Department of Computer Science

Game-Related Learning and Exposure in Computer Science

Lassi Haaranen

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 13/2019

Game-Related Learning and Exposure in Computer Science

Lassi Haaranen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall AS1 of the school on February 8th 2019 at noon.

Aalto University
School of Science
Department of Computer Science
Learning + Technology Research Group

Supervising professor
Professor Lauri Malmi

Preliminary examiners
Associate Professor Nickolas Falkner, The University of Adelaide, Australia
Associate Professor Juho Hamari, Tampere University of Technology, Finland

Opponent
Kristian Kiili, PhD, Tampere University, Finland

Aalto University publication series
DOCTORAL DISSERTATIONS 13/2019

ISBN 978-952-60-8385-8 (printed)
ISBN 978-952-60-8386-5 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-8386-5

Unigrafia Oy
Helsinki 2019

Finland

Printed matter
4041-0619

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Lassi Haaranen

Name of the doctoral dissertation
Game-Related Learning and Exposure in Computer Science

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 13/2019

Field of research Computing Education Research

Manuscript submitted 9 May 2018 **Date of the defence** 8 February 2019

Permission to publish granted (date) 21 November 2018 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Given the importance of computers, and by extension computer science (CS), in contemporary society, it is crucial to provide the best possible education in the field. This dissertation looks at two different game-related approaches in computer science education: how digital games and gaming communities expose people to CS concepts; and how game-related approaches can be used to improve computer science education (CSE) in universities.

In order to structure this dissertation as well as future research, we present a classification of game-related approaches focused specifically on CSE. We see three broad approaches with games and CS: gameful approaches (e.g. gamification), designing and programming games, and entertainment games with learning content.

Modern digital games are complex systems that the players need to learn and master. With certain games, programming and CS concepts can be used to enhance the playing experience. This provides a spark of interest in computing for some that might eventually lead to studying CS or related fields in a university. This phenomenon was studied by interviewing students as well as through reflection essays in which the students not only outlined how games had piqued their interest in computers and CS but also how game development as a career was appealing to some of them.

Modern games are not just played in isolation or with a group of friends on the same couch. Instead, there are online gaming communities in which games are discussed and the actual gameplay is also recorded and either broadcasted live to an audience or uploaded to a video sharing service. We investigated two different online gaming communities in which CS and programming were a part of the games featured. We did this by gathering data on the interactions between the audience members. Through the discussion that we analyzed, we found that these communities are places where people are exposed to CS and programming. Moreover, these communities are places where both experienced programmers and newcomers come and discuss CS topics.

Using games in formal education was researched with two approaches. Firstly, we implemented two software systems, Acos and Daechschen, to support gamification in online learning management systems. The core design principles behind these systems strive for interoperability and extensibility so that they continue to be relevant and used in fast pacing ecosystems of modern online learning tools. Secondly, we investigated implementing achievement badges on a course with Daechschen. We looked into students' reactions to the badges and found out that overall it was slightly positive with a large group of students being indifferent to them.

Keywords computer science education, game-related approaches, informal learning, gamification, achievement badges, online learning management systems, interoperability

ISBN (printed) 978-952-60-8385-8 **ISBN (pdf)** 978-952-60-8386-5

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2019

Pages 218 **urn** http://urn.fi/URN:ISBN:978-952-60-8386-5

Aalto-yliopisto

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tiivistelmä

Tekijä
Lassi Haaranen

Väitöskirjan nimi
Peleihin pohjautuva tietotekniikan löytäminen ja oppiminen

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 13/2019

Tutkimusala Tietotekniikan opetustutkimus

Käsikirjoituksen pvm 09.05.2018 **Väitöspäivä** 08.02.2019

Julkaisuluvan myöntämispäivä 21.11.2018 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Tietotekniikka on kriittinen osa modernia yhteiskuntaa ja tietotekniikan opetus on tästä syystä erityisen tärkeää. Tässä väitöskirjassa tutkitaan kahta peleihin pohjautuvaa lähestymistapaa liittyen tietotekniikan opetukseen. Ensimmäinen näistä tarkastelee, kuinka jotkin verkossa olevat peliyhteisöt liittyvät ohjelmointiin ja tietotekniikkaan, ja joissa ihmiset tulevat tietoiseksi ohjelmoinnista ja ohjelmointiin liittyvistä aiheista. Toinen lähestymistapa keskittyy siihen, kuinka pelillistämistä voidaan käyttää hyväksi tietotekniikan yliopisto-opetuksessa.

Työn jäsentämiseksi ja tulevan tutkimuksen avuksi esitämme tavan luokitella peleihin pohjautuvia lähestymistapoja erityisesti keskittyen tietotekniikan opetukseen. Nämä ovat: pelilliset menetelmät (esim. suoritusmerkit), pelien ohjelmointi ja ohjelmointiin liittyviä konsepteja sisältävät viihdepelit.

Modernit digitaaliset pelit ovat monitasoisia kompleksisia järjestelmiä, jotka usein vaativat pelaajalta tietoista harjoittelua. Joissain peleissä on sisäänrakennettuna mekaniikkoja, jotka sisältävät konsepteja ja käsitteitä tietotekniikasta ja ohjelmoinnista. Nämä pelit itsessään innostavat joitakin ihmisiä käyttämään tietotekniikkaa ja joissakin tapauksissa tämä innostus johtaa tietotekniikan opiskeluun yliopistossa. Tätä ilmiötä lähestyttiin analysoimalla reflektioesseitä sekä haastattelemalla tietotekniikan opiskelijoita heidän suhteestaan pelaamiseen ja peleihin.

Pelaaminen ei nykyään tapahdu pelkästään yhdessä tilassa yksin tai kavereiden kanssa vaan enenevissä määrin internetiin on muodostunut pelaamiseen keskittyviä verkkoyhteisöjä. Näissä yhteisöissä peleistä keskustelemisen lisäksi pelaamista nauhoitetaan ja jaetaan videonjakopalveluiden kautta. Tätä ilmiötä lähestyttiin tutkimalla kahta eri verkkoyhteisöä, joissa tietotekniikka ja ohjelmointi liittyi peleihin. Näissä yhteisöissä käytyjen keskustelujen pohjalta kävi ilmi, että nämä ovat paikkoja joissa ihmiset kohtaavat tietotekniikkaan liittyviä konsepteja. Tämän lisäksi ne ovat paikkoja, joissa sekä kokeneet että aloittelevat ohjelmoijat keskustelevat ohjelmoinnista.

Pelien käyttöä tietotekniikan koulutuksessa yliopistotasolla tutkittiin kahdella tavalla liittyen pelillistämiseen eli pelien osien hyödyntämiseen ei-pelillisissä konteksteissa. Pelillistämistä varten kehitettiin kaksi avoimen lähdekoodin ohjelmistoa: Acos ja Daechschen. Näiden ohjelmistojen keskeisenä suunnitteluperiaatteena on ollut laajennettavuus ja yhteensovitettavuus, joka on erityisen tärkeää moderneissa verkko-oppimisympäristöissä. Lisäksi opiskelijoiden reaktioita pelillistämiseen tutkittiin tietorakenteiden ja algoritmien kurssilla käyttäen Daechschen-järjestelmää. Yleisesti opiskelijat suhtautuivat suoritusmerkkeihin hieman positiivisesti.

Avainsanat tietotekniikan opetus, pelilliset menetelmät, vapaamuotoinen oppiminen, pelillistäminen, suoritusmerkit, verkko-oppimisympäristöt, yhteensovitettavuus

ISBN (painettu) 978-952-60-8385-8 **ISBN (pdf)** 978-952-60-8386-5

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2019

Sivumäärä 218 **urn** http://urn.fi/URN:ISBN:978-952-60-8386-5

Isälle

Preface

When I first started in the Learning+Technology research group in 2013 to work on my master's thesis I had no idea that it would eventually lead to the very book you are now holding. I have learned a great deal from many different people throughout the years – and not just about computing education research. My work has been guided and enriched by the many interesting conversations and collaborations I've had along the way.

First of all, I wish to extend my sincerest gratitude to Professor Lauri Malmi for his guidance and for encouraging me to pursue the lines of research that I found most interesting and promising. Additionally, I wish to thank D.Sc. Lasse Hakulinen and Adjunct Professor Ari Korhonen as the ones who recruited me to LeTech and guided me through my master's thesis.

I also wish to thank all of my co-authors in the publications included in this dissertation: Petri Ihantola, Lasse Hakulinen, Ari Korhonen, Teemu Sirkiä, Rodrigo Duran, Päivi Kinnunen, and Lauri Malmi. Clearly, without your collaboration and contributions this would not have been possible!

I wish express gratitude to all the people that frequented room B207 where most of this work was done: Petri, Teemu S., Aleksi, Juha, Otto, and others. From that room, I also wish to thank Teemu Lehtinen in particular for fruitful collaborations that were happening alongside this dissertation process. I later migrated to room A135 where I had the pleasure of enjoying many interesting conversations with Rodrigo.

I am also grateful for my pre-examiners Associate Professors Nickolas Falkner and Juho Hamari. Their many helpful comments and suggestions

improved the final product considerably. I am also honored to have Dr. Kristian Kiili as my opponent.

I have had a lot of support from many people in many ways. Friends have been crucial during this process. I wish to thank Niko, Antti, Osku, Satu, and others for their friendship and support. A shout out to #vbp05s as well! Additionally I want to thank Dad & Seija, as well as Grandma Alli and my sisters, for their encouragement in this process! And finally, thank you Frieda! Your encouragement and care reliably supported me during challenging times – and helped to reduce the number of typos in this work.

Gwangju, Korea, December 2018,

Lassi Haaranen

Contents

Preface 3

Contents 5

List of Publications 7

Author's Contribution 9

1. Introduction 11
- 1.1 Gaps in Current Research 13
- 1.2 Research Questions 14
- 1.3 Structure 18

2. Background on Learning and Exposure in Different Contexts 23
- 2.1 Learning, Education, and Contexts 24
- 2.2 Constructivism and Situated Learning 28
- 2.3 Motivation and Engagement 30
- 2.4 Outreach 32
- 2.5 Summary 34

3. Background on Games and Gamification 35
- 3.1 Defining Games and Game-Related Learning 36
- 3.2 Gamification 39
- 3.3 Games as a Context for Learning and Serious Games 44
- 3.4 Entertainment Games with Learnable Content 48
- 3.5 Summary of Game-Related Learning in Computer Science Education 54

4. Games and Exposure to Computer Science **55**
4.1 People Watching Other People Play 56
4.2 Computer Science in Online Gaming Communities 62
4.2.1 Programming in Gaming-related Videos on YouTube 62
4.2.2 Discussions During a Game Programming Live-stream Event 67
4.2.3 Combined Analysis of Comment Data from YouTube and Twitch.tv 69
4.3 Exposure to Games and Studying Computer Science 72
4.4 Conclusions 74

5. Gameful Approaches in Formal Computer Science Education **77**
5.1 Learning Management Systems and Smart Learning Content 78
5.2 Systems Supporting Gamification Schemes 79
5.2.1 Acos 80
5.2.2 Daechschen 84
5.3 Utilizing Achievement Badges on a Data Structures and Algorithms Course 85
5.4 Conclusions 88

6. Discussion **91**
6.1 Conclusions and Contributions 92
6.1.1 **RQ1** How do game-related approaches and communities relate to exposure to computer science in informal settings? 94
6.1.2 **RQ2** How can game-related approaches be utilized in formal CS education? 97
6.2 Validity and Trustworthiness 100
6.3 Constructive Research 103
6.4 Ethical Considerations 104
6.5 Future Work 105

References **107**

Publications **119**

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Lassi Haaranen, Rodrigo Duran. Link Between Gaming Communities in YouTube and Computer Science. In *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU '17)*, Porto, Portugal, pages 17-24, April 2017.

II Lassi Haaranen. Programming as a Performance: Live-streaming and Its Implications for Computer Science Education. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*, Bologna, Italy, pages 353-358, June 2017.

III Lassi Haaranen, Rodrigo Duran. Computer Science in Online Gaming Communities. *Communications in Computer and Information Science volume 865 CSEDU 2017*, pages 279-299, 2018.

IV Lassi Haaranen, Päivi Kinnunen, Lauri Malmi. Reaching Out Through Games: Informal Outreach in Computer Science. ***manuscript***. *11 pages*, 2018.

V Teemu Sirkiä, Lassi Haaranen. Improving online learning activity interoperability with Acos server. *Software: Practice and Experience*, pages 1657–1676, 2017.

