners how these approaches are used and how they can use them. Only in this way can a meaningful learning experience be enabled.

4.2 Limitations

In this section, the limitations of this thesis are discussed.

This thesis investigates engagement from different angles. However, engagement has multiple dimensions and is influenced by several factors. There are different evaluation instruments to measure the students' levels of engagement. Each instrument is based on a specific definition of engagement. This work aimed not to design several studies that measure the same level of engagement; the goal was to consider engagement as a multi-dimensional concept. This means that engagement has to be considered from its specific context for each study.

The *FELCS* model explains all the factors that influence engaging learning in computer science. Within this thesis, only some concepts for each factor could be considered. Since there are countless models in teaching and learning concepts or a large number of instructional strategies, only a sample could be considered. The selected concepts were based on the fact that these are frequently used in literature and related studies. The model can be easily extended for future studies to include other concepts.

Finally, the starting point of this thesis was to create models for the entire field of computer science education. However, the field is broad and covers lots of different topics. Several computer science curricula, standards, or frameworks exist with different focuses. For this reason, the stated research questions and the model consider a subset of topics. These topics were digital competencies and programming education since these are a part of many computer science programs.

4.3 Future Work

As mentioned in the limitations, the scope of the proposed model for engaging learning is limited to digital competencies and programming education. For future work, the model can also be applied to other topics in the field of computer science, such as algorithms or networks. Due to the advent of artificial intelligence (AI) and large language models (LLMs) this opens the floor to further exciting questions for the field of computing education, especially how these technologies can be effectively integrated into an engaging learning environment.

The evaluation aspect in the *Engaging Learning Tree* used several evaluation methodologies for the studies. There are promising evaluation approaches and instruments related to different aspects of engaging learning. Future efforts could either be to adapt the existing model or to derive a model of its own in order to derive concrete evaluation methods from it. In this way, practitioners and researchers may have an opportunity to assess learning experiences using a model that is as established and validated.

List of Figures

List of Tables

Bibliography

Abdul Rahman, M. H., Ismail Yusuf Panessai, I., Mohd Noor, N. A. Z., & Mat Salleh, N. S. (2018). Gamification Elements and Their Impacts on Teaching and Learning – A Review. *The International journal of Multimedia & Its Applications*, *10*(06), 37–46. https://doi.org/10.5121/ijma.2018.10604 (cit. on p. 26)

Abel, J. P., Buff, C. L., Burr, S. A., et al. (2016). Social media and the fear of missing out: Scale development and assessment. *Journal of Business & Economics Research (JBER)*, *14*(1), 33–44 (cit. on p. 126).

Abramovich, G. (2019). *If you think email is dead, think again - adobe experience cloud*. Retrieved October 4, 2022, from https://business.adobe.com/blog/perspectives/if-you-think-email-is-dead-think-again. (Cit. on p. 108)

Abt, C. C. (1981). *Serious games*. The Viking Press. (Cit. on p. 74).

Adams, N. E. (2015). Bloom's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association : JMLA*, *103*(3), 152–153. https://doi.org/10.3163/1536-5050.103.3.010 (cit. on p. 17)

Adestra. (2019). *Teens and email: Getting in the inbox*. https://uplandsoftware.com/adestra/resources/blog/teens-email-getting-inbox/. (Cit. on p. 110)

Alammary, A. (2019). Blended learning models for introductory programming courses: A systematic review. *PLOS ONE*, *14*(9), 1–26. https://doi.org/10.1371/journal.pone.0221765 (cit. on p. 171)

Alario-Hoyos, C., Delgado-Kloos, C., Estévez-Ayres, I., Fernández Panadero, C., Blasco, J., Pastrana, S., Suarez-Tangil, G., & Villena, J. (2016). Interactive activities: The key to learning programming with moocs. *Proceedings of the European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCS 2016)* (cit. on p. 45).

Alario-Hoyos, C., Estévez-Ayres, I., Pérez-Sanagustín, M., Delgado Kloos, C., & Fernández-Panadero, C. (2017). Understanding learners' motivation and learning strategies in moocs. *The International Review of Research in Open and Distributed Learning*, *18*(3). https://doi.org/10.19173/irrodl.v18i3.2996 (cit. on p. 45)

Albluwi, I., & Salter, J. (2020). Using static analysis tools for analyzing student behavior in an introductory programming course. *Jordanian Journal of Computers and Information Technology (JJCIT)*, *6*, 215–233. https://doi.org/10.5455/jjcit.71-1584234700 (cit. on p. 74)

Albrecht, E., Gumz, F., & Grabowski, J. (2018). Experiences in introducing blended learning in an introductory programming course. *Proceedings of the 3rd European Conference of Software Engineering Education*, 93–101. https://doi.org/10.1145/3209087.3209101 (cit. on p. 171)

Alonso-Fernandez, C., Calvo-Morata, A., Freire, M., Martinez-Ortiz, I., & Fernández-Manjón, B. (2022). Game learning analytics:: Blending visual and data mining techniques to improve serious games and to better understand player learning. *Journal of Learning Analytics*, *9*. https://doi.org/10.18608/jla.2022.7633 (cit. on p. 39)

Alonso-Fernández, C., Calvo-Morata, A., Freire, M., Martínez-Ortiz, I., & Fernández-Manjón, B. (2022). Game learning analytics: Blending visual and data mining techniques to improve serious games and to better understand player learning. *Journal of Learning Analytics*, *9*(3), 32–49. https://doi.org/10.18608/jla.2022.7633 (cit. on p. 71)

Al-Samarraie, H., & Saeed, N. (2018). A systematic review of cloud computing tools for collaborative learning: Opportunities and challenges to the blended-learning environment [Publisher: Elsevier]. *Computers and Education*, *124*, 77–91. https://doi.org/10.1016/j.compedu.2018.05.016 (cit. on p. 2)

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). *A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives* (2nd ed.). Allyn & Bacon. (Cit. on pp. 16, 181).

Arnold, P., Kilian, L., Thillosen, A., & Zimmer, G. (2018). *Handbuch e-learning: Lehren und lernen mit digitalen medien*. https://doi.org/10.36198/9783838549651. (Cit. on p. 40)

Artino, A. R. (2012). Academic self-efficacy: From educational theory to instructional practice. *Perspectives on Medical Education*, *1*(2), 76–85. https://doi.org/10.1007/s40037-012-0012-5 (cit. on p. 20)

Astin, A. W. (1984). Student involvement: A developmental theory for higher education. *Journal of College Student Personnel*, *25*, 297–308 (cit. on p. 20).

Austing, R. H., Barnes, B. H., & Engel, G. L. (1977). A survey of the literature in computer science education since curriculum '68. *Commun. ACM*, *20*(1), 13–21. https://doi.org/10.1145/359367.359417 (cit. on p. 11)

Azmi, S., Iahad, N. A., & Ahmad, N. (2015). Gamification in online collaborative learning for programming courses: A literature review. *ARPN journal of engineering and applied sciences*, *10*, 18087–18094 (cit. on pp. 89, 90).

Bandura, A. (1978). Self-efficacy: Toward a unifying theory of behavioral change [Perceived Self-Efficacy: Analyses of Bandura's Theory of Behavioural Change]. *Advances in Behaviour Research and Therapy*, *1*(4), 139–161. https://doi.org/https://doi.org/10.1016/0146-6402(78)90002-4 (cit. on p. 20)

Bandura, A., Pervin, L., & John, O. (1999). Handbook of personality. *New York, NY Guilford* (cit. on p. 89).

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905 (cit. on p. 74)

Baturay, M. H. (2015). An overview of the world of moocs [International Conference on New Horizons in Education, INTE 2014, 25-27 June 2014, Paris, France]. *Procedia - Social and Behavioral Sciences*, *174*, 427–433. https://doi.org/https://doi.org/10.1016/j.sbspro.2015.01.685 (cit. on p. 33)

Baumgartner, I. (2014). Einfluss von fachinteresse auf situationales interesse bei der bearbeitung von aufgaben im fach geschichte. eine treatment-studie mit variation der aufgabeninteressantheit am beispiel "deutscher widerstand im nationalsozialismus". https://doi.org/10.15475/paradigma.2014.1.9. (Cit. on p. 139)

Bell, T., & Vahrenhold, J. (2018). Cs unplugged—how is it used, and does it work? In H.-J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes: Essays dedicated to juraj hromkovič on the occasion of his 60th birthday* (pp. 497–521). Springer International Publishing. https://doi.org/10.1007/978-3-319-98355-4_29. (Cit. on p. 31)

Bell, T., & Witten, I. H. (1998). Computer Science Unplugged . . . off-line activities and games for all ages. https://classic.csunplugged.org/documents/books/english/unplugged-book-v1.pdf (cit. on p. 30)

Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., & Berta, R. (2013). Assessment in and of serious games: An overview. *Adv. in Hum.-Comp. Int.*, *2013*. https://doi.org/10.1155/2013/136864 (cit. on pp. 38, 53, 57)

Ben-Eliyahu, A., Moore, D., Dorph, R., & Schunn, C. D. (2018). Investigating the multidimensionality of engagement: Affective, behavioral, and cognitive engagement across science activities and contexts. *Contemporary Educational Psychology*, *53*, 87–105. https://doi.org/https://doi.org/10.1016/j.cedpsych.2018.01.002 (cit. on p. 21)

Bergmann, J., & Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International Society for Technology in Education. https://books.google.at/books?id=nBi2pwAACAAJ. (Cit. on p. 170)

Biggs, J. (2014). Constructive alignment in university teaching. *HERDSA Review of Higher Education*, *1*, 5–22 (cit. on p. 20).

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. McGraw-Hill Education. https://books.google.at/books?id=VC1FBgAAQBAJ. (Cit. on p. 19)

Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning : The solo taxonomy (structure of the observed learning outcome) / john b. biggs, kevin f. collis*. Academic Press New York. (Cit. on p. 18).

Black, P., Harrison, C., Lee, C., Marshall, B., & William, D. (2003). Formative and summative assessment: Can they serve learning together. *AERA Chicago*, *23* (cit. on pp. 40, 41).

Bloom, B. S., Engelhart, M. B., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives. the classification of educational goals. handbook 1: Cognitive domain*. Longmans Green. (Cit. on p. 16).

Bonk, C. J., Graham, C. R., Cross, J., & Moore, M. G. (2006). *The handbook of blended learning: Global perspectives, local designs*. Wiley. (Cit. on p. 170).

Brabrand, C., & Dahl, B. (2009). Using the SOLO taxonomy to analyze competence progression of university science curricula. *Higher Education*, *58*(4), 531–549. https://doi.org/10.1007/s10734-009-9210-4 (cit. on p. 19)

Breimer, E., Fryling, M., & Yoder, R. (2016). Full flip, half flip and no flip: Evaluation of flipping an introductory programming course. *Information Systems Education Journal*, *14*, 4–16 (cit. on p. 171).

Brinda, T., Brüggen, N., Diethelm, I., Knaus, T., Kommer, S., Kopf, C., Missomelius, P., Leschke, R., Tilemann, F., & Weich, A. (2019). Frankfurt-dreieck zur bildung in der digital vernetzten welt. *Informatik für alle* (cit. on p. 128).

Brinda, T., Diethelm, I., Gemulla, R., Romeike, R., Schöning, J., & Schulte, C. (2016). Dagstuhl-erklärung: Bildung in der digitalen vernetzten welt. https://doi.org/10.13140/RG.2.1.3957.2245. (Cit. on pp. 123, 126, 128, 139)

Brindley, J. E., Blaschke, L. M., & Walti, C. (2009). Creating effective collaborative learning groups in an online environment. *International Review of Research in Open and Distributed Learning*, *10*(3). https://doi.org/https://doi.org/10.19173/irrodl.v10i3.675 (cit. on p. 27)

Brooke, J. (1995). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, *189* (cit. on pp. 79, 80).

Cano, A. E., Fernandez, M., & Alani, H. (2014). Detecting child grooming behaviour patterns on social media. *Social Informatics: 6th International Conference, SocInfo 2014, Barcelona, Spain, November 11-13, 2014. Proceedings 6*, 412–427 (cit. on p. 126).

Carr, P. B., & Walton, G. M. (2014). Cues of working together fuel intrinsic motivation. *Journal of Experimental Social Psychology*, *53*, 169–184. https://doi.org/https://doi.org/10.1016/j.jesp.2014.03.015 (cit. on p. 88)

Caspari-Sadeghi, S. (2022). Applying Learning Analytics in Online Environments: Measuring Learners' Engagement Unobtrusively. *Frontiers in Education*, *7*, 840947. https://doi.org/10.3389/feduc.2022.840947 (cit. on p. 39)

Chakraverty, S., & Chakraborty, P. (2020). Tools and techniques for teaching computer programming: A review. *Journal of Educational Technology Systems*, *49*(2), 170–198 (cit. on p. 157).

Chaudron, S. (2015). *Young children (0-8) and digital technology: A qualitative exploratory study across seven countries* (Other LB-NA-27052-EN-N). Luxembourg (Luxembourg), Publications Office of the European Union. https://doi.org/10.2788/00749. (Cit. on p. 109)

Chaudy, Y., Connolly, T., & Hainey, T. (2014). Learning analytics in serious games: A review of the literature. http://www.ed2owork.eu/conference (cit. on pp. 74, 75, 82)

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357 (cit. on p. 162).

Chen, C., Sonnert, G., Sadler, P. M., Sasselov, D. D., Fredericks, C., & Malan, D. J. (2020a). Going over the cliff: Mooc dropout behavior at chapter transition. *Distance Education*, *41*(1), 6–25 (cit. on p. 158).

Chen, C., Sonnert, G., Sadler, P. M., Sasselov, D. D., Fredericks, C., & Malan, D. J. (2020b). Going over the cliff: Mooc dropout behavior at chapter transition. *Distance Education*, *41*(1), 6–25 (cit. on pp. 165, 166).

Chen, K.-C., & Jang, S.-J. (2010). Motivation in online learning: Testing a model of self-determination theory [Emerging and Scripted Roles in Computer-supported Collaborative Learning]. *Computers in Human Behavior*, *26*(4), 741–752. https://doi.org/https://doi.org/10.1016/j.chb.2010.01.011 (cit. on p. 23)

Chen, X., Chen, S., Wang, X., & Huang, Y. (2021). "i was afraid, but now i enjoy being a streamer!": Understanding the challenges and prospects of using live streaming for online education. *Proc. ACM Hum.-Comput. Interact.*, *4*(CSCW3). https://doi.org/10.1145/3432936 (cit. on p. 36)

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology*, *6*. https://doi.org/10.3389/fpsyg.2015.00049 (cit. on p. 12)

Chevalier, M., Giang, C., El-Hamamsy, L., Bonnet, E., Papaspyros, V., Pellet, J.-P., Audrin, C., Romero, M., Baumberger, B., & Mondada, F. (2022). The role of feedback and guidance as intervention methods to foster computational thinking in educational robotics learning activities for primary school. *Computers & Education*, *180*, 104431. https://doi.org/https://doi.org/10.1016/j.compedu.2022.104431 (cit. on p. 41)

Clark, R., Besterfield-Sacre, M., Budny, D., Bursic, K., Clark, W., Norman, B., Parker, R., Patzer, J., & Slaughter, W. (2016). Flipping engineering courses: A school wide initiative. *Advances in engineering education*, *5* (cit. on p. 171).

Combéfis, S., Beresnevičius, G., & Dagiene, V. (2016a). Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, *10*, 39–60. https://doi.org/10.15388/ioi.2016.03 (cit. on pp. 43, 44)

Combéfis, S., Beresnevičius, G., & Dagiene, V. (2016b). Learning programming through games and contests: Overview, characterisation and discussion. *10*, 39–60. https://doi.org/10.15388/ioi.2016.03 (cit. on p. 75)

Commission, E., Centre, J. R., Carretero, S., Vuorikari, R., & Punie, Y. (2018). *Digcomp 2.1 : The digital competence framework for citizens with eight proficiency levels and examples of use*. Publications Office. https://doi.org/doi/10.2760/836968. (Cit. on p. 110)

Computer Science Teachers Association. (2017). *Csta k-12 computer science standards, revised 2017*. Retrieved January 26, 2023, from http://www.csteachers.org/standards. (Cit. on p. 14)

Cook, D., & Artino, A. (2016). Motivation to learn: An overview of contemporary theories. *Medical Education*, *50*, 997–1014. https://doi.org/10.1111/medu.13074 (cit. on pp. 22, 24)

Creighton, S., & Szymkowiak, A. (2014). The effects of cooperative and competitive games on classroom interaction frequencies [2nd World Conference on Psychology and Sociology, PSYSOC 2013, 27-29 November 2013, Brussels, Belgium]. *Procedia - Social and Behavioral Sciences*, *140*, 155–163. https://doi.org/https://doi.org/10.1016/j.sbspro.2014.04.402 (cit. on p. 88)

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. Harper & Row. https://www.amazon.com/Flow-Psychology-Experience-Mihaly-Csikszentmihalyi/dp/0060162538?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&

tag = chimborio5 - 20 & linkCode = xm2 & camp = 2025 & creative = 165953 & creativeASIN=0060162538. (Cit. on p. 53)

Cvetković, D. (2016). *Virtual learning*. IntechOpen. https://books.google.at/books?id=HfiODwAAQBAJ. (Cit. on p. 33)

The cyber security body of knowledge. (2020). University of Bristol. https://www.cybok.org. (Cit. on p. 110)

Dagiene, V., Hromkovic, J., & Lacher, R. (2020). A two-dimensional classification model for the bebras tasks on informatics based simultaneously on subfields and competencies. In K. Kori & M. Laanpere (Eds.), *Informatics in schools. engaging learners in computational thinking* (pp. 42–54). Springer International Publishing. (Cit. on p. 17).