VI Lassi Haaranen, Petri Ihantola, Lasse Hakulinen and Ari Korhonen. Software Architectures for Implementing Achievement Badges – Practical Experiences. In *Proceedings of the 2014 International Conference on Teaching and Learning in Computing and Engineering (LaTiCE '14')*, Kuching, Malaysia, pages 41-46, April 2014.

VII Lassi Haaranen, Petri Ihantola, Lasse Hakulinen and Ari Korhonen. How (not) to Introduce Badges to Online Exercises. In *Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)*, Atlanta, GA, USA, pages 33-38, March 2014.

Author's Contribution

Publication I: "Link Between Gaming Communities in YouTube and Computer Science"

I provided the original research idea and implemented the scripts that received the data. The analysis was done equally by both of the authors. I led the writing efforts and both of the authors contributed to the final article.

Publication II: "Programming as a Performance: Live-streaming and Its Implications for Computer Science Education"

I was the sole author and researcher on this article.

Publication III: "Computer Science in Online Gaming Communities"

I led the writing of the article but both authors contributed equally to the reporting. The research was done collaboratively and both authors contributed equally to it.

Publication IV: "Reaching Out Through Games: Informal Outreach in Computer Science"

I provided the original idea for the paper and proposed the research. I directed analyzing the data with contributions from the second author. The writing process was led by me in collaboration with the other authors.

Publication V: "Improving online learning activity interoperability with Acos server"

Both of the authors contributed equally to the design and implementation of the Acos system. Similarly, the article was a joint effort and both of the authors contributed equally to the writing.

Publication VI: "Software Architectures for Implementing Achievement Badges – Practical Experiences"

I designed and implemented the second system, Daechschen, presented in the article. I led the requirement collection and designed the interoperable system to work with existing A+ software. I led the writing of the article along with the other authors.

Publication VII: "How (not) to Introduce Badges to Online Exercises"

I implemented the system used for the experiment and led the design process and requirement gathering, with valuable comments from other authors. The design of the experiment was also led by me with contributions from other authors. I directed the data analysis and writing process of the article in collaboration with the other authors.

1. Introduction

The importance of computers and computer science (CS) to modern society cannot be overstated. Software and algorithms govern numerous aspects of our lives. From this arises an ever growing need for people who master the art of computer programming. Thus, it is vital to investigate how to attract people to study the field and how CS education can be improved.

This dissertation discusses two separate, yet related, themes both in the domain of CS. First of them being how people are *exposed* to computer science. Where do they encounter it and what are some of the reasons they decide to study it further? Once they do end up studying CS in universities, how can we *improve and enhance* the students' experiences is the second theme explored.

Both of these themes are enormous topics and plenty of research has been published about exposure and outreach [16, 79], as well as novel tools [33, 114, 116], and other instructional improvements [53, 105]. To limit the scope of this work, these two topics are investigated in how they relate to games. Using game-related approaches to improve computer science education (CSE) is investigated through implementing software systems to enhance online learning activities – and exploring students' reactions to these systems.

The theme of exposure to CS is investigated by looking into how the games that people play might affect their choice of major. Additionally, there are games and gaming communities online that are intertwined with CS concepts and topics. These communities are investigated as well, especially with regard to the exposure to CS they provide.

Together these two approaches, utilizing games to improve CSE and exposure to CS through games, are discussed as *game-related learning and exposure in computer science*.

While knowledge on its own is a goal worth pursuing, there is also a practical reason for investing in CS education and CS education research. Projections to 2020 estimate that there will be a large number of CS positions that need to be filled globally. In Europe there will be a need for 800,000 CS professionals [13] and in the United States, more than a million vacancies will be open by 2020 [27].

Given the demand for people capable of understanding CS and related skills, there is an additional incentive to improve and enhance CS education. The field that is concerned with this is called *computing education research* (CER) which has emerged as a discipline in recent decades [113]. The growing discipline of CER is the main context of this dissertation and the general body of knowledge to which this work aims to contribute instead of larger fields of game studies or general educational research.

In order to follow this work, it is beneficial to understand how the emergence of social media, especially sharing live, as well as, recorded video has shaped the hobby, and for some the profession, of playing video games. Hence, it is worth briefly discussing the broader cultural context in which this dissertation is situated. Especially relevant is the growth of online gaming communities and how watching other people play games – whether live or from a recording. The increased popularity of playing or watching other people play games has created a massive industry in and of itself.

The growth of gaming has led to a whole host of online gaming communities. This is relevant for the present discussion due to the fact that in some of these communities CS is discussed and gamers are exposed to CS content even if they might not have sought that out on purpose.

Another trend arising in recent years is gamification – utilizing elements from games in non-game contexts. One aspect where gamification has been researched actively is in educational contexts. It was the focus of investigations on how to improve and enhance CSE in universities.

Naturally, these topics will be discussed in more detail in the coming chapters. For now, it is enough to indicate that games as a hobby, as well as a profession, are growing. And this work is situated in this emerging and evolving cultural landscape.

1.1 Gaps in Current Research

The overall motivation for the research is to both improve current CS education with game-related approaches, as well as, investigate how game-related approaches can be utilized to expose more potential students to CS.

Games have been utilized in CS for a long time in various forms. In 2005, Kelleher & Pausch [72] created a taxonomy of various programming environments aimed at novices. Under their category of empowering systems, they list entertainment systems as one option. However, they do not go further into classifying games and relating their content to CS education. This dissertation aims to enrich that classification by providing different types of game-related approaches to CS education.

Notably, some of the approaches, especially when it comes to live-streaming games and programming, where not yet available in 2005. Games have been used in CS education in universities in many ways [15, 128, 96, 71]. However, to the best of our knowledge, there is no existing literature for the game-related approaches in CS investigated in informal contexts. Overall, learning in informal settings has not received much attention from the CER community. Though, some prior research [76, 77] does suggest that games might be relevant when choosing a major to study.

Another central theme of this dissertation is how to bring game-related approaches to formal CS education. More specifically, how to support game-related approaches from a *technical point of view*. Modern learning management systems are complex pieces of software, and increasingly there is a call for more interoperability to enable different types of smart content to be integrated for the benefit of the learner [21].

Gamification, and especially achievement badges, have been studied in various educationals contexts including CS. Often the research has been implemented as between-subject studies, utilizing a control and treatment group. We were not aware of any publications that experimented with badges in a within-subject study, especially in CS education. This gap had also been noted in prior literature [57]. We fill this gap by providing a within-subject experiment with achievement badges on a data structures and algorithms course.

1.2 Research Questions

The overall objective of this work is building a broader understanding of **how game-related approaches are relevant to computer science education?** That is to say, our aim is to highlight different approaches in utilizing games that can be used to enhance CSE as well as expose more people to CS. Relevancy in this sense should not be thought of as a scale to be measured, but rather in whether it is useful for CSE or not. Though CS and games are at the center of the focus, this dissertation does not try to cover all the different ways in which the two are intertwined. Rather, we focus on two distinct themes: *exposure to CS in online gaming communities* (**RQ1**) and *utilizing gamification in CS education* (**RQ2**).

The methodologies used are described briefly with the research questions in this section. For more detailed discussion refer to the included publications.

RQ1 How do game-related approaches and communities relate to exposure to computer science in informal settings?

This research question is situated in informal learning contexts. In other words, how people might encounter CS and programming through their hobbies of playing games and participating in online gaming communities. It is further divided into questions **RQ1.1** and **RQ1.2** which are introduced here and then further discussed in Chapter 4.

There are some inherent difficulties when it comes to researching learning in informal contexts. Participants are often at home when the potential learning and exposure to computer science happens. Additionally, their identities are unknown and their backgrounds can vary a lot. Since these types of experiences are difficult to investigate in authentic environments (e.g. people watching live-streams at home), alternative approaches need to be utilized.

RQ1.1 *How do online gaming communities facilitate exposure to computer science and learning programming?*

The growth of online gaming communities provides opportunities to expose oneself to CS by encountering it in games – either by playing or watching someone else play. At the same time, some content creators in these communities have a background in computer science, programming, and software development and they release content (videos, live-streams, tutorials, and so on) that teach and disseminate CS content.

To shed light on this phenomenon and investigate **RQ1.1**, two online communities, one on YouTube and the other one on Twitch.tv were examined. To be more specific, we looked at how viewers and participants in these communities discuss matters related to CS.

The data for **RQ1.1** was gathered from publicly available sources. The collection happened either via logging conversations as they happened or by implementing custom programs that collected data automatically.

For **PUB I**, comments from YouTube on selected videos relating to games and computer science were collected and subsequently analyzed. Based on the data distinct categories of comments emerged which are described and discussed. Similarly in **PUB II**, grounded theory was used to analyze discussion logs gathered from Twitch.tv from a stream that related to games and programming.

The qualitative analysis in **PUB III** uses the data collected in **PUB I** and **PUB II**. By applying similar methodology to two different data sets we build a categorization of CS discussion in gaming communities that is more general.

Overall, we find in **PUB I** that gaming communities in YouTube provide a way to expose people to CS concepts and build a classification of different types of discussions relating to CS that the viewers had. Similarly, in **PUB II** we show that live-streaming programming during a game development competition exposes new people to programming and the discussions during the live-streams provide newcomers an opportunity to meet more experienced programmers. Finally, in **PUB III** we combine these data sets and provide a more general classification of the types of discussions participants in these communities have regarding CS.

Most of the research conducted within **RQ1.1** was qualitative in nature and utilized grounded theory. Grounded theory is used within social sciences, particularly when researching new phenomena [49] so it is particularly fitting. Additionally, more frequent use of grounded theory has been called for in computing education research [75].

Alternative approaches could have been used as well. For example, finding participants from the communities and doing semi-structured interviews would have yielded potentially interesting rich data. While we had plans of doing this, we were ultimately unable to carry out interviews. We tried to contact live-streamers in order to conduct surveys and recruit viewers for interviews on their channel but received no response. However, the chosen approach allowed us to form an overall picture of the phenomenon that serves as a starting point for further research.

RQ1.2 *How playing games in adolescence is related to choosing to major in computer science?*

The motivation for this research came from reading first-year students' reflections on their past and current gaming hobbies and how gaming had an impact on their choice of career. Through discussions with colleagues and classmates, similar experiences have come to light: games have played an important role in learning how to use a computer and later on to write computer programs.

It seems that for many, games are the drivers behind early computing experiences that lead to the desire to further understand the inner workings of computers and, perhaps, to study the field. These experiences also reflect those of the author of this work. There is also some prior work suggesting this link between computer games and interest in studying fields closely related to CS [76, 77].

To explore this, we analyzed students' reflections regarding their reasons for choosing CS as their major in **PUB IV**. We highlight the various ways in which games contributed to their decision regarding their discipline. As with publications relating to **RQ1.1**, the analysis is based upon grounded theory. The data for the analysis comes from two sources: reflection essays written by freshmen CS major students and semi-structured interviews. In addition to the qualitative analysis, we provide descriptive statistics regarding the prevalence of students that recounted games being influential to their current field of study.

Based on the analysis in **PUB IV**, we show three types of ways in which game-playing in early on in life can affect selecting a career in CS. Firstly, gaming experiences provide a way to get interested in computers and the skills required to use them. Secondly, games can contain programming and thus trigger interest towards CS, as well as, learning to program games can be a goal to pursuit. Finally, game development as an industry is seen as an appealing career choice by some students.

RQ2 How can game-related approaches be utilized in formal CS education?

While **RQ1** is focused on game-related learning in informal learning contexts, in other words, not in a classroom, **RQ2** focuses on learning CS in a university context. The investigation under this research question is divided into finding suitable software solutions to implement gamification systems (**RQ2.1**) and empirically evaluating students' perceptions of these systems (**RQ2.2**).

To investigate technical solutions to support gamification in universities, a constructive approach was used: the requirements for these systems were identified, the systems were constructed, and finally evaluated. Whereas the students' reactions to gamification on a CS course was investigated with qualitative and quantitative approaches.

RQ2.1 *What kinds of technical solutions are needed to support gamification in computer science education?*

Software systems are needed in order to experiment with various gamification mechanics in formal education. **RQ2.1** focuses on how these systems might be implemented based on the work done in **PUB V** and **PUB VI**. Modern online learning environments are increasingly complex and there is significant overhead in implementing new content. The solutions presented aim at easing the workload of developers while providing an improved learning experience.