Dalipi, F., Imran, A. S., & Kastrati, Z. (2018). Mooc dropout prediction using machine learning techniques: Review and research challenges. *2018 IEEE global engineering education conference (EDUCON)*, 1007–1014 (cit. on p. 157).

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319–340 (cit. on p. 83).

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior* [Originally published by Plenum Press, New York 1985]. Springer Science+Business Media. (Cit. on p. 21).

Dede, C. (2010). Introduction to virtual reality in education. *Themes in Science and Technology Education*, *2*(1-2), 7–9 (cit. on p. 150).

Deng, R., Benckendorff, P., & Gannaway, D. (2020). Learner engagement in moocs: Scale development and validation. *British Journal of Educational Technology*, *51*(1), 245–262. https://doi.org/https://doi.org/10.1111/bjet.12810 (cit. on pp. 21, 152)

Dengel, A., & Gehrlein, R. (2022). Comparing teachers' and preservice teachers' opinions on teaching methods in computer science education. *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/3556787.3556866 (cit. on p. 23)

Department of Education, U. S. (2010a). National education technology plan. http://www.ed.gov/technology/netp-2010 (cit. on p. 75)

Department of Education, U. S. (2017). Reimagining the role of technology in education: National education technology plan. https://tech.ed.gov/netp/. (accessed: 21.01.2021) (cit. on p. 75)

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, *MindTrek 2011*, *11*, 9–15. https://doi.org/10.1145/2181037.2181040 (cit. on pp. 25, 26, 57)

Diethelm, I. (2022). Digital education and informatics – you can't have one without the other. *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/3556787.3556790 (cit. on p. 139)

DiFranzo, D., Choi, Y. H., Purington, A., Taft, J. G., Whitlock, J., & Bazarova, N. N. (2019). Social media testdrive: Real-world social media education for the next generation. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–11. https://doi.org/10.1145/3290605.3300533 (cit. on p. 129)

DiFranzo, D., Choi, Y. H., Purington, A., Taft, J. G., Whitlock, J., & Bazarova, N. N. (2023). Social media testdrive. https://socialmediatestdrive.org/. (Cit. on pp. 126, 129)

Djaouti, D., Alvarez, J., Jessel, J.-P., & Rampnoux, O. (2011). Origins of serious games. *Serious Games and Edutainment Applications*, 25–43 (cit. on p. 74).

Dorn, J. (2019). Instahub datenbanken und datenschutz mit einem extra für den unterricht entwickelten sozialen netzwerk unterrichten. In A. Pasternak (Ed.), *Informatik für alle* (p. 375). Gesellschaft für Informatik. https://doi.org/10.18420/infos2019-f3. (Cit. on pp. 126, 129)

Draffan, E. A., & Rainger, P. (2006). A model for the identification of challenges to blended learning. *ALT-J*, *14*(1), 55–67. https://doi.org/10.1080/09687760500479787 (cit. on p. 30)

Driver, M. (2002). Exploring student perceptions of group interaction and class satisfaction in the web-enhanced classroom. *The Internet and Higher Education*, *5*, 35–45. https://doi.org/10.1016/S1096-7516(01)00076-8 (cit. on p. 97)

Du, X., Yang, J., Shelton, B. E., Hung, J.-L., & Zhang, M. (2021). A systematic meta-review and analysis of learning analytics research. *Behaviour & Information Technology*, *40*(1), 49–62. https://doi.org/10.1080/0144929X.2019.1669712 (cit. on p. 39)

Dunleavy, M., Dede, C., & Mitchell, R. (2009). Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. *Journal of Science Education and Technology*, *18*(1), 7–22. https://doi.org/10.1007/s10956-008-9119-1 (cit. on p. 2)

Elmaleh, J., & Shankararaman, V. (2017). Improving student learning in an introductory programming course using flipped classroom and competency framework. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 49–55. https://doi.org/10.1109/EDUCON.2017.7942823 (cit. on p. 45)

A european strategy for a better internet for our children. (2012). European Commission. https://digital-strategy.ec.europa.eu/en/policies/european-strategy-better-internet-children. (Cit. on p. 110)

Feierabend, S., Rathgeb, T., Kheredmand, H., & Glöckler, S. (2022). *Jim-studie 2022. jugend, information, medien* (tech. rep.). Medienpädagogischer Forschungsverbund Südwest. https://www.mpfs.de/fileadmin/files/Studien/JIM/2022/JIM_2022_Web_final.pdf. (Cit. on p. 125)

Feklistova, L., Lepp, M., & Luik, P. (2021). Learners' performance in a mooc on programming. *Education Sciences*, *11*(9), 521 (cit. on p. 157).

Fidalgo-Blanco, Á., Sein-Echaluce, M. L., & García-Peñalvo, F. J. (2016). From massive access to cooperation: Lessons learned and proven results of a hybrid xmooc/cmooc pedagogical approach to moocs. *International Journal of Educational Technology in Higher Education*, *13*(1), 24. https://doi.org/10.1186/s41239-016-0024-z (cit. on p. 33)

Fielding, J. A. (2019). Rethinking CRAAP: Getting students thinking like fact-checkers in evaluating web sources. *College & Research Libraries News*, *80*(11), 620. https://doi.org/10.5860/crln.80.11.620 (cit. on p. 133)

Freire, M., Serrano-Laguna, Á., Manero, B., Martinez-Ortiz, I., Ger, P. M., & Baltasar, F.-M. (2016). Game learning analytics: Learning analytics for serious games. https://doi.org/10.1007/978-3-319-17727-4_21-1. (Cit. on pp. 77, 84)

Frommel, J., Rogers, K., Brich, J., Besserer, D., Bradatsch, L., Ortinau, I., Schabenberger, R., Riemer, V., Schrader, C., & Weber, M. (2015). Integrated questionnaires: Maintaining presence in game environments for self-reported data acquisition. *Association for Computing Machinery*, 359–368 (cit. on pp. 57, 70).

Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., & Suehiro, D. (2017). Real-time learning analytics for c programming language courses. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 280–288. https://doi.org/10.1145/3027385.3027407 (cit. on p. 12)

Gafoor, K. (2013). Types and phases of evaluation in educational practice. https://doi.org/10.13140/2.1.3801.1680. (Cit. on p. 39)

Gan, B., Menkhoff, T., & Smith, R. (2015). Enhancing students' learning process through interactive digital media: New opportunities for collaborative learning [Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era]. *Computers in Human Behavior*, *51*, 652–663. https://doi.org/https://doi.org/10.1016/j.chb.2014.12.048 (cit. on p. 28)

Garrison, D., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, *7*(2), 95–105.

https://doi.org/https://doi.org/10.1016/j.iheduc.2004.02.001 (cit. on pp. 28, 30)

Geißler, N. (2008). *Konzeption und evaluation eines unterrichtsprogramms zur prävention und verminderung von antisemitismus* (Doctoral dissertation). Justus-Liebig-Universität. Otto-Behaghel-Str. 8, 35394 Gießen. http://geb.uni-giessen.de/geb/volltexte/2008/6642. (Cit. on p. 139)

Göbl, B., Hristova, D., Jovicic, S., Chevron, M.-F., Slunecko, T., & Hlavacs, H. (2019). Fostering social media literacy through a participatory mixed-methods approach: Discussion of workshop findings. *2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH)*, 1–8. https://doi.org/10.1109/SeGAH.2019.8882464 (cit. on p. 129)

Google. (2023). Be internet awesome. https://beinternetawesome.withgoogle.com/en_us/. (Cit. on pp. 126, 129)

Grillenberger, A., & Romeike, R. (2014). Big data–challenges for computer science education. *Informatics in Schools. Teaching and Learning Perspectives: 7th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2014, Istanbul, Turkey, September 22-25, 2014. Proceedings 7*, 29–40 (cit. on p. 126).

Guo, M., & Ottenbreit-Leftwich, A. (2020). Exploring the k-12 computer science curriculum standards in the us. *Proceedings of the 15th Workshop on Primary and Secondary Computing Education*, 1–6 (cit. on p. 126).

Gupta, G. K. (2007). Computer science curriculum developments in the 1960s. *IEEE Annals of the History of Computing*, *29*(2), 40–54. https://doi.org/10.1109/MAHC.2007.20 (cit. on p. 11)

Gütl, C., Chang, V., & Hernandez Rizzardini, R. (2014). Must we be concerned with the massive drop-outs in mooc? an attrition analysis of open courses [International Conference on Interactive Collaborative Learning ; Conference date: 03-12-2014 Through 06-12-2014]. *International Conference on Interactive Collaborative Learning*, 1–8 (cit. on p. 156).

Hailikari, T., Virtanen, V., Vesalainen, M., & Postareff, L. (2022). Student perspectives on how different elements of constructive alignment support active learning. *Active Learning in Higher Education*, *23*(3), 217–231. https://doi.org/10.1177/1469787421989160 (cit. on p. 20)

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., Tatham, R. L., et al. (2006). Multivariate data analysis (vol. 6) (cit. on p. 137).

Halverson, L. R., & Graham, C. R. (2019). Learner Engagement in Blended Learning Environments: A Conceptual Framework. *Online Learning*, *23*(2). https://doi.org/10.24059/olj.v23i2.1481 (cit. on p. 21)

Hamm, M. P., Newton, A. S., Chisholm, A., Shulhan, J., Milne, A., Sundar, P., Ennis, H., Scott, S. D., & Hartling, L. (2015). Prevalence and effect of cyberbullying on children and young people: A scoping review of social media studies. *JAMA pediatrics*, *169*(8), 770–777 (cit. on p. 126).

Hanz, K., & Kingsland, E. S. (2020). Fake or for real? A fake news workshop. *Reference Services Review*, *48*(1), 91–112. https://doi.org/10.1108/RSR-09-2019-0064 (cit. on p. 133)

Harms, J., Biegler, S., Wimmer, C., Kappel, K., & Grechenig, T. (2015). Gamification of online surveys: Design process, case study, and evaluation. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, & M. Winckler (Eds.), *Human-computer interaction – interact 2015* (pp. 219–236). Springer International Publishing. (Cit. on pp. 58, 70).

Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*, 904–908 (cit. on pp. 56, 62).

Hashim, H. (2018). Application of Technology in the Digital Era Education. *International Journal of Research in Counseling and Education*, *1*(2), 1. https://doi.org/10.24036/002za0002 (cit. on p. 2)

Hauge, J. B., Berta, R., Fiucci, G., Manjón, B. F., Padrón-Nápoles, C., Westra, W., & Nadolski, R. (2014). Implications of learning analytics for serious game design. *2014 IEEE 14th International Conference on Advanced Learning Technologies*, 230–232. https://doi.org/10.1109/ICALT.2014.73 (cit. on pp. 74, 82, 84)

Hayat, A. A., Shateri, K., Amini, M., & Shokrpour, N. (2020). Relationships between academic self-efficacy, learning-related emotions, and metacognitive learning strategies with academic performance in medical students: A structural equation model. *BMC Medical Education*, *20*(1), 76. https://doi.org/10.1186/s12909-020-01995-9 (cit. on p. 20)

Heller, N., & Mader, S. (2021). Effects of Competitive Coding Games on Novice Programmers Effects of Competitive Coding Games on Novice Programmers. *ProceedingsProceedings of the 23rd International Conference on Interactive Collaborative Learning (ICL2020)*, (March). https://doi.org/10.1007/978-3-030-68198-2 (cit. on p. 90)

Henderikx, M. A., Kreijns, K., & Kalz, M. (2017). Refining success and dropout in massive open online courses based on the intention–behavior gap. *Distance Education*, *38*(3), 353–368 (cit. on p. 156).

Hernandez Rizzardini, R., Gütl, C., Chang, V., & Morales, M. (2014). Mooc in latin america: Implementation and lessons learned. *International Workshop on Learning Technology for Education in Cloud*, 147–158 (cit. on pp. 155, 156).

Hinkle, D. E., Wiersma, W., & Jurs, S. G. (2003). *Applied statistics for the behavioral sciences* (Vol. 663). Houghton Mifflin college division. (Cit. on p. 145).

Höfler, E., Zimmermann, C., & Ebner, M. (2017). A case study on narrative structures in instructional MOOC designs. *Journal of Research in Innovative Teaching & Learning*, *10*(1), 48–62 (cit. on p. 165).

Hörmann, C., Schmidthaler, E., Kuka, L., Rottenhofer, M., & Sabitzer, B. (2022). From non-existent to mandatory in five years–the journey of digital education in the austrian school system (cit. on pp. 126, 149).

Horst, R., Naraghi-Taghi-Off, R., Diez, S., Uhmann, T., Müller, A., & Dörner, R. (2019). Funplogs – a serious puzzle mini-game for learning fundamental programming principles using visual scripting. https://doi.org/10.1007/978-3-030-33720-9_38. (Cit. on p. 90)

Hsu, Y.-C., Irie, N. R., & Ching, Y.-H. (2019). Computational thinking educational policy initiatives (ctepi) across the globe. *TechTrends*, *63*(3), 260–270. https://doi.org/10.1007/s11528-019-00384-4 (cit. on p. 13)

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., Pal, Y., Jackova, J., & Jasute, E. (2015). A global snapshot of computer science education in k-12 schools. *Proceedings of the 2015 ITiCSE on Working Group Reports*, 65–83. https://doi.org/10.1145/2858796.2858799 (cit. on p. 15)

Hunter, R. Mooc pedagogy and learning theories [HKCPD Hub Virtual International Conference 2021 ; Conference date: 08-01-2021 Through 10-01-2021]. English. In: HKCPD Hub Virtual International Conference 2021 ; Conference date: 08-01-2021 Through 10-01-2021. 2021, January (cit. on p. 157).

Hutchison-Green, M. A., Follman, D. K., & Bodner, G. M. (2008). Providing a Voice: Qualitative Investigation of the Impact of a First-Year Engineering Experience on Students' Efficacy Beliefs. *Journal of Engineering Education*, *97*(2), 177–190. https://doi.org/10.1002/j.2168-9830.2008.tb00966.x (cit. on p. 20)

Ibáñez, M.-B., Di-Serio, Á., & Delgado-Kloos, C. (2014). Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, *7*(3), 291–301. https://doi.org/10.1109/TLT.2014.2329293 (cit. on p. 42)

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C., & Toll, D. (2015). Educational data mining and learning analytics in programming: Literature review and case studies, 41–63. https://doi.org/10.1145/2858796.2858798 (cit. on pp. 75, 76)

Indrawan, E., & Jalinus, S. N. (2019). Review project based learning. *International Journal of Science and Research (IJSR)*, *8*(4) (cit. on p. 28).

Interland - be internet awesome. (2017). Google. https://beinternetawesome.withgoogle.com/en_us/. (Cit. on p. 111)

Jackson, C. K., & Bruegmann, E. (2009). Teaching students and teaching each other: The importance of peer learning for teachers. *American Economic Journal: Applied Economics*, *1*(4), 85–108. https://doi.org/10.1257/app.1.4.85 (cit. on p. 27)

Javidi, G., & Sheybani, E. (2018). K-12 cybersecurity education, research, and outreach. *2018 IEEE Frontiers in Education Conference (FIE)*, 1–5 (cit. on p. 126).