Both **PUB V** and **PUB VI** use a constructive approach and focus on the software designed and implemented. The first step of the constructive process was to gather requirements from stakeholders that the software should fulfil. The design phase was followed by implementation, especially focusing on interoperability which was identified as the most crucial feature based on the requirements gathered. Its significance is also described in prior literature [21, 29]. Finally, the systems were evaluated in how they operated in practice and by collecting and analysing the perceptions of different stakeholders. In addition, a technical evaluation of the performance of the system described in **PUB V** was also conducted.

Overall, both publications answer the research question in a concrete way by showing a possible way to solve the problem of interoperability in a modern online learning environment. In **PUB VI** the solution is focused around the needs of a single learning management system and thus is more localized. This was later on improved in the design and implementation of the system described in **PUB V**, where interoperability of multiple learning management systems is also considered.

RQ2.2 *How do students react to adding achievement badges to online exercises?*

In order to gather evidence on the effects of gamification, an experiment was conducted to gauge the students' reactions to achievement badges in an online learning management system.

In **PUB VII** both qualitative and quantitative methods were used to investigate the effects of implementing the badge system on a CS course. Quantitatively, we measured students' performance using a within-subject approach, to complement earlier between-subject research on the same course [57]. Additionally, we gathered and evaluated log data from the platform regarding students' performance and interest in the badges. We also analyzed students' feedback regarding the badges on the course that was gathered by likert scale questions and freeform text feedback.

Based on the results in **PUB VII**, we find that for the majority of the students achievement badges were of little to no significance. A small minority was very vocally against the addition of badges to the course whilst a slightly larger portion of students seemed to really enjoy them. Overall, we did notice a small positive correlation in the interest towards badges to the amount of badges collected.

1.3 Structure

Table 1.1 encapsulates the overall structure of this dissertation regarding the research questions. It also relates the publications in this dissertation

Table 1.1. Research questions mapped to sections of the dissertation and publications

	Section	Publications
RQ1.1 How do online gaming communities facilitate exposure to computer science and learning programming?	4.2	**PUB I** **PUB II** **PUB III**
RQ1.2 How playing games in adolescence is related to choosing to major in computer science?	4.3	**PUB IV**
RQ2.1 What kinds of technical solutions are needed to support gamification in computer science education?	5.2	**PUB V** **PUB VI**
RQ2.2 How do students react to adding achievement badges to online exercises?	5.3	**PUB VII**

to the specific research questions. Similarly, Figure 1.1 shows the main sections of the work, which are described in the following paragraphs.

Chapter 2 focuses on providing the framing for three different contexts of learning: formal, non-formal, and informal. It also situates this work into the broader context of education research. The aim here is to provide a framework that can be used to view the present work.

Chapter 3 discusses the background with games and game-related learning and provides a taxonomy of game-related approaches in CS education. Additionally, a new type of game-related approach – entertainment games with learnable content – is introduced.

Chapter 4 provides an overview of different types of online communities in which people watch gaming either live or as recordings. Then, CS learning and exposure in online gaming communities is discussed based on **PUB I**, **PUB II**, and **PUB III** in Section 4.2. Section 4.3 discusses exposure to CS through past gaming experiences and its influence on the decision to study CS, which relates to analysis in **PUB IV**. The chapter concludes with a discussion of the key findings in Section 4.4.

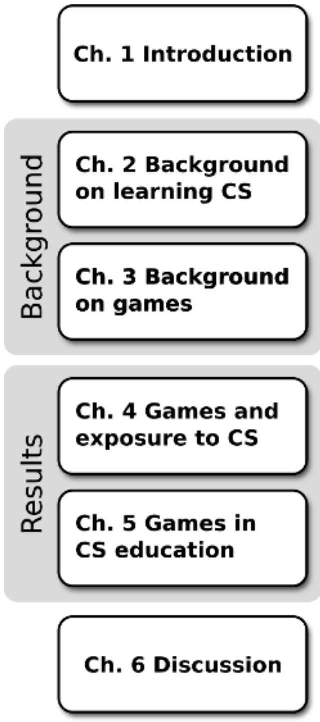

Figure 1.1. Overall structure of the dissertation

Chapter 5 begins with a brief overview of modern online learning management systems in Section 5.1. Interoperability and the ability to incorporate game-related approaches to CS education is discussed as well. Then, in Section 5.2 two software solutions for gameful approaches are discussed that relate to **RQ2.1**.

First, we describe Acos, introduced in **PUB V**, an interoperable system that supports incorporating online learning activities into different LMSs via multiple protocols. After this, an earlier system described in **PUB VI**, Daechschen, is discussed. As a system, Daechschen was focused solely on enabling achievement badges in a specific LMS. To answer **RQ2.2**, an empirical investigation of students' reactions to badges earned in a CS course is explored in Section 5.3, based on **PUB VII**. The chapter concludes in Section 5.4 where different software systems for game-related approaches are considered and the results of the empirical investigation are discussed.

Finally, in Chapter 6 we conclude the work by discussing conclusions arising from the publications and the dissertation as a whole. Validity and trustworthiness of the work are also discussed. The chapter concludes by highlighting interesting avenues for future research.

2. Background on Learning and Exposure in Different Contexts

When it comes to learning there is an abundance of theoretical frameworks to choose from. For this dissertation and the work related to it, a constructivist approach was adopted. Constructivism is in the mainstream of current educational research and hence it is particularly well represented in computer science education research.

Constructivism is based on the idea that the learner (who may or may not be a student) actively constructs knowledge based on his or her prior understanding and experience [39]. However, the process of knowledge construction does not happen in a vacuum, and hence the different contexts for learning also discussed in this chapter.

We consider learning in the following contexts: formal, non-formal, and informal [40]. Formal context refers to learning in the traditional institutions that one thinks of hearing the word 'education', such as universities and schools. Traditionally CER has focused on research in the formal learning context. Whereas, non-formal education, is intentional learning happening outside formal learning, such as different clubs and organizations centered around hobbies. Informal learning might or might not be intentional and it typically happens alongside other activities.

The aim of this chapter is to situate the present work into different learning contexts. The goal is to utilize existing frameworks to view CER – focusing on game-related approaches. Next, in Section 2.1, learning and education in its various contexts is defined. After this, learning is considered from a constructivist point of view particularly from a situated learning perspective in Section 2.2. The focus of Section 2.3 is in describing various theoretical approaches to motivation and engagement in educational settings, with an emphasis on games in education. Finally, in Section 2.4 research focusing on outreach is discussed.

2.1 Learning, Education, and Contexts

When reviewing definitions that are used for learning, De Houwer et al. [30] concluded that there have been two approaches. One of them, a *functional definition*, states that learning is the changes in behavior resulting from experience. Complementary to this, *a mechanistic view of learning* sees learning as the changes that occur in the organism based on experience. Finally, De Houwer et al. define learning as the *changes in the behavior of an organism that are the result of regularities in the environment of that organism*. This definition expands from previous ones and has three components:(1) The change in behavior, (2) regularity in the environment, and (3) the causal relationship between these two.

Given this broad definition of learning, it is natural to ask where does learning happen. Education is generally considered to be the process that facilitates learning. But education as a concept is broader than the formal education that is offered by institutions. A century ago Dewey considered the different types of education that exist:

"Hence one of the weightiest problems with which the philosophy of education has to cope is the method of keeping a proper balance between the informal and the formal, the incidental and the intentional, modes of education" [35]

The terms, informal, formal, incidental, and intentional reflect the variety of educational contexts where learning can happen. Learning may be pursued as a primary goal making it very intentional. A typical example of this type of learning might be a student pursuing a degree in a university. A degree offered by an institution would also be considered formal education. On the other hand, learning can also happen as an unintended by-product of an activity. In that case it would be more incidental than intentional. An example could be playing video games that also happen to include computer science content that the player needs to understand in order to succeed in the game.

There are no clear boundaries when it comes to intentionality and formality of learning and education. This has also been noted in the literature, there are no universally agreed upon definitions of what informal, non-formal, and formal learning are [85]. The rest of this section focuses on outlining the different definitions that have been used for these terms and aligning the current work within these learning contexts.

The Commission of the European Communities (CEC) has sought to provide definitions for informal, non-formal, and formal learning [40]. Their approach focuses on the key aspects in differentiating the learning contexts: structure, certification, and intentionality. In their definition *formal learning* is:

*"Learning typically provided by an education or training institution, **structured** (in terms of learning objectives, learning time or learning support) and **leading to certification**. Formal learning is **intentional** from the learner's perspective"* [40] (emphasis added)

Formal education is perhaps the easiest to recognize and define due to its structure. It is provided by an institution (e.g. a university), has structure (e.g. courses, lectures, exams), and it leads to certification (e.g. master's degree in computer science). Also, it is intentional from the students' point of view (e.g. requiring entrance exams or tuition fees).

This definition of formal learning can be contrasted to the CEC's definition of *informal learning*:

*"Learning resulting from daily life activities related to work, family or leisure. It is **not structured** (in terms of learning objectives, learning time or learning support) and **typically does not lead to certification**. Informal learning **may be intentional** but in most cases it is non-intentional (or 'incidental'/random)"* [40] (emphasis added)

Examples of this type of learning could be learning in relation to hobbies (e.g. learning to play an instrument), and/or in the form of consuming online resources (e.g. discussion forums, videos) that relate to a particular interest. This type of learning does not lead to certification nor is there a clear structure or schedule. If the learner has specific knowledge or skills that he or she wishes to acquire that would mean that learning is intentional and there is a goal to be pursued.

This definition of informal learning categorizes it to be mostly incidental. However, this is not always the case. Marsick and Watkins consider incidental learning a subcategory of informal learning [88], wherein incidental learning alongside another goal:

"Incidental learning is defined as a byproduct of some other activity, such as task accomplishment, interpersonal interaction, sensing the organizational culture, trial-and-error experimentation, or even formal learning."

Non-formal learning is a term that is also used in the literature. For some, for example Malcolm et al. [85], it is mostly interchangeable term with informal learning and the key difference is that the two terms are used in different disciplines. Alternatively, the European Commission has defined *non-formal learning* as distinct from informal learning:

"Learning that is ***not provided by an education or training institution*** *and typically* ***does not lead to certification****. It is, however,* ***structured*** *(in terms of learning objectives, learning time or learning support). Non-formal learning is* ***intentional*** *from the learner's perspective"* [40] (emphasis added)

An example of non-formal learning could be professional development events that many technology companies offer. Typically, they happen after work hours and involve demonstrations or talks. They are structured (there's a time and place) and there are learning goals (e.g. a talk focusing on unit testing within a particular software framework). Those who participate do it out of desire to learn resulting from personal or professional interest. The key differences from formal education are that they do not involve certification and they are not offered by an educational institution. Another example of non-formal learning are structured activities organized by hobby groups. For instance, members of a sports club attend regular, structured exercise sessions that involve group and individual learning.

Malcolm et al. devised four aspects to consider when differentiating between the formality and informality of a learning context: process, location and setting, purpose, and content [85]. *Process* refers to characteristics such as whether there's a mentor (formal) or not (informal) and how assessment is handled. If the physical *location* of learning is restricted (e.g. a classroom) or there's a specific schedule to follow, that also implies more formal learning. *Purpose* entails whether the learning is the result of deliberate focus and whether it leads to certification or if it is done for the pursuit of other interests such as hobbies. Finally, the *content* aspect refers to whether knowledge is learned for expertise and practice or for the purpose of developing a new skill due to pursuing an interest such as in the case of hobbies.

There are many similarities between the aspects listed in the CEC document and the four aspects Malcolm et al. proposed. Both approaches consider the intentionality of the learning as well as process/structure of the learning and whether there is official certification involved. The key difference between the two is that Malcolm et al. also consider the *content* of learning

Focusing purely on informal learning, Marsick & Volpe identified key features of informal learning [87]. They consider it to be something that is integrated into daily routines and which is not typically highly conscious. They also consider it as a haphazard process that is influenced by chance. One of the features of informal learning that they emphasize is the importance of others in informal learning: *"Informal learning is enhanced when people's chances for meeting new people and ideas are increased."* [87]

Previous research on informal learning has focused on adult education and especially learning in the workplace in general (e.g. [85, 88, 89]) or for a specific profession (e.g. sports coaches [97]). Research in informal learning outside of the work environment seems to have been neglected in the literature, though changes in technology and the rise of online communities have been noted in studies focused on learning at the workplace at the beginning of the millenium:

"... technology is changing the face of organizations and having an impact on the nature of informal and incidental learning. In fact, given the distributed, asynchronous nature of technology-facilitated interactions, more may be learned incidentally by learners reading between the lines." [89]

While the specific details in defining informal, non-formal, and formal learning differ, there are also many similarities. In this work the term *informal learning is used to refer to learning that happens outside educational institutions, is driven by internal desire, and is not highly structured*. This context is relevant for the publications **PUB I**, **PUB II**, and **PUB III**. The other context for learning that is relevant for publications **PUB V**, **PUB VI**, and **PUB VII** is *formal learning, which refers to learning happening within an institution, consists of deliberate practice, and* *leads to a degree or certification*. Figure 2.1 summarizes the learning contexts, as well as situates the publications in this thesis into them.