Jena, R. (2015). Technostress in ict enabled collaborative learning environment: An empirical study among indian academician [Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era]. *Computers in Human Behavior*, *51*, 1116–1123. https://doi.org/https://doi.org/10.1016/j.chb.2015.03.020 (cit. on p. 26)

Johnson, C. G., & Fuller, U. (2006). Is bloom's taxonomy appropriate for computer science? *Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling 2006*, 120–123. https://doi.org/10.1145/1315803.1315825 (cit. on p. 18)

Johnson, J. (2021a). *Number of sent and received e-mails per day worldwide from 2017 to 2025*. https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/. (Cit. on p. 107)

Johnson, J. (2021b). Digital 2021: Global digital overview. https://www.statista.com/statistics/617136/digital-population-worldwide/. (Cit. on p. 109)

Johnson, R. T., Johnson, D. W., & Stanne, M. B. (1986). Comparison of computer-assisted cooperative, competitive, and individualistic learning. *American Educational Research Journal*, *23*(3), 382–392. https://doi.org/10.3102/00028312023003382 (cit. on p. 88)

Joshi, D. R., Adhikari, K. P., Khanal, B., Khadka, J., & Belbase, S. (2022). Behavioral, cognitive, emotional and social engagement in mathematics learning during covid-19 pandemic. *PLOS ONE*, *17*(11), 1–22. https://doi.org/10.1371/journal.pone.0278052 (cit. on p. 21)

K-12 Computer Science Framework Steering Committee. (2016). *K-12 computer science framework* (tech. rep.). New York, NY, USA, Association for Computing Machinery. (Cit. on p. 13).

Kaiser, H. F. (1974). An index of factorial simplicity. *Psychometrika*, *39*(1), 31–36. https://doi.org/10.1007/BF02291575 (cit. on p. 137)

Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. *Education and Information Technologies*, *23*. https://doi.org/10.1007/s10639-018-9736-0 (cit. on p. 74)

Kannadhasan, S., Shanmuganantham, M., Nagarajan, R., & Deepa, S. (2020). The role of future e-learning system and higher education. *International Journal of Advanced Research in Science, Communication and Technology*, *12*(2), 261–266 (cit. on p. 154).

Kaspersky. (2017). *Internet safety for kids: How to protect your child from the top 7 dangers they face online*. Retrieved October 4, 2022, from https://usa.kaspersky.com/resource-center/threats/top-seven-dangers-children-face-online. (Cit. on p. 108)

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming [Cyprus International Conference on Educational Research (CY-ICER-2012)North Cyprus, US08-10 February, 2012]. *Procedia - Social and Behavioral Sciences*, *47*, 1991–1999. https://doi.org/https://doi.org/10.1016/j.sbspro.2012.06.938 (cit. on p. 44)

Ke, F., Xie, K., & Xie, Y. (2015). Game-based learning engagement: A theory- and data-driven exploration. *British Journal of Educational Technology*, *47*. https://doi.org/10.1111/bjet.12314 (cit. on p. 24)

Kickmeier-Rust, M. D. (2009). Talking digital educational games. *Proceedings of the 1st international open workshop on intelligent personalization and adaptation in digital educational games*, 55–66 (cit. on p. 37).

Knutas, A. (2016). *Increasing Beneficial Interactions in a Computer-Supported Collaborative Environment*. Acta Universitatis Lappeenrantaensis. (Cit. on p. 90).

Kojic, A., Kojic, M., Pirker, J., Gütl, C., Mentzelopoulos, M., & Economou, D. (2018). Scool - a mobile flexible learning environment. *iLRN 2018 Montana*, 72–84 (cit. on pp. 7, 58, 76).

Kovah, J. (2014). Final thoughts: Leadership in the "classroom". *37*(1), 39–40 (cit. on p. 170).

Kozaris, I. A. (2010). Platforms for e-learning. *Analytical and Bioanalytical Chemistry*, *397*(3), 893–898. https://doi.org/10.1007/s00216-010-3587-x (cit. on p. 31)

Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory into practice*, *41*(4), 212–218 (cit. on p. 130).

Krugel, J., & Hubwieser, P. (2017). Computational thinking as springboard for learning object-oriented programming in an interactive mooc. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 1709–1712 (cit. on p. 158).

Krugel, J., & Hubwieser, P. (2018). Strictly objects first: A multipurpose course on computational thinking. In M. S. Khine (Ed.), *Computational thinking in the stem disciplines: Foundations and research highlights* (pp. 73–98). Springer International Publishing. (Cit. on p. 159).

Krugel, J., & Hubwieser, P. (2020). Web-based learning in computer science: Insights into progress and problems of learners in moocs. (Cit. on pp. 158, 166).

Kuh, G., Kinzie, J., Buckley, J., Ed, B., & Hayek, J. (2007). Piecing together the student success puzzle: Research, propositions, and recommendations. ashe higher education report, volume 32, number 5. *ASHE Higher Education Report*, *32* (cit. on p. 21).

Kumaraguru, P. (2009). Phishguru: A system for educating users about semantic attacks (cit. on p. 111).

Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M., & Pham, T. (2009). School of phish: A real-world evaluation of anti-phishing training. *SOUPS 2009 - Proceedings of the 5th Symposium On Usable Privacy and Security*. https://doi.org/10.1145/1572532.1572536 (cit. on p. 111)

Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., & Hong, J. (2010). Teaching johnny not to fall for phish. *ACM Trans. Internet Technol.*, *10*(2). https://doi.org/10.1145/1754393.1754396 (cit. on pp. 111, 121)

Laamarti, F., Eid, M., & Saddik, A. E. (2014). An overview of serious games. *Int. J. Comput. Games Technol.*, *2014*. https://doi.org/10.1155/2014/358152 (cit. on p. 37)

Lab, K. (2016). Growing up online what kids conceal. https://www.kaspersky.com/about/press-releases/2016_one-in-two-children-hide-risky-online-behavior-from-parents--kaspersky-lab-research. (Cit. on p. 109)

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *SIGCSE Bull.*, *37*(3), 14–18. https://doi.org/10.1145/1151954.1067453 (cit. on p. 74)

Lamb, A., & Johnson, L. (2011). Scratch: Computer programming for 21st century learners. *Teacher Librarian*, *38*(4), 64–68 (cit. on p. 12).

Lamprou, A., Repenning, A., & Escherle, N. A. (2017). The solothurn project: Bringing computer science education to primary schools in switzerland. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 218–223. https://doi.org/10.1145/3059009.3059017 (cit. on pp. 126, 149)

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. Retrieved March 23, 2023, from http://www.jstor.org/stable/2529310 (cit. on p. 142)

Larkin, H., & Richardson, B. (2012). Creating high challenge/high support academic environments through constructive alignment: Student outcomes. *Teaching in Higher Education - TEACH HIGH EDUC*, *18*, 1–13. https://doi.org/10.1080/13562517.2012.696541 (cit. on p. 20)

Larmer, J., & Mergendoller, J. (2023). *Gold standard pbl: Essential project design elements*. Retrieved June 19, 2023, from https://www.pblworks.org/blog/gold-standard-pbl-essential-project-design-elements. (Cit. on p. 28)

Lastdrager, E., Gallardo, I. C., Hartel, P., & Junger, M. (2017). How effective is anti-phishing training for children? *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*, 229–239 (cit. on p. 109).

Leff, A., & Rayfield, J. T. (2001). Web-application development using the model/view/controller design pattern. *Proceedings fifth ieee international enterprise distributed object computing conference*, 118–127 (cit. on p. 78).

Lewis, J. R., & Sauro, J. (2018). Item benchmarks for the system usability scale. *J. Usability Studies*, *13*(3), 158–167 (cit. on pp. 81, 83).

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, *49*(4), 764–766. https://doi.org/https://doi.org/10.1016/j.jesp.2013.03.013 (cit. on p. 81)

Lindberg, R. S. N., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in k-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, *50*(4), 1979–1995. https://doi.org/https://doi.org/10.1111/bjet.12685 (cit. on p. 43)

Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: Novice programmers and the solo taxonomy. *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 118–122. https://doi.org/10.1145/1140124.1140157 (cit. on p. 19)

Long, P., & Siemens, G. (2011). Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, *46*(5) (cit. on p. 39).

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., & Szabo, C. (2018). Introductory programming: A systematic literature review. *Proceedings Companion of the*

23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, 55–106. https://doi.org/10.1145/3293881.3295779 (cit. on p. 89)

M., K. (2020). Evaluating a first-year engineering course for project based learning (pbl) essentials [9th World Engineering Education Forum (WEEF 2019) Proceedings : Disruptive Engineering Education for Sustainable Development]. *Procedia Computer Science*, *172*, 364–369. https://doi.org/https://doi.org/10.1016/j.procs.2020.05.056 (cit. on p. 28)

Maclellan, E. (2001). Assessment for learning: The differing perceptions of tutors and students. *Assessment & Evaluation in Higher Education*, *26*(4), 307–318 (cit. on p. 40).

Maher, M. L., Latulipe, C., Lipford, H., & Rorrer, A. (2015). Flipped classroom strategies for cs education. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 218–223. https://doi.org/10.1145/2676723.2677252 (cit. on p. 45)

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2013). Towards a new massive multiplayer online role playing game for introductory programming. *Proceedings of the 6th Balkan Conference in Informatics*, 156–163. https://doi.org/10.1145/2490257.2490284 (cit. on p. 85)

Marasco, E., Moshirpour, M., & Moussavi, M. (2017). Flipping the foundation: A multi-year flipped classroom study for a large-scale introductory programming course. https://doi.org/10.18260/1-2--28372 (cit. on p. 45)

Masapanta-Carrión, S., & Velázquez-Iturbide, J. Á. (2018). A systematic review of the use of bloom's taxonomy in computer science education. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 441–446. https://doi.org/10.1145/3159450.3159491 (cit. on pp. 17, 18)

Mayring, P. (2000). Qualitative content analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *1*(2). https://doi.org/10.17169/fqs-1.2.1089 (cit. on p. 142)

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, *62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133 (cit. on p. 45)

Michael R., D., & Chen, S. (2006). Serious games: Games that educate, train, and inform (cit. on p. 73).

Micheuz, P., Pasterk, S., & Bollin, A. (2017). Basic digital education in austria – one step further. In A. Tatnall & M. Webb (Eds.), *Tomorrow's learning: Involving everyone. learning with and about technologies and computing* (pp. 432–442). Springer International Publishing. (Cit. on p. 15).

Miljanovic, M., & Bradbury, J. (2017). Robobug: A serious game for learning debugging techniques, 93–100. https://doi.org/10.1145/3105726.3106173 (cit. on p. 44)

Miller, M., Gronlund, N., & Linn, R. (2012). *Measurement and assessment in teaching*. Addison Wesley. https://books.google.at/books?id=bnx6MAEACAAJ. (Cit. on p. 39)

Miller, S. (2001). Workload measures. *National Advanced Driving Simulator. Iowa City, United States* (cit. on p. 81).

Moroney, W. F., Biers, D. W., Eggemeier, T. F., & Mitchell, J. A. (1992). A comparison of two scoring procedures with the nasa task load index in a simulated flight task. *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference (at NAECON) 1992*, 734–740 vol.2. https://doi.org/10.1109/NAECON.1992.220513 (cit. on p. 82)

Morris, M. G., & Dillon, A. (1997). How user perceptions influence software use. *IEEE software*, *14*(4), 58–65 (cit. on p. 82).

Mpungose, C. B., & Khoza, S. B. (2022). Postgraduate students' experiences on the use of moodle and canvas learning management system. *Technology, Knowledge and Learning*, *27*(1), 1–16. https://doi.org/10.1007/s10758-020-09475-1 (cit. on p. 36)

Nagrecha, S., Dillon, J. Z., & Chawla, N. V. (2017). Mooc dropout prediction: Lessons learned from making pipelines interpretable. *Proceedings of the 26th International Conference on World Wide Web Companion*, 351–359 (cit. on p. 156).

Nah, F. F.-H., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., & Eschenbrenner, B. (2014). Gamification of education: A review of literature. In F. F.-H. Nah (Ed.), *Hci in business* (pp. 401–409). Springer International Publishing. (Cit. on p. 26).

NASA, N. (1986). Task load index (tlx) v. 1.0 manual. *NASA, NASA-Ames Research Center Moffett Field* (cit. on p. 79).

Nazamud-din, A., Zaini, M., & Jamil, N. (2020). The relationship of affective, behavioral and cognitive engagements in esl higher learning classroom. *English Language Teaching and Linguistics Studies*, *2*, p48. https://doi.org/10.22158/eltls.v2n4p48 (cit. on p. 21)

Ng, E. (2018). Integrating self-regulation principles with flipped classroom pedagogy for first year university students. *Computers & Education*, *126*, 65–74. https://doi.org/https://doi.org/10.1016/j.compedu.2018.07.002 (cit. on p. 171)

Nicholson, J., Javed, Y., Dixon, M., Coventry, L., Ajayi, O., & Anderson, P. (2020). Investigating teenagers' ability to detect phishing messages. *2020 IEEE*

European Symposium on Security and Privacy Workshops (EuroS&PW), 140–149. https://doi.org/10.1109/EuroSPW51379.2020.00027 (cit. on p. 111)

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A cs unplugged design pattern. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 231–235. https://doi.org/10.1145/1508865.1508951 (cit. on p. 30)

Oeda, S., & Hashimoto, G. (2017). Log-data clustering analysis for dropout prediction in beginner programming classes. *Procedia computer science*, *112*, 614–621 (cit. on p. 157).

Onah, D. F., Sinclair, J., & Boyatt, R. (2014). Dropout rates of massive open online courses: Behavioural patterns. *EDULEARN14 proceedings*, *1*, 5825–5834 (cit. on pp. 33, 156).

Orlando, J. (2019). Kids need to learn about cybersecurity, but teachers only have so much time in the day. https://theconversation.com/kids-need-to-learn-about-cybersecuritybut-teachers-only-have-so-much-time-in-the-day-112136. (Cit. on p. 110)

Owen, E., & Baker, R. (2019). Learning analytics for games. (Cit. on p. 77).

Owston, R., York, D., & Murtha, S. (2013). Student perceptions and achievement in a university blended learning strategic initiative [Blended Learning in Higher Education: Policy and Implementation Issues]. *The Internet and Higher Education*, *18*, 38–46. https://doi.org/https://doi.org/10.1016/j.iheduc.2012.12.003 (cit. on p. 28)

Palaigeorgiou, G., & Papadopoulou, A. (2019). Promoting self-paced learning in the elementary classroom with interactive video, an online course platform and tablets. *Education and Information Technologies*, *24*(1), 805–823. https://doi.org/10.1007/s10639-018-9804-5 (cit. on p. 12)

Panagiotakopoulos, T., Kotsiantis, S., Kostopoulos, G., Iatrellis, O., & Kameas, A. (2021). Early dropout prediction in moocs through supervised learning and hyperparameter optimization. *Electronics*, *10*, 1701 (cit. on p. 157).

Pandas DataFrame. (2021). Pandas dataframe. *Pandas*. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html (accessed: 16.04.2021). (Cit. on p. 77)

Parson, D., & Haden, P. (2006). Parson's programming puzzles: A fun and effective learning tool for first programming courses. *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, 157–163 (cit. on p. 173).

Pelletier, K., McCormack, M., Reeves, J., Robert, J., & Arbino, N. (2022). *2022 educause horizon report, teaching and learning edition*. (Cit. on pp. 156, 157).

Peng, H., Ma, S., & Spector, J. M. (2019). Personalized adaptive learning: An emerging pedagogical approach enabled by a smart learning environment. In M. Chang, E. Popescu, Kinshuk, N.-S. Chen, M. Jemni, R. Huang, J. M. Spector, & D. G. Sampson (Eds.), *Foundations and trends in smart learning* (pp. 171–176). Springer Singapore. (Cit. on p. 2).

Peng, W., & Hsieh, G. (2012). The influence of competition, cooperation, and player relationship in a motor performance centered computer game. *Computers in Human Behavior*, *28*(6), 2100–2106. https://doi.org/10.1016/j.chb.2012.06.014 (cit. on p. 88)

Perrotta, C., Featherstone, G., Aston, H., & Houghton, E. (2013). Game-based learning: Latest evidence and future directions. *Slough: nfer*, 1–49 (cit. on p. 23).

Pirker, J., Dengel, A., Holly, M., & Safikhani, S. (2020). Virtual reality in computer science education: A systematic review. *26th ACM Symposium on Virtual Reality Software and Technology*. https://doi.org/10.1145/3385956.3418947 (cit. on p. 90)

Pirker, J., Riffnaller-Schiefer, M., & Gütl, C. (2014). Motivational active learning: Engaging university students in computer science education. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 297–302. https://doi.org/10.1145/2591708.2591750 (cit. on p. 42)

Pirker, J., Steinmaurer, A., & Karakas, A. (2021). Beyond gaming: The potential of twitch for online learning and teaching. In *Proceedings of the 26th acm conference on innovation and technology in computer science education v. 1* (pp. 74–80). Association for Computing Machinery. https://doi.org/10.1145/3430665.3456324. (Cit. on pp. 37, 173, 174)

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist*, *50*(4), 258–283. https://doi.org/10.1080/00461520.2015.1122533 (cit. on pp. 24, 53)

Podmurnyi, S. (n.d.). *6 growth strategies for elearning business - visartech blog*. Retrieved June 18, 2023, from https://www.visartech.com/blog/business-growth-strategies-for-elearning-marketplaces/. (Cit. on p. 31)

Quayyum, F. (2020). Cyber security education for children through gamification: Research plan and perspectives. *Proceedings of the 2020 ACM Interaction Design and Children Conference: Extended Abstracts*, 9–13 (cit. on p. 126).