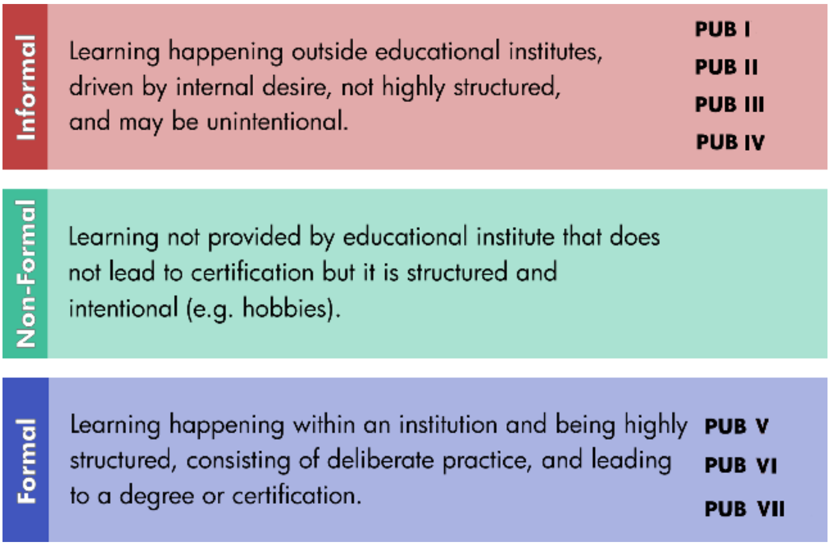

Figure 2.1. Formal, non-formal, and informal learning contexts with the publications in this dissertation categorized into them.

2.2 Constructivism and Situated Learning

Constructivism is an educational theoretical framework that considers learning from the point of view that an individual is not an empty slate but has previous knowledge and experiences that affect the learning process. The learner is an active participant who constructs knowledge instead of passively receiving it from a book or a teacher [67, 17].

Constructivism underpins a great deal of computer science education research. When looking into publications in major CSE forums between the years 2005-2011 Malmi et al. found that constructivism and subtheories based on it were the most common ones to be utilized [86]. Applying constructivism to computer science education, however, is not always straightforward. Ben-Ari [17] noted this by considering the epistemology of computer science and how it differs from that of physics by students not having a preconceived notion of how computers work. Though he concludes that the basic principle of constructed knowledge applies to CS as well.

When designing learning environments from a constructivist point of view, Jonassen et al. [67] have argued that there are four salient aspects to consider: context, constructions, collaboration, and conversation.

By *context*, they refer to the features of the 'real world' that are pertinent to the learning task at hand. These include physical, organizational, cultural, social, and political issues. *Construction* refers to the construction of knowledge resulting from the individual's experience, either in the 'real world' or in a learning environment. *Collaboration* refers to the process of knowledge construction among learners and how that aids in developing, testing, and evaluating hypotheses. *Conversation* is the way that learners negotiate plans for the task at hand. Conversation is required to construct meaning as knowledge acquisition is mainly mediated by language.

Lave and Wenger have pioneered the theory of situated learning [81] within the constructivist school of thought. At the core of the theory is the idea that learning should be situated in real-world scenarios. Situated learning posits that there are communities of practice, in essence the practitioners of a particular profession. At the core of these communities are experts; novices are at the periphery of the communities. Novices learn and become part of the community through what Lave and Wenger call *Legitimate Peripheral Participation* or LPP. Through LPP novices first participate in simple tasks (peripheral) that are relevant to the community (legitimate). Through practice and increasing knowledge novices can move from the periphery to a more central role in the community and become experts.

Situated learning and communities of practice have also been investigated in online communities. Specifically, Henri & Pudelko [60] researched virtual communities of interest. They propose a model that describes the various types of virtual communities based on their intentionality and strength of social bonds. An example of these communities are virtual *communities of interest*. They form as people with interest in a particular topic gather together. The participants in these communities identify as having more interest in the topic than the members of the public. To formalise knowledge and strengthen group identity, these online communities often produce documents, such as Frequently Asked Questions (FAQs). The communities discussed in Chapter 4 and in **PUB I**, **PUB II**, and **PUB III** can be seen as online communities of interest.

There are also *Goal-oriented communities of interest* [60]. The difference between these communities is that they exist due to an external mandate and have a limited lifespan. Members of these communities identify more with the goal of the community, e.g. finishing a project, than with their group identity. Members of these communities are experts that have been recruited with the purpose of sharing knowledge.

A *learners' community* [60] is a community formed by a group of students that share a course or a similar activity. These communities are typically lead by a tutor who encourages collaboration. The focus of these communities is on learning relevant content knowledge and the communities themselves spring up and disappear alongside courses.

The final community that Henri & Podenko identified is *community of practice* [60]. The members of these communities are already part of a community of practice in the real world. These virtual communities are not focused on a specific task but rather consist of professionals working in a particular field.

Situated learning is not without its critics. For example, Anderson et al. [10] reviewed empirical studies related to aspects of situated learning. They showed that the claims of situated learning, such as that instruction should happen in a complex social environment, was not always supported by research. They conclude that while some learning is context-dependent there is also learning that is independent of contexts.

We use the theoretical constructs of LPP and the factors of context, constructions, collaboration, and conversation from Jonassen et al. [67] when we consider the gaming communities and activities in **PUB I**, **PUB II**, and **PUB III**. Specifically, we focus on the conversation in the context Youtube videos where programming concepts are discussed inside entertainment games and the conversations that happened during a rapid game development contest that was streamed live.

2.3 Motivation and Engagement

Typically, motivation has been dichotomized into *intrinsic* and *extrinsic* [110] motivation. Overall the distinction between these two hinges on whether an activity is pursued because it is inherently interesting (intrinsic motivation) or if it is being done for a *separable outcome* (extrinsic motivation), such as a reward or avoiding sanctions. A simple example of a positive separable outcome (and thus extrinsic motivation) would be the salary that one receives from work. In education, an example of avoiding sanctions would be studying for an exam to avoid failing the course or getting a bad grade.

Gagné & Deci [46] have presented *self-determination theory* (SDT) which views motivation as a continuum that ranges from amotivation (complete lack of motivation) through extrinsic motivation (with various types of regulations of motivation) to intrinsic motivation. The key difference to previous cognitive evaluation theories is that extrinsic motivation is comprised of varying degrees of autonomy. SDT considers four different levels of self-determination in extrinsic motivation (in order of increasing autonomy): external regulation, introjected regulation, identified regulation, and integrated regulation.

External regulation in SDT refers to activity that is done for external reasons only, for example working only when supervised. In turn, *introjected regulation* refers to the situation where the regulation itself is driving the person, e.g. when one works in order to feel as a worthy person. *Identified regulation* refers to the type of of extrinsic motivation where the person identifies on a personal level with the values and goals of the activity, e.g. a teacher might consider grading essays to not be intrinsically motivating task but giving feedback and guidance is aligned with their values as a teacher. Finally, *integrated regulation* refers to the case where a person has integrated the regulation as an important part of their identity, e.g. a teacher might consider being a teacher inherently valuable and consider that to be an important part of their identity. [46]

Ryan et al. [111] utilized SDT to look into the motivational factors of playing video games. Through multiple studies they investigated factors of autonomy, competence, and relatedness as to the reasons why video games are psychologically attractive. They concluded that in the case of single-player games, autonomy and competence are motivating factors and that in multiplayer games all three factors were important.

Myriad techniques have been studied to increase motivation and engagement in the classroom. In recent years, gameful approaches and particularly gamification (see Section 3.2) have been used in education. The gamification of educational systems and classroom experiences typically focuses on extrinsic motivation by providing a system of extrinsic rewards, such as badges, points or leaderboards (e.g. **PUB VII**, [32, 57]). In a review of the literature Hamari et al. [5] concluded that gamification does indeed seem to provide some positive outcomes in terms of engagement. However, they note that there seems to be underlying confounding factors which require further study. It should also be noted, that in their meta-analytical review Deci et al. [31] concluded that giving out extrinsic rewards undermines existing intrinsic motivation.

In addition to SDT, *goal orientation* has been used to explain motivation and has been shown to predict performance and motivation in students in educational settings [42]. Goal orientation is concerned with the *"why and how people are trying to achieve various objectives and refer to overarching purposes of achievement behavior"* [68]. Kaplan & Maehr ([68]) identified the mastery and performance. Students with a mastery orientation seek knowledge for its own sake, and as the name implies, wish to master the subject. Performance oriented students, in contrast, wish to demonstrate their competence relative to others. Performance orientation can be further divided into *performance approach* (seeking to demonstrate competence to peers) and to *performance avoidance* (trying to avoid demonstrating incompetence). Finally, avoidance oriented students wish to minimize effort and generally avoid challenging tasks [12]. Different goal orientations have been shown to affect the outcomes of educational interventions. In particular, students with performance approach have been shown to be more interested in achievement badges in CS contexts [12]. Additionally, goal orientation has been shown to have an effect on gamification approaches in other domains as well [6].

2.4 Outreach

Somewhere between formal and informal learning and education lies outreach. The goals and methods of outreach can vary but generally the aim is to increase awareness and knowledge of a subject or a field of study, and often to recruit more people to it.

To promote more strategic approaches to educational outreach particularly in the STEM (Science, Technology, Engineering, Mathematics) field

Ward [129] has proposed a STEM-EO model. The model is comprised of three elements: domains, goals, and dimensions. *Domains* refer to the different stakeholders in an outreach activity, such as K-12 students, teachers, or the general community. *Goals* are the primary objectives of a particular activity. Examples of goals could be to increase STEM career awareness or STEM literacy, additionally the goals often have to do with STEM recruitment and retention. Finally, the *dimensions* refer to the specific features of the outreach activity. These include things as such the context (whether the activity is aimed at informal or formal learning), how it is organized (e.g. face-to-face sessions, distance outreach), and instructional styles (e.g. inquiry-based learning, demonstrations).

Various game-related outreach activities in the domain of computer science have been proposed over the years. For example, Lakanen et al. [79, 80] have organized and studied a summer course on game programming as a form of outreach. They've demonstrated that while students generally tend to view programming more positively after the course this did not extend to everyone who took this voluntary course. In particular, they found out that there are student clusters they've labeled as *experimenters* and *unsatisfieds* who are not interested in computer science.

Other game-based approaches have focused on improving the perception of CS amongst students of different fields. To this end, Zorn et al. [136] found out that their modified version of Minecraft had a positive impact on the perception of CS in psychology students.

Gameful approaches to computer science outreach have also been used without computers. The "CS Unplugged" project aims to increase interest in CS in primary school students by employing *"activities, games, magic tricks and competitions"* [16] without the use of computers. This type of approach sidesteps issues such as computer availability or the details of a particular programming language and focuses on the principles behind computer science. An example of this type of approach is the international challenge on informatics and computer science thinking, Bebras, (`http://www.bebras.org/`) which aims *"to promote Informatics (Computer Science, or Computing) and computational thinking among school students at all ages."*.

2.5 Summary

In this chapter, we have introduced three contexts in which learning takes place, formal, non-formal, and informal. The formal learning context is relevant for **PUB V**, **PUB VI**, and **PUB VII**. While the informal learning context is relevant for **PUB I**, **PUB II**, **PUB III**, and **PUB IV**.

The effect of achievement badges on students behavior and engagement on a CS course is the topic of **PUB VII**. Goal-orientation is discussed as one possible option to explain the results observed. The other publications related badges are more technical in nature and are informed by prior literature in software engineering discussed in Chapter 5.

Situated learning and LPP [81] are considered in online gaming communities where games and CS concepts are intertwined **PUB I**, **PUB II**, **PUB III**). We also consider how gaming hobby is related to choosing to study CS in a formal context and how this might be utilized in addition to other novel outreach activities discussed in Section 2.4.

3. Background on Games and Gamification

A wide variety of approaches combining games and education have been used ranging from using aspects of games in non-game contexts, known as gamification [34], to using whole games that aim at teaching, referred to as serious games [92].

In education, *gamification* refers to adding game mechanics or game-like elements into exercises and courses in a formal learning context. The elements can be an implicit part of the course (e.g. various points that one needs to collect from exercises) or more explicit gamification systems such as awarding achievement badges in online learning management systems for completing some specific actions. Gamification rose as a trend in educational research (and elsewhere) during 2010-2012 and Section 3.2 summarizes research in this area focusing on computer science education.

Serious games are another way of integrating games and computer science. The term is used to refer to games which have been specifically designed to achieve some ulterior goal beyond enjoying a game or having fun. Serious games exist for various purposes but naturally, in this context, Section 3.3 highlights previous research in regarding serious games in general and in the domain of computer science education. Section 3.3 also covers programming games and using games as a context for programming courses.

Various aspects and approaches using gamification and serious games in computer science education have been researched. However, learning programming and computer science concepts through commercial games meant for entertainment has received little to no attention from the CSE research community, especially in the context of informal learning. Section 3.4 presents a novel categorization of entertainment games based on how they incorporate or interact with computer science concepts. These games are also important in the phenomena of watching other people play games which have become hugely popular over the last ten years. Through playing, or watching other people play, games that relate to computer science in some way provide a way to gain exposure to computer science and this phenomenon is discussed further in Chapter 4.

However, before looking at various ways of mixing games and computer science the concept of a 'game' is first discussed in Section 3.1. The section also defines terminology used in this dissertation regarding games and learning. Section 3.5 concludes the chapter and provides a summary of the concepts discussed.

3.1 Defining Games and Game-Related Learning

Defining the word 'game' is not straightforward. In fact, the philosopher Wittgenstein argued that games are indefinable [132]. In response to Wittgenstein's work, Suits argued that games are definable and with three criteria: prelusory goal, constitutive rules, and lusory attitude [120]. Prelusory goal (from Latin 'ludus' meaning structured play, discussed further in Section 3.2) defines a specific state of affairs that the players pursue within the confines of constitutive rules. The constitutive rules restrict the players actions such that they are not the most efficient. For example, achieving checkmate in chess is trivial if you move your opponents pieces as well. Finally, with lusory attitude he referred to the act of accepting the rules and thus making play possible. He provided a summary of the definition as: *"playing a game is the voluntary attempt to overcome unnecessary obstacles."*

Stenros [118] reviewed the literature on how 'game' has been defined since the 1930s. His work explicitly focuses on the definition of a 'game' instead of defining 'video game'. Based on the 63 definitions gathered, he provided ten topics that differentiate the various definitions of games.

What Is a Game?

One of the categories in definitions of a 'game' that Stenros ([118]) identified is whether the game is viewed as an *artifact* (e.g. [121]) or an *activity*, such as in Abt's definition: *"a game is an activity among two or more inde-*

pendent decision-makers seeking to achieve their objectives in some limiting context" [7]. In the context of this dissertation, a game is viewed as an artifact as opposed to an activity. The term game can generally refer to a variety of different artifacts such as board games or role-playing games. However, in this dissertation game is used to refer to digital games and more specifically computer games. Additionally, the activity of playing these games is also referred to as 'gaming'.

In the realm of digital games, Keith Burgun has proposed the definition of a game to be *"a system of rules in which agents compete by making ambiguous decisions"*. The concept of an agent in this definition covers both human players and computer controlled agents. He further argues that there four are categories of interactive digital entertainment: *interactive systems*, *puzzles*, *contests*, and *games*. [22]

The basis for the different types of media is an *interactive system*, such as a flight simulator. The other categories add something to the interactive systems. *Contests* add a competition, an example of a contest could Guitar Hero (requiring to hit keys in correct order and timing) or other skills-based systems. On the other hand, puzzles add a problem to be solved. *Puzzles* have a 'correct' solution and to steps to acquire it do not involve interesting decisions. Finally, *games* add interesting decisions which in this context refer to the fact that the player has a choice or a dilemma, usually with a trade-off, which alters the game state and affects future decisions.

As game designer Sid Meier [9] phrased it *"Games are a series of interesting decisions"*, further clarifying that *"Good decisions are situational. There's a very key idea that when the decision is presented to the player, ideally it acts in an interesting way with the game situation"*. In this sense, sudoku, much like a system of equations, is a puzzle. In order to solve a particular sudoku steps can be completed in a different order but ultimately there is a correct solution and whichever path the solver to took achieve it has no real difference in the outcome. This can be contrasted to games, which present decisions to the player, such as whether a character should stay slightly longer in a safer area or proceed to a high-risk place that has higher rewards. Tied to the idea of decisions is the fact that they imply consequences which makes them meaningful.

McGonigal approached the definition of games from a trait perspective. She identifies four traits that games have: goals, rules, feedback system, and voluntary participation [90]. A *goal* gives a sense of purpose to the players giving them a reason to play. *Rules* make pursuing this goal non-trivial and interesting. The *feedback system* gives the player information on how well they are progressing towards the goal. And finally, the *voluntary participation* ensures that *"the freedom to enter or leave a game at will ensures that intentionally stressful and challenging work is experienced as safe and pleasurable activity"* [90].

Both Burgun's and McGonigal's definitions of games have implications for using gameful approaches in educational contexts. Typically exercises, for example in mathematics, have a correct solution and steps to achieve that are taught. This means that they are more akin to a puzzle and the learner does not have meaningful decisions to make in the process. This is also true for many gamified exercises. Similarly, the participation in exercises is often mandatory, whether they are gamified or not. Furthermore, tests and exams can hardly be called pleasurable or non-stressful activities which rules out voluntary participation in the sense that McGonigal [90] was referring to it, even if they involve awarding points and grades.

Game-Based Learning, Gameful Approaches, and Game-Related Learning

Though there is no single definition of game-based learning, it generally refers to using games to somehow enhance learning. The use of these games might involve games that are designed for educational purposes or just incorporating elements from games to other educational activities. Since it is sometimes hard to pin down what makes an approach game-based learning as opposed to, for instance, experiential learning [78] with playful elements, an alternative term is used in this dissertation.

In his dissertation, Hakulinen [55] defined 'gameful approach' as *"an umbrella term for any method that has some game-like features, ranging from gamification to fully-fledged games."*. The term includes gamification (Section 3.2) and serious games (Section 3.3). However, it implies that there is a deliberate approach that is being taken which does not fit well with entertainment games with learning content which are discussed in Section 3.4.

Another similar term to gameful approaches is game-based learning (GBL) [103, 106]. There doesn't seem to be a commonly accepted definition of GBL, but generally, it can be understood as roughly synonymous with gameful approaches in education. Though, there is a stronger focus on using complete games (serious games or educational games) as opposed to gamification.

To encompass all different aspects and approaches when it comes to learning computer science through games, we adopt the term *game-related learning* (GRL). Figure 3.1 summarises the different terms used with games and learning and it relates them to each other in the way that they are used in this dissertation. Furthermore, the figure also includes the special case in computer science where programming games (or watching someone else program games) itself can be an educational activity.

One additional game-related term used is 'games with a purpose'. Though this term is sometimes used to mean a serious game, it is typically used to mean games that use some form of "human computation" [8]. These are games where the developers are looking to do some form of computation through the actions of the players.

These games with a purpose are typically linked to citizen science activities. Citizen science projects are *"thousands of research projects are engaging millions of individuals – many of whom are not trained as scientists – in collecting, categorizing, transcribing, or analyzing scientific data."* [18]. A typical example of these types of games is Foldit (`https://fold.it/`) where players solving how proteins fold in three-dimensional space. Utilizing the players' computation in these types of games can lead to novel discoveries [73]. However, games with a purpose fall outside the scope of this thesis.

3.2 Gamification

Given that defining what we mean we use the word 'game' is difficult (see Section 3.1), it is similarly difficult to define 'gamification'. Various definitions have been proposed. For example, Huotari & Hamari approached

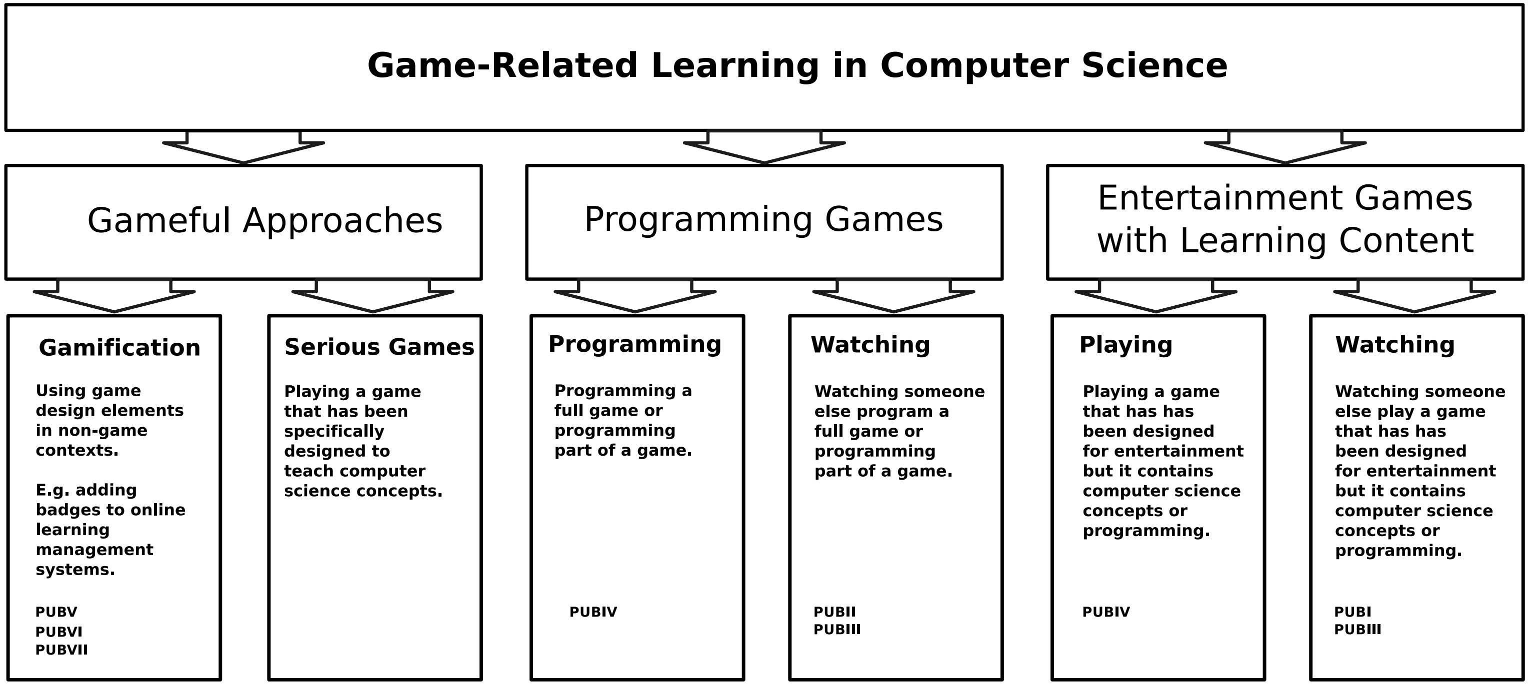

Figure 3.1. Game-related learning in computer science education

the definition from a marketing perspective and defined gamification to be *"a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation"* [62]. For this dissertation, we use the term gamification as it has been defined by the most cited definition by Deterding et al. [34]: *"the use of game design elements in non-game contexts"*. They situate gamification within these axes: playing versus gaming and whole versus parts (see Figure 3.2). Playing and gaming refer to the terms *paidia* (freeform, unstructured play) and *ludus* ("gaming" with rules, goals, and perhaps competition) as used by Caillois [23]. In this sense, gamification is more akin to gaming with rules and structure. But at the same time, gamification doesn't constitute a whole as games do, instead, it is using game design elements in some other system.

On the gaming side of the axis in Figure 3.2 examples of whole games would include complete games, meant for entertainment whether digital or not, such as chess or League of Legends [109]. On the parts side of the gaming axis lies various gamification schemes such as those investigated, for example, in [32, 56, 6].

On the opposing side of gaming is playing or paidia. Deterding et al. divide playful approaches to those that utilize it fully – toys – and to those that utilize parts of play – playful design. Playful design or playfulness has been described as a *"mindset whereby people approach every day, even mundane, activities with an attitude similar to that of paidia – as something not serious, with neither a clear goal nor real-world consequences"* [83]. A common example of this, enhance regular stairs by adding speakers and pressure sensors to make each stair play a note and thus hopefully encouraging people to walk the stairs instead of using an escalator.

Gamification, and similar approaches, can be thought of as *persuasive technologies*. They are a broad category of approaches designed to "nudge" users towards particular activities or choices. The approaches to persuasion are varied, or as IJsselsteijn et al. defined them: *"Persuasive technology is the general class of technologies that purposefully apply psychological principles of persuasion – principles of credibility, trust, reciprocity, authority and the like – in interactive media, in the service of changing their users' attitudes and behavior"* [65].

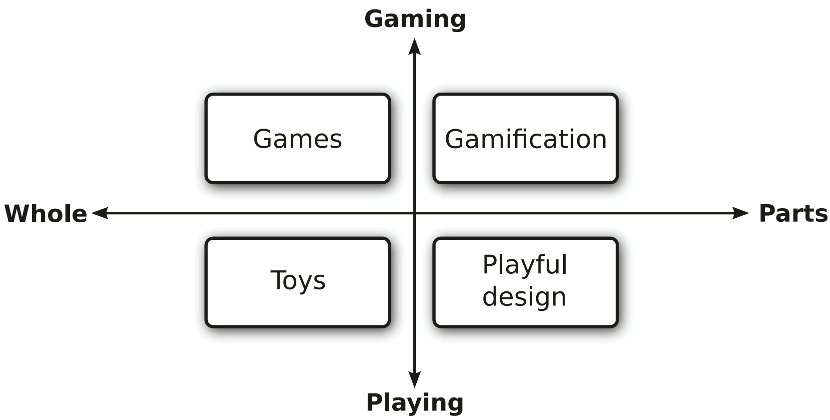

Figure 3.2. Situating gamification, adapted from [34]

Fogg [45] has suggested a behavior model to aid in the design of persuasive technologies. In the behavior model there are three components to consider when trying to nudge a user towards a particular behavior: motivation, ability, and triggers. Motivation, whether high or low, obviously affects the behavior. And similarly, whether a user has the ability (knowledge, physical capabilities, etc.) to act in a certain way affects the behavioral outcomes. The last one, trigger, is the final component that initiates a behavior. He argues that even if a user wants to complete an activity (motivation) and is capable of doing so (ability), a trigger that pushes the user to start an activity is still required. In an educational context, various gamification schemes such as achievement badges can act as such triggers.

Gamification can be applied in different ways but points, leaderboards, and badges seem to be the most common approach [5]. Achievement badges, or just badges, are an optional and visual reward for some activity. Hamari & Eranti [58] have proposed a framework for designing and evaluating achievement. In it, they identify three parts to badges: visual, name, and description (shown also in Figure 3.3). The name denotes and identifies the badge from other badges, as does the visual. The description is the part that explains to the user what conditions need to be satisfied and what action has to be taken in order to achieve that particular badge.

While gamification can be applied in many different contexts, it is often investigated in educational settings and particularly on online courses.

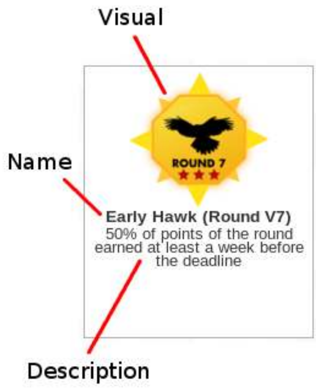

Figure 3.3. Anatomy of an achievement badge

Hamari et al. [5] conducted a review on the effects of gamification that included different contexts for gamification with educational settings being by far the most common. They note that overall gamification seems to have mostly positive outcomes. However, they note that the qualitative studies reviewed also highlighted that some people heavily dislike gamification schemes. This finding is aligned with results in **PUB VII** where the student population seemed mostly indifferent towards badges, with smaller groups of students liking or disliking them.

Denny [32] conducted a large scale (N>1000) controlled trial on the effects of achievement badges on a student question-answer platform called PeerWise. They did not discover adverse effects to using achievement badges. Furthermore, they found that the students answered more questions and were active more active on the platform. They also noted a positive correlation between student visits to the page that showed the students' badges and the number of badges achieved. A similar correlation was also observed in **PUB VII**.

Gamification, and especially achievement badges, have been studied in a computer science education context, as well. Ibáñez et al. [63] investigated the use of gamification on a programming course using C. They concluded that the use of badges was particularly effective. Their results showed a positive increase in engagement as well as a moderate improvement in learning outcomes. One of the interesting behaviors they observed was that some students continued working even after earning maximum points for an exercise if there were still badges they could earn.

There are also efforts to understand how different students react to gamification schemes based on their goal orientation. Goal orientation refers to different strategies and preferences in individuals goal selection and approach to attaining that goal [98]. Hakulinen & Auvinen [56] looked at badges on a Data Structures and Algorithms course from the perspective of goal orientation. They found that those students who reported high motivation regarding badges (based on the end of course questionnaire) scored higher on mastery-intrinsic, mastery-extrinsic, and performance-approach orientations. Notably, these students were high performing *before* the implementation of badges. Interestingly, they also found a group of students with avoidance orientation who had a low motivation towards badges. These results echo those of Abramovich et al. [6] where they found that middle-school students with different levels of prior knowledge pursued different badges on an applied mathematics course.

Badges and other similar approaches have been criticized due to focusing on extrinsic motivation with possible negative effects on intrinsic motivation [31]. Similarly, badges can have unwanted side effects such as compromising carefulness in exercises due to badges focusing on speed [57].

Factors related to motivation and goal orientation might explain why literature reviews (e.g. [5]) have found studies with both positive and negative effects, as well as no significant effects, from badges. Altogether, it seems that there is great importance in *how* and what *kind of* badges are implemented into the curriculum. Furthermore, the ability to turn off gamification features, whether badges or otherwise, that are extraneous to the course content should be considered.

3.3 Games as a Context for Learning and Serious Games

While badges have been criticized for their focus on extrinsic motivation, other gameful approaches that focus on increasing intrinsic motivation and interest towards computer science have also been studied.

Serious games have been defined as *"games that do not have entertainment, enjoyment, or fun as their primary purpose"* [92]. This means that serious games have been created for some other purpose, typically, but not always, education. Other purposes include, for example, military recruitment with an America's Army platform (`https://www.americasarmy.com/`) being an early example from 2002. Serious games have also been used in other contexts, such as in advertising, sometimes referred to as advergaming, healthcare, and activism. [36]

There have been many varied approaches to using serious games in education. In 2012, Connolly et al. [28] conducted a systematic literature review on the effects of games in education and game-based learning. They found that serious games indeed have been used in many different disciplines but especially in health, business, and social issues. They also identified a range of outcomes from games that were investigated. Outcomes related to learning (knowledge acquisition/content understanding) were amongst the most common themes in addition to motivational outcomes. They concluded that there is a large range of research on different impacts when it comes to GBL. However, they also highlighted the difficulty of classifying learning outcomes. Finally, since the research methods and evidence on the efficacy of GBL were so varied, they called out for more randomized controlled trial research to strengthen the evidence.

An updated systematic literature based on the earlier work by Connolly et al. [28] was published in 2016 [19]. In it, they found stronger evidence for the positive outcomes linked to games. Games for learning, their term for serious games with learning as an intended outcome, comprised about half of the 143 papers. Unsurprisingly, knowledge acquisition was the most common outcome that was intended. The other half of the articles they reviewed focused on entertainment games. With these, the intended outcomes were more varied, ranging from affective and behavioral changes to developing soft and social skills. Most of these games were categorized with the subject area as "not relevant" so they were not tied to a particular curriculum subject.

They concluded that there has been increased interest in research on the positive impact that games can have on learning, and though they found stronger evidence in this review, they call for more systematic research on what game features most promote learning and knowledge acquisition.

Finally, they note that even with *"the intense interest in games, it is important to realize that developing games for learning can be very complex and costly and still provides significant challenges."* [19]

Both Connolly [28] et al. and Boyle et al. [19] use the term COTS (commercial-off-the-shelf) to refer to the entertainment games that have been studied in some educational setting. It is worth pointing out that these differ from the category of games introduced next in Section 3.4 in two major ways. Firstly, COTS generally do not focus on any particular curriculum subject or learning outcomes specific to a subject. Secondly, they are studied in the context of formal education. An example this type of research that was included in the later review [19] comes from Ventura et al. [126]. They looked at the relationship between video gameplay and academic performance finding that a cohort playing games for a medium amount of time (which they classified 11-50 hours per week) had significantly higher grade averages compared to low playing (0-10 hours per week) cohort.

Games in Computer Science Education

In the context of computer science education, Wallace et al. [128] identified four different approaches to using games. The first two approaches involve either (1) programming a full game or (2) implementing a critical part of a game. The third approach involves (3) programming an agent to interact with the game. Finally, the last way to facilitate learning CS with games is to (4) play a game that has been designed to teach particular concepts.

The categorization from Wallace et al. [128] can be simplified to two broad approaches. In the first approach, the game is used as a context for learning – by implementing a part or a full game (categories 1-3). While the other approach is to play a serious game and learn through that (category 4).

Though perhaps not immediately obvious, using games as a context in a CS1 course can create a more authentic environment to learn. This approach was investigated by Bayliss & Strout [15] to use games as a "flavor" for a CS1 course. They measured student pass rates on the exam (no significant differences from last year) and the level of student comfort and anxiety. They noted a considerable drop in the number of students who were intimidated by the knowledge of their peers as compared with the previous year. However, they note that the drop in anxiety might also be due to the fact that the course was taught at a distance. They conclude that students generally liked the game-themed approach.

The other way of utilizing games in CSE, serious games or games for learning, has also been studied in the context of formal education and there are a plethora of games that cover different topics typically included in CS1 curricula [125]. Approaches using serious games in CSE typically aim to improve engagement and motivation while also supporting learning, e.g. [96, 71, 14]. Similarly, the types of games that have been used vary. On one end, there are block-based programming puzzle-based games, such as commercially available Lightbot (`http://lightbot.com/`), or more games developed based on the same idea that has been aimed at a more specific context and evaluated through research such as Program Your Robot [71]. At the other end there are more open-ended real-time strategy games, where the gameplay loop involves more control and planning such as in Prog&Play [96].

Alternate Reality Games in Computer Science Education

One novel approach utilizing a game with a purpose of learning and increasing interest in CS that has been recently tested is the use of Alternate Reality Games (ARG). Alternate reality games operate in the real world purporting to replace or modify reality with an alternative one. They use various normal channels of communication such as blogs, emails, website, Twitter accounts etc. to distribute and disseminate information relevant to the game. A typical feature of ARGs is that they do not claim to be a game, this is referred to as the 'this is not a game' aesthetic [130].

Hakulinen [54] organized and investigated a case study for utilizing ARGs in computer science education and increasing interest in computing. He conducted ten-week long game featuring puzzles of varying difficulty levels that were revealed to the players. Solving these puzzles required skills and knowledge of various topics in computer science such as converting binary numbers to ASCII characters or finding a solution to the traveling salesman problem. Based on solutions to the puzzles that were submitted, as well as observing discussions that participants had on a forum, he concluded that ARGs can be used to teach computer science concepts. Furthermore, they can be a way to expose people that are not presently studying or working CS to computer science content.

3.4 Entertainment Games with Learnable Content

Games, both for entertainment and/or for learning, have been used and studied in the context of formal computer science education. What has not received research interest so far is the potential that games have to engage, inspire and teach computer science concepts in informal learning contexts. By its very nature, studying informal learning is a difficult task. And as was outlined in Section 2.1, the focus of research in informal learning has been on adult education and workplace learning.

This section outlines a new type of game-related learning in computer science and programming concepts through games that have been designed for entertainment. A key aspect of these games is that people not only play them, and thus potentially learn, but also watch other people play and participate in online communities due to the enjoyment they get from the games. The phenomenon of watching other people play games has risen in popularity in gaming communities over the past decade. The online communities and watching other people play games (and program) is covered in more detail in Chapter 4. The focus of this section is to provide examples of learnable content in entertainment games with emphasis on computer science concepts.

Games, especially more complex ones available today, are systems that the player tries to learn and master. Papert has commented on the relationship between games and learning as:

"The crux of what I want to say is that game designers have a better take on the nature of learning than curriculum designers. They have to. Their livelihoods depend on millions of people being prepared to undertake the serious amount of learning needed to master a complex game." [101]

If these games contain learnable content that is applicable in the real-world then it stands to reason that players might then learn skills or knowledge that transcends the game that has been the focus of their learning.

For example, strategy games might be set in real-world locations and thus teach players about geographies, countries, or municipalities in actual locations. Furthermore, these features might be relevant for gameplay, thus incentivising the player to learn and master these things in order to advance in the game. Again, this does not mean that the game is designed with educational outcomes in mind but rather than a compelling game was designed that used a real world setting.

A concrete example of learning incidentally from games are historical grand strategy games, such as Europa Universalis IV [119] that is set in Europe between 15th and 19th century. Even though it does not aim for complete historical accuracy, it is inspired and involves historical events. During the gameplay, if certain conditions are met events can occur. For example, if playing as an English ruler without a legal heir to the throne in the 15th century an event referring to the War of Roses triggers, as shown in Figure 3.4.

Figure 3.4. War of the Roses event in Europa Universalis IV [119]

The previous examples show a variety of learning content that is present in entertainment games. However, the relevant domain for this dissertation is computer science content and how entertainment games relate to CS. We have devised a categorization of three distinct different ways in which entertainment games and computer science content are intertwined. We call these categories of interaction: *integral*, *integrated*, and *external*.

Integral CS content games are those in which successfully playing the game requires learning some computer science content. However, these are not serious games due to the fact that they are primarily meant for entertainment and thus do not aim to include specific topics to satisfy curriculum requirements. In fact, they might omit important topics covered typically CS1 courses. Nonetheless, they do contain CS content in a way that is meaningful to the game. Perhaps the best-known example is Lightbot [82], where the player is guiding a robot through various levels using simple block-based commands. However, there are games which feature richer and more complex environments that contain programming.

Shenzhen I/O [134] is an example of this type of game, shown in Figure 3.5. In the game, the player is tasked, through a narrative, to design various devices using both digital logic and components as well as code in an assembly language. In essence, the player progresses through various levels by combining physical parts with code. The game tests and verifies that the player-built devices function as they should and gives feedback to the player.

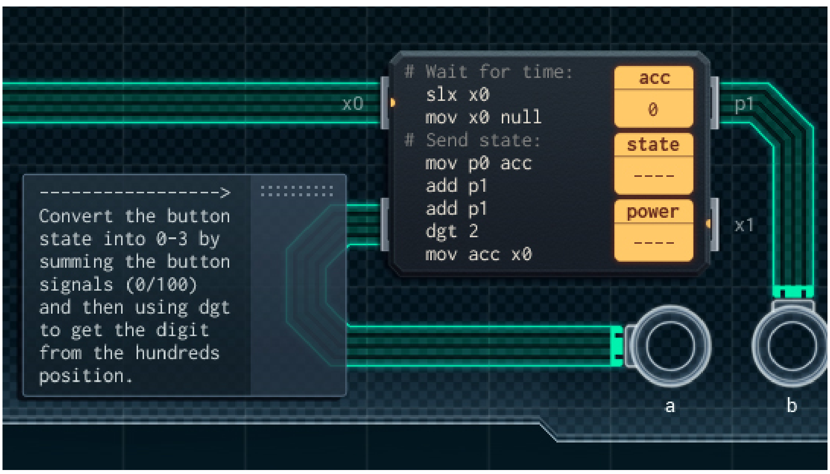

Figure 3.5. Screenshot from Shenzhen I/O by Zachtronics showing a few components and a code snippet

Integrated CS content encompasses games which include a way of programming something in the game, typically to automate tasks. What makes this category different from the previous is that while the programming is tied to the core gameplay loop it is *not necessary* in order to proceed in the game.

In practice, this typically means that the game offers an application programming interface (API) to control undertakings within the game world. These might be included in the game, for example, the popular online role-playing game World of Warcraft has a macro system that can be used to combine and customize abilities within the game (see Listing 3.1).

In other cases, the game might not support programming or macros on its own, but support can be added to through modifications or mods. An example of this type of game is Kerbal Space Program [117] in which the player designs and flies spaceships and rockets to explore the planetary system in the game. A mod called kOS (`https://ksp-kos.github.io/KOS/`) has been developed which adds a custom programming language to the game that enables players to control the rockets programmatically, as can be seen in Figure 3.6.

Listing 3.1. An example of a World of Warcraft macro that will teleport the player to different locations depending on which modifier keys are currently being pressed. Example taken from `https://us.battle.net/forums/en/wow/topic/16200990425`

```
#showtooltip
/use [mod:altshift]Teleport:Vale of Eternal Blossoms;
[mod:ctrl]Teleport:Stormwind;
[mod:alt]Garrison Hearthstone;
```

A slightly different form of integrating CS content into games is to use them as a simulation environment. A number of games include basic digital logic gates, such as AND, OR, and NOT. From these the construction of calculators and even fully functioning CPUs is possible. Perhaps the most widespread examples of this come from the hugely popular building game Minecraft [94]. Occasionally these CPUs are featured in popular media (e.g. [124]), but a myriad of examples can be found on YouTube, such as a technical explanation of how an Atari 2600 Emulator was built on Minecraft[1]. Naturally, implementing CPUs is not the only way to cre-

[1] `https://www.youtube.com/watch?v=jPRkjNDmTlc`

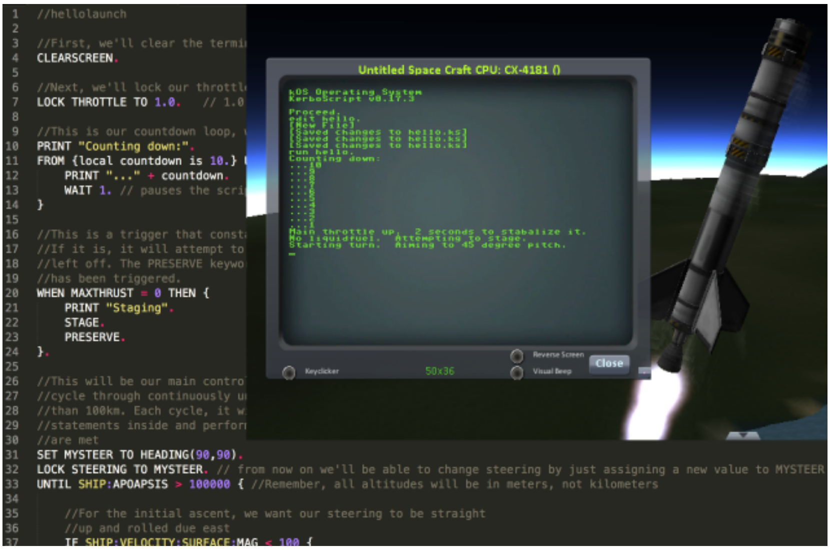

Figure 3.6. Kerbal Space Program running with kOS mod with the code shown in the background. Example code taken from `https://ksp-kos.github.io/KOS_DOC/tutorials/quickstart.html`

ate CS related artifacts, as was the case in one of the videos investigated in **PUB I** where an interpreter for BASIC programming language was built inside Minecraft. It should be noted that not all CS concepts in Minecraft are so involved and some are just part of the play and a form computational thinking, e.g. wiring blocks containing logic to enable more advanced building [122].

External CS content can be summarized by being related to the game but separated from it. These are programs designed to augment or improve the gameplay in some way without directly interacting with the game. An example of this type approach would be SimulationCraft (`http://www.simulationcraft.org/`) which is a project to *"to explore combat mechanics in the popular MMO RPG World of Warcraft. It is a multi-player event-driven simulator written in C++ that models raid damage"*. It simulates multiple people playing characters in the game in a complex environment with many different variables (such as what type of equipment the character has). The goal of the simulation is to discover optimal equipment and actions for the characters to use in the actual game.

Another example, aimed at improving the player's life, comes from Factorio (`https://www.factorio.com/`) which is a base building and logistics

Figure 3.7. Finding a way to connect pumpjacks in Factorio, arrows denote the places where pipes need to connect.

management game. Oil, a resource in the game, appears in groups small patches which need to be interconnected with a pipe (see Figure 3.7. One player designed a Python script which calculates a way to connect all the patches with pipes by using the A* algorithm to solve the distances between patches and then calculates a minimum spanning tree to figure out the connections – activities which would not be out of place in a data structures and algorithms course! This example can be found online[2] and similar examples of CS content in entertainment games are further discussed in Chapter 4.

[2] https://www.reddit.com/r/factorio/comments/6all0k/after_those_blueprintwizardryposts_i_decided_to/

3.5 Summary of Game-Related Learning in Computer Science Education

Even though defining a 'game' is not as straightforward as it might seem, some features attributed to games are of particular interest when it comes to combining games and education. Chief amongst these are the concepts of voluntary participation and the idea of meaningful decisions.

This section introduced how games and learning programming, as well as, computer science has been combined. These approaches vary from utilizing game mechanics and elements in non-game contexts, known as gamification, discussed in Section 3.2 and that is the context of **PUB V PUB VI**, and **PUB VII**. The approaches to combining CS and games have varied from serious games in formal contexts [15] and informal contexts [50]. We are presently not aware of any prior research investigating gamification of CS education in informal contexts, which is the topic of **PUB I**, **PUB II**, and **PUB III**.

Using games designed for education, also called serious games, was introduced in 3.3. Varying approaches to serious games in the computer science education have been researched, ranging from alternate reality games to encourage learning CS [54] concepts to using games as a context for CS courses [15].

Finally, in Section 3.4, categorization of games that are relevant for learning computer science concepts are introduced. The games can be organized as having CS content being *integral*, *integrated*, or *external* to the gameplay. Research in the phenomenon of people, perhaps, learning CS through games not meant for that purpose has been so far non-existent. Whether or not learning happens is hard to verify. However, through some online gaming communities participants are *exposed* to computer science concepts and programming. A related phenomenon to the games meant for entertainment is one of people watching other people play these games which will be discussed in more detail in the next chapter.

4. Games and Exposure to Computer Science

The culture around playing games has changed drastically in the past few decades. Whereas once games were played mostly at home either alone or with friends gathered around the same monitor or television, games are increasingly being played online. In addition to playing games online, gaming is broadcasted allowing anyone interested with an internet connection to tune in and watch it.

This chapter begins with Section 4.1 outlining different ways in which playing games are watched: *esports*[1], *let's plays* (a narrated recording of one's playing), and *live-streaming* (similar to let's plays, except the audience is watching it in real time as it happens). The section also outlines how these online gaming communities are relevant for computer science education.

Section 4.2 focuses on research related to learning and exposure to computer science through investigating two different types of online gaming communities. Firstly, **PUB I** investigated the types of gaming videos that relate to CS on the YouTube platform. From the found videos,two were selected for closer analysis by examining the discussions happening in the comment sections. Secondly, **PUB II** looked into live-streaming and computer science focusing on the discussions that happened during live-streams related to game programming. Finally, the data sets from **PUB I** and **PUB II** were combined and analyzed together in **PUB III**.

The third publication relevant for this chapter, **PUB IV**, is discussed in Section 4.3. It looks into the histories of CS major students who reflected

[1]Various different spellings of 'esports' have been suggested, for example, e-sports and eSports. In this dissertation, the spelling that is used is esports which is aligned with other similar terms, such as email.

their reasons for choosing this major. The research focuses on building a categorization of the types of game-related experiences students had in the past. In addition to this, part of the analysis focuses on how students view game development as a career option.

This chapter concludes in Section 4.4 by looking at the different research in online communities conducted in **PUB I**, **PUB II**, and **PUB III** as a whole.

4.1 People Watching Other People Play

Playing games for entertainment is hugely popular and an important part of the current cultural landscape. Recent numbers from the Entertainment Software Association (ESA) show that in the United States alone 67 % of households own a device that is used for playing games [43]. Another aspect worth noting from the ESA study is that the average gamer is 35 years old, so gaming is not just a children's pastime.

Alongside the popularity of playing games, watching other people play has also grown in popularity. As early as 2012, Kaytoue et al. [70] suggested that these live-streamers and those who watch them form new types of online communities.

While not the only form of watching people play games online, esports – watching people play video games competitively – is perhaps the one that is more easily understood for someone who is not familiar with the current gaming trends. After all, watching regular sports is a familiar pastime for many and also an industry on its own. The popularity of esports has risen in the past two decades whether measured in prize money [37] or in viewers [48]. Already in 2013, Witkowski et al. noted that *"e-sports tournaments have spectator numbers in the millions, recent franchise games have logged over a billion hours of gameplay, while experts and amateur e-sports enthusiasts alike regularly broadcast and share their competitive play online"* [131].

Due to the popularity of watching people playing games, it has also created more business opportunities. As an example, Amazon acquired a popular live streaming platform Twitch.tv for $ 970 million in 2014 and at that time the platform had more than 40 percent of live streaming video traffic [74]. Earlier in 2014 Twitch.tv had hosted a game project 'Twitch plays Pokémon' in which 1.6 million participated playing a game collaboratively. The viewers had the option to type out commands (e.g. 'left', 'up') to the chat which were relayed to the game. In addition to the participants, the event also gathered 55 million viewers [135].

Broadly speaking, there are three categories of watching other people play digital games: esports, let's plays, and live-streaming. Often these types of watching people play are talked separately, but there are similarities between them and the line between let's plays and live-streaming is not well defined. While there are multiple platforms online for consuming and creating different types of digital content, Twitch.tv is the most popular one when it comes to live-streams with gaming.

Esports

Esports, sometimes also called e-sports or eSports, short for electronic sports refers to competitively playing video games, typically for prize money. Similar to other sports, esports have their own leagues for different games but instead of teams based on nationality or particular geographic location, the teams are arranged into organizations.

As an example, the popular competitive battle arena game Dota 2 has an annual tournament where the prize money has been increasing every year. In 2017, 'The International' tournament held its largest prize pool yet of over $24 million with $10 million going to the winning team [37]. During the final, more than 400,000 people watched the event live on Twitch.tv [102].

However, the games played within esports are not relevant when it comes to learning CS concepts or being exposed to computer science in general. Esports is merely introduced here to give a broader view to the phenomenon of watching other people play. Though esports is not relevant for learning computer science, people do watch streams in order to learn aspects of a particular game, and for entertainment more broadly [59].

Let's plays and YouTube

Where esports focuses on competition and watching professional players compete against each other, *let's plays* focus on watching someone play a particular game and narrate the events. Typically these let's plays focus on the experience of the one who is playing and they are uploaded to popular video sharing services, such as YouTube, as a series of episodes.

Which games get played depends on the content creator's preferences. Typically, different creators focus on different types of games or a particular form of let's plays. These videos are relevant for computer science education since they can feature games with computer science content. Furthermore, sometimes the content creator has a background in CS and explicitly discusses the CS content within the games. One example of an author explaining CS concept through a game, is the YouTube video 'BASIC Programming Language in Minecraft' (`https://www.youtube.com/watch?v=t4e7PjRygt0`) in which the author explains how he implemented BASIC interpreter using blocks available in Minecraft. This video and analysis are discussed further in Section 4.2, as well as, in **PUB I**.

YouTube has been noted as an online platform for learning where user-generated content plays an important role [127]. Similarly to viewing live-streams having participatory communities, in 2010 Chau [25] looked at YouTube and wrote about it as *participatory culture*. Participatory culture consists of five features: low barrier to expression and civic engagement, support for creating and sharing project, informal mentorship, belief that contributions matter, and a sense of social connection. Chau looks at YouTube through these different features and concludes that it truly is participatory culture.

Two of these five aspects are particularly relevant to the topic at hand: the low barrier to expression, as well as, informal mentorship. Chau looks at expression and civic engagement through the ideas of communities of practice and legitimate peripheral participation [81] (see also, Section 2.2). Creating videos and leaving comments to videos is central to the community whereas watching, liking, or sharing video is more akin to peripheral participation. Related to this, the informal mentorship is an important part of the community as well. Videos teaching various topics are popular, and Chau showcases examples where YouTube creators make videos that teach topics related to video creation, such as how to edit videos or produce a particular type of video, and thus showing new members how to join to the community of video creators.

Videos posted on YouTube have been studied in the formal context of education. For example, Carlisle [24] created YouTube videos to partly replace traditional lectures on their CS1 course. They found that students preferred the videos, as well as the shortened lectures that accompanied the videos. Importantly, they found that students performed at least as well after the videos as they did prior to watching the video lectures. Interestingly, they conclude that even though the videos were created for a particular course and an audience, 13-17-year olds also viewed the lectures. They further speculate that video lectures might be incorporated into an outreach program for their institution.

In the informal context of learning, the communal aspect and creating instructional videos on YouTube is also present in computer science and programming content. Though the impact of these videos in informal learning has not been studied, the videos themselves have been studied. For example, Poché et al. [104] analyzed 6000 comments taken from 12 programming tutorial videos from YouTube. Their main aim was to create a system to categorize and summarize the comments to aid the content creators in improving their videos. However, it is interesting to note that they found that out of all the comments approximately 30 % contained useful information for the video author, showcasing the participatory nature of YouTube.

In physics education, Mohanty & Cantu [93] explored how to use commercial games, built for entertainment not for education, to teach introductory physics concepts. They note that the games they discuss do have game-play videos available on YouTube, though they do not focus on this aspect. However, they do discuss anecdotal evidence that a student considered the concepts of Newtonian mechanics outside the classroom and in another game. While discussing this, they note that: *"games allow this kind of self-motivated learning to continue outside the classroom."* [93]

Researching online videos for educational purposes is not limited to just YouTube. The surge in popularity for Massive Open Online Courses (MOOCs) has led to increasing interest in analyzing the effects of utilizing different type of educational videos. For example, Guo et al. looked at the viewing habits of 6.9 million video watching sessions from MOOCs held on the edX platform and concluded lectures recorded specifically for online usage were more engaging compared to using video recordings from traditional lectures[52]. However, those who watch these videos have a clear intention of learning and they belong to formal learning context.

Live-streaming and Twitch.tv

Live-streaming, as the name implies, refers to streaming video of playing a game to a live audience. In addition to the actual gameplay, emphasis is also placed on interacting with the audience. Typically, this is helped by the fact that the content creator has an additional webcam feed showing her face thus allowing for more nuanced reactions to situations in the game as they occur. Though it is possible to record and watch these live-streams as video-on-demand (vod), the immediacy of audience participation is particularly important. The audience not only has an opportunity to interact with the streamer but also with each other.

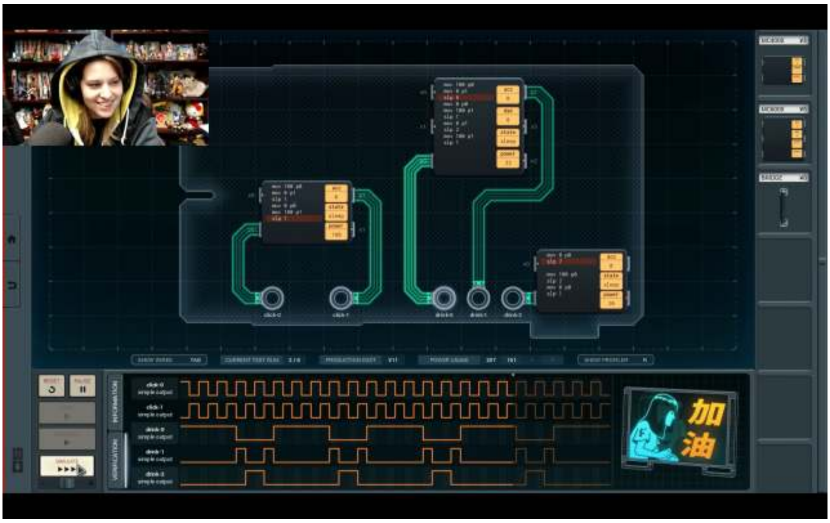

Figure 4.1. A streamer playing Shenzhen I/O

Figure 4.1 shows a typical setup for live-streaming (as well as many let's plays). The streamer has a webcam in a corner of the stream to show his or her reactions and connect with the audience while the game itself takes up the majority of the space. An important aspect of live-streams is the discussion that happens during the stream with the audience, an example of this is shown in Figure 4.2.

Originally Twitch.tv was solely focused on streaming gaming-related content, but in 2015 they changed policies and allowed what they called the Creative community to start streaming their stating that they *"encourage you to broadcast your creative process on Twitch, be that visual art, woodworking, costume creation, prop building, music composition, or*

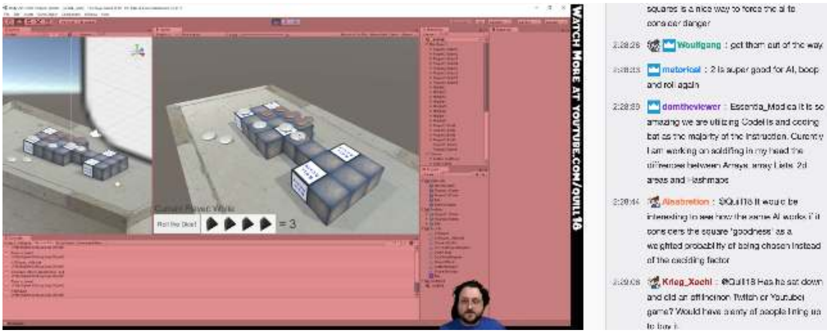

Figure 4.2. A streamer programming in Unity with the chat displayed on the right

any other process in which you entertain and connect around a creative activity." [95]. Programming is included in Twitch.tv's creative community and thus people have been able to broadcast the process of programming, as seen in Figure 4.2.

In a recent study, Sjöblom & Hamari [115] approached the reasons and motivations of the viewers of live-streams featuring games using *uses and gratifications* framework. Specifically, the motivations they looked into were: cognitive, affective, personal integrative, social integrative, and tension release. Of these motivations, they found that tension release, social integrative, and affective motivations were positively associated with the number of hours watching live-streams.

Interestingly, they also found that information seeking was also associated with more hours watched, though the effect was not as pronounced. Sjöblom & Hamari speculated that this might be due to viewers watching the streams for other motivations but end up receiving *"cognitive gratification as a by-product"* [115]. They go on to speculate that while YouTube and similar recordings of gameplay allow pausing and studying the videos in detail, live-streaming affords more of a teacher-student relationship due to two-way communication that is possible during live-streams. This type of effect was also noticed in **PUB III** where some of the viewers saw the streamer as a mentor.

Hamilton et al. [59] conducted an ethnographic study on streaming games on Twitch.tv. They see two main reasons for interest in a particular stream: the *content* that the stream offers and participating in the streams *community*. They view the streams as *participatory communities*. These communities can be reflected through the lens of communities of interest, as described in Section 2.2. However, Hamilton et al. reflect the communities and gatherings of people during streams through the idea of a *third place*. Oldenburg describes third places as *"public places that host the regular, voluntary, informal, and happily anticipated gatherings of individuals beyond the realms of home and work"* [99]. Later on, the concept of third place has been extended to online communities [108].

In their study, Hamilton et al. [59] placed emphasis on exploring the *regulars* in the streaming communities. Regulars, as they were seen by Oldenburg, are the ones *"whose mood and manner provide the infectious and contagious style of interaction and whose acceptance of new faces is crucial"* [99]. Hamilton et al. see the role of regulars as crucial in creating, alongside with the streamer, the community.

4.2 Computer Science in Online Gaming Communities

Though the phenomenon of watching other people play games is relatively new, some research looking into the motivation of the viewers of these livestreams exist. Though it should be noted that this research is often still preliminary, as well as, looking at the phenomenon from a general point of view – instead of looking into specific areas such as exposure to computer science.

The datasets used in **PUB I**, **PUB II**, and **PUB III** are described briefly in Table 4.1. The abbreviations for the datasets are used throughout Sections 4.2.1, 4.2.2, and 4.2.3. Different quotes from these datasets are identified by user/dataset marking, e.g. user1/V1.

4.2.1 Programming in Gaming-related Videos on YouTube

This section summarizes work in **PUB I** that relates gaming videos on YouTube to computer science content. To investigate the connection data was gathered with two goals: (1) find out what types of videos there are that relate games to computer science (2) what type of discussions the viewers had in the comments related to the videos.

In early 2016, a preliminary set of 50 videos were collected to provide a basis for classification for the different types of gaming videos on YouTube

Table 4.1. Datasets from **PUB I**, **PUB II**, and **PUB III** with the abbreviations for them used in this chapter

Abbreviation	Description
V1	Comments from a video describing an implementation of a BASIC interpreter inside Minecraft
V2	Comments from video discussing using a customized programming language inside a game to control rockets