Rahman, N., Sairi, I., Zizi, N., & Khalid, F. (2020). The importance of cybersecurity education in school. *International Journal of Information and Education Technology*, *10*, 378–382. https://doi.org/10.18178/ijiet.2020.10.5.1393 (cit. on p. 111)

Ratheeswari, K. (2018). Information communication technology in education. *Journal of Applied and Advanced research*, *3*(1), 45–47 (cit. on p. 26).

Renz, J., Staubitz, T., Pollack, J., & Meinel, C. (2014). Improving the onboarding user experience in moocs. *Proceedings EduLearn*, 3931–3941 (cit. on p. 82).

Ritchie, D., & Kernighan, B. (1988). *The C programming language*. Bell Laboratories. (Cit. on p. 12).

Rivadeneira, J., & Inga, E. (2023). Interactive peer instruction method applied to classroom environments considering a learning engineering approach to innovate the teaching - learning process. *Education Sciences*, *13*(3). https://doi.org/10.3390/educsci13030301 (cit. on p. 27)

Robins, A. V. (2019). Novice programmers and introductory programming. In S. A. Fincher & A. V. Robins (Eds.), *The cambridge handbook of computing education research* (pp. 327–376). Cambridge University Press. https://doi.org/10.1017/9781108654555.013. (Cit. on p. 17)

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(3), e1355 (cit. on p. 83).

Røynesdal, Ø., Magnus, J. H., & Moen, A. (2022). Pedagogical approaches and learning activities, content, and resources used in the design of massive open online courses (moocs) in the health sciences: Protocol for a scoping review. *JMIR Res Protoc*, *11*(5), e35878 (cit. on p. 157).

S., Y., N., S., V., H., & J., L. (2018). Pedagogy, practice, participation: Reimagining edtech onboarding. *29th Australasian Association for Engineering Education Conference 2018 (AAEE 2018)*, 705 (cit. on p. 83).

Saskatchewan Education. (1991). *Instructional approaches : A framework for professional practice*. https://books.google.at/books?id=kAMyNAEACAAJ. (Cit. on p. 23)

Sayuti, H., Ann, T., Saimi, W., Bakar, M., Dawawi, S., & Mohamad, M. (2020). Using gold standard project based learning for intermediate year three pupils to enhance english speaking skill: A conceptual paper. *Creative Education*, *11*, 1873–1889. https://doi.org/10.4236/ce.2020.1110137 (cit. on p. 28)

Schneider, M. (2016). Interface usability testing to maximize user onboarding experience. *PlotProjects-Geofencing Notifications for apps* (cit. on p. 83).

Schreurs, J., & Dumbraveanu, R. (2014). A shift from teacher centered to learner centered approach. *International Journal of Engineering Pedagogy (iJEP)*, *4*. https://doi.org/10.3991/ijep.v4i3.3395 (cit. on p. 20)

Seale, J., & Schoenberger, N. (2018). Be internet awesome: A critical analysis of google's child-focused internet safety program. *Emerging Library & Information Perspectives*, *1*, 34–58 (cit. on p. 129).

Security, H. N. (2021). The latest trends in online cybersecurity learning and training. https://www.helpnetsecurity.com/2021/11/17/trends-cybersecurity-training/. (Cit. on p. 109)

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B. B., Stephenson, C., & Verno, A. (2011). *Csta k–12 computer science standards: Revised 2011* (tech. rep.). New York, NY, USA, Association for Computing Machinery. (Cit. on p. 110).

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*. https://doi.org/10.1007/s10639-016-9482-0 (cit. on p. 13)

Seralidou, E., & Douligeris, C. (2021). Learning programming by creating games through the use of structured activities in secondary education in greece. *Education and Information Technologies*, *26*(1), 859–898. https://doi.org/10.1007/s10639-020-10255-8 (cit. on p. 44)

Serrano-Laguna, Á., Martínez-Ortiz, I., Haag, J., Regan, D., Johnson, A., & Fernández-Manjón, B. (2017). Applying standards to systematize learning analytics in serious games. *Computer Standards & Interfaces*, *50*, 116–123. https://doi.org/https://doi.org/10.1016/j.csi.2016.09.014 (cit. on p. 71)

Shahid, M., Wajid, A., Haq, K. U., Saleem, I., & Shujja, A. H. (2019). A Review of Gamification for Learning Programming Fundamental [ISBN: 9781728146829]. *3rd International Conference on Innovative Computing, ICIC 2019*, (November). https://doi.org/10.1109/ICIC48496.2019.8966685 (cit. on p. 43)

Shemshack, A., Kinshuk, & Spector, J. M. (2021). A comprehensive analysis of personalized learning components. *Journal of Computers in Education*, *8*(4), 485–503. https://doi.org/10.1007/s40692-021-00188-7 (cit. on p. 2)

Shi, J., Shah, A., Hedman, G., & O'Rourke, E. (2019). Pyrus: Designing a collaborative programming game to promote problem solving behaviors. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300886 (cit. on p. 90)

Short, E. J., Noeder, M., Gorovoy, S., Manos, M. J., & Lewis, B. (2011). The importance of play in both the assessment and treatment of young children. In *Play in clinical practice: Evidence-based approaches.* (pp. 264–289). The Guilford Press. (Cit. on p. 57).

Shoukry, L., Göbel, S., & Steinmetz, R. (2014). Learning analytics and serious games: Trends and considerations. *Proceedings of the 2014 ACM International*

Workshop on Serious Games, 21–26. https://doi.org/10.1145/2656719.2656729 (cit. on p. 71)

Shpeizer, R. (2019). Towards a successful integration of project-based learning in higher education: Challenges, technologies and methods of implementation. *Universal Journal of Educational Research*, *7*(8), 1765–1771 (cit. on p. 28).

Silva, L., Mendes, A. J., & Gomes, A. (2020a). Computer-supported collaborative learning in programming education: A systematic literature review. *IEEE Global Engineering Education Conference, EDUCON*, *2020-April*(May), 1086–1095. https://doi.org/10.1109/EDUCON45650.2020.9125237 (cit. on pp. 90, 103)

Silva, L., Mendes, A. J., & Gomes, A. (2020b). Computer-supported collaborative learning in programming education: A systematic literature review. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 1086–1095. https://doi.org/10.1109/EDUCON45650.2020.9125237 (cit. on p. 90)

Simon, B., Hundhausen, C., McDowell, C., Werner, L., Hu, H., & Kussmaul, C. (2019). Students as teachers and communicators. In S. A. Fincher & A. V. Robins (Eds.), *The cambridge handbook of computing education research* (pp. 827–858). Cambridge University Press. https://doi.org/10.1017/9781108654555.030. (Cit. on p. 85)

Slade, S., & Prinsloo, P. (2013). Learning analytics ethical issues and dilemmas. *American Behavioral Scientist*, *57*, 1510–1529. https://doi.org/10.1177/0002764213479366 (cit. on p. 84)

Smith, A., Toor, S., & Van Kessel, P. (2018). *Many turn to youtube for children's content, news, how-to lessons*. University of Melbourne. https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2018/11/PI_2018.11.07_youtube_FINAL.pdf. (Cit. on p. 36)

SpamBayes-Development-Team. (2002). *Spambayes: Bayesian anti-spam classifier written in python*. http://spambayes.sourceforge.net/. (Cit. on p. 117)

Spyropoulou, N., Demopoulou, G., Pierrakeas, C., Koutsonikos, I., & Kameas, A. (2015). Developing a computer programming mooc. *Procedia Computer Science*, *65*, 182–191 (cit. on p. 157).

Srinivasacharlu, A. (2020). Using youtube in colleges of education. *Education 3-13*, *8*, 21–24 (cit. on p. 37).

Stapleton, A. (2004). Serious games: Serious opportunities. *Australian Game Developers Conference* (cit. on p. 37).

Steinhorst, P., Petersen, A., & Vahrenhold, J. (2020). Revisiting Self-Efficacy in Introductory Programming. *Proceedings of the 2020 ACM Conference on Inter-*

national Computing Education Research, 158–169. https://doi.org/10.1145/3372782.3406281 (cit. on p. 20)

Steinmaurer, A., Pirker, J., & Gütl, C. (2019). Scool - game-based learning in computer science class a case study in secondary education. *International Journal of Engineering Pedagogy*, *9*(2). https://doi.org/10.3991/ijep.v9i2.9942 (cit. on pp. 7, 53, 56, 58, 63, 76, 91)

Steinmaurer, A., Tilanthe, A. K., & Gütl, C. (2022). Designing and developing a learning analytics platform for the coding learning game scool. In M. E. Auer & T. Tsiatsos (Eds.), *New realities, mobile systems and applications* (pp. 547–558). Springer International Publishing. (Cit. on p. 71).

Stoykova, V. (2015). Interactive environments for training in the higher education. *International Conference on e-Learning, e-Learning*, *15*, 268–273 (cit. on p. 26).

Stracke, C. M., & Trisolini, G. (2021). A systematic literature review on the quality of moocs. *Sustainability*, *13*(11), 5817 (cit. on p. 157).

Tacouri, H., & Nagowah, L. (2021). Code saga – a mobile serious game for learning programming. *2021 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 190–195. https://doi.org/10.1109/IoTaIS53735.2021.9628484 (cit. on p. 44)

Tedre, M., Simon, & Malmi, L. (2018). Changing aims of computing education: A historical survey. *Computer Science Education*, *28*(2), 158–186. https://doi.org/10.1080/08993408.2018.1486624 (cit. on p. 11)

Thakur, G., Olama, M. M., McNair, W., & Sukumar, S. R. (2014). Towards adaptive educational assessments: Predicting student performance using temporal stability and data analytics in learning management systems. https://www.osti.gov/biblio/1147724 (cit. on p. 36)

Thurston, A., Keere, K., Topping, K., Kosack, W., Gatt, S., Marchal, J., Mestdagh, N., Schmeinck, D., Sidor, W., & Donnert, K. (2007). Peer learning in primary school science: Theoretical perspectives and implications for classroom practice. *Electronic Journal of Research in Educational Psychology*, *5* (cit. on p. 27).

Tillmann, N., Bishop, J., Horspool, N., Perelman, D., & Xie, T. (2014). Code hunt: Searching for secret code for fun. *7th International Workshop on Search-Based Software Testing, SBST 2014 - Proceedings*. https://doi.org/10.1145/2593833.2593838 (cit. on p. 44)

Toda, A. M., Klock, A. C. T., Oliveira, W., Palomino, P. T., Rodrigues, L., Shi, L., Bittencourt, I., Gasparini, I., Isotani, S., & Cristea, A. I. (2019). Analysing gamification elements in educational environments using an existing gamification taxonomy. *Smart Learning Environments*, *6*(1), 16. https://doi.org/10.1186/s40561-019-0106-1 (cit. on p. 25)

Topping, K. J. (2005). Trends in peer learning. *Educational Psychology*, *25*(6), 631–645. https://doi.org/10.1080/01443410500345172 (cit. on p. 27)

Toro, U., & Joshi, M. (2012). Ict in higher education: Review of literature from the period 2004-201 1. *International Journal of Innovation, Management and Technology*, *3*(1), 20–23 (cit. on p. 26).

Tran, K.-N., Alazab, M., & Broadhurst, R. (2013). Towards a feature rich model for predicting spam emails containing malicious attachments and urls. *11th Australasian Data Mining Conference*, 161–171 (cit. on p. 113).

Tretinjak, M. F., & Andelic, V. (2016). Digital competences for teachers: Classroom practice. *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 807–811. https://doi.org/10.1109/MIPRO.2016.7522250 (cit. on p. 127)

Tucker, A. (2003a). *A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee* (tech. rep.). New York, NY, USA, Association for Computing Machinery. (Cit. on p. 14).

Tucker, A. (2003b). *A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee*. ACM. (Cit. on p. 31).

Turnbull, D., Chugh, R., & Luck, J. (2020). Learning management systems, an overview. *Encyclopedia of education and information technologies*, 1052–1058 (cit. on p. 35).

UNESCO. (2021). Covid-19 impact on education. *UNESCO*. https://en.unesco.org/covid19/educationresponse (accessed: 18.01.2021). (Cit. on p. 73)

Unicef. (2019). Growing up in a connected world understanding children's risks and opportunities in a digital age. https://www.unicef-irc.org/growing-up-connected. (Cit. on p. 109)

Verbert, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, *57*(10), 1500–1509 (cit. on pp. 74, 84).

Vergelis, M., Shcherbakova, T., Sidorina, T., & Kulikova, T. (2020). *Spam and phishing in 2019*. Retrieved January 22, 2021, from https://securelist.com/spam-report-2019/96527/. (Cit. on p. 107)

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. *Proceedings of the Tenth Annual Conference on International Computing Education Research*, 19–26. https://doi.org/10.1145/2632320.2632349 (cit. on p. 89)

Vihavainen, A., Luukkainen, M., & Kurhila, J. (2012). Multi-faceted support for mooc in programming. *Proceedings of the 13th annual conference on Information technology education*, 171–176 (cit. on p. 157).

Vitiello, M., Walk, S., Helic, D., Chang, V., & Gütl, C. (2018). User behavioral patterns and early dropouts detection: Improved users profiling through analysis of successive offering of mooc. *Journal of Universal Computer Science*, *24*(8), 1131–1150 (cit. on pp. 156, 163).

Vuorikari, R., Kluzer, S., & Yvey, P. (2022). *Digcomp 2.2: The digital competence framework for citizens - with new examples of knowledge, skills and attitudes* (Scientific analysis or review KJ-NA-31006-EN-N (online),KJ-NA-31006-EN-C (print)). Luxembourg (Luxembourg), Publications Office of the European Union. https://doi.org/10.2760/115376(online),10.2760/490274(print). (Cit. on p. 15)

Walvoord, B., & Anderson, V. (1998). *Effective grading: A tool for learning and assessment*. Wiley. https://books.google.at/books?id=yXTuAAAAMAAJ. (Cit. on p. 40)

Wanzer, D. L. (2021). What is evaluation?: Perspectives of how evaluation differs (or not) from research. *American Journal of Evaluation*, *42*(1), 28–46. https://doi.org/10.1177/1098214020920710 (cit. on p. 39)

Watson, J. (2008). *Blending Learning: The Convergence of Online and Face-to-Face Education* (tech. rep.). North American Council for Online Learning. https://files.eric.ed.gov/fulltext/ED509636.pdf. (Cit. on p. 30)

Wattanasoontorn, V., Boada, I., García, R., & Sbert, M. (2013). Serious games for health. *Entertainment Computing*, *4*(4), 231–247. https://doi.org/https://doi.org/10.1016/j.entcom.2013.09.002 (cit. on p. 37)

Wendel, V., Gutjahr, M., Göbel, S., & Steinmetz, R. (2013a). Designing collaborative multiplayer serious games. *Education and Information Technologies*, *18*(2), 287–308. https://doi.org/10.1007/s10639-012-9244-6 (cit. on p. 88)

Wendel, V., Gutjahr, M., Göbel, S., & Steinmetz, R. (2013b). Designing collaborative multiplayer serious games. *Education and Information Technologies*, *18*(2), 287–308. https://doi.org/10.1007/s10639-012-9244-6 (cit. on p. 85)

What is cybersecurity? (2022). IBM. https://www.ibm.com/topics/cybersecurity. (Cit. on p. 109)

Wheeler, D. L. (2016). Zxcvbn: Low-Budget password strength estimation. *25th USENIX Security Symposium (USENIX Security 16)*, 157–173. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler (cit. on p. 135)

Wilkowski, J., Deutsch, A., & Russell, D. M. (2014). Student skill and goal achievement in the mapping with google mooc. *Proceedings of the First ACM Conference on Learning @ Scale Conference*, 3–10. https://doi.org/10.1145/2556325.2566240 (cit. on p. 34)

Wisniewski, B., Zierer, K., & Hattie, J. (2020). The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in Psychology*, *10*, 3087. https://doi.org/10.3389/fpsyg.2019.03087 (cit. on p. 41)

Xinogalos, S., Ivanović, M., Savić, M., & Pitner, T. (2020). Technology-enhanced learning in programming courses, role of. In A. Tatnall (Ed.), *Encyclopedia of education and information technologies* (pp. 1726–1736). Springer International Publishing. https://doi.org/10.1007/978-3-030-10576-1_218. (Cit. on p. 36)

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7 (cit. on p. 12)

Yan, A., Lee, M. J., & Ko, A. J. (2017). Predicting abandonment in online coding tutorials. *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 191–199 (cit. on pp. 158, 165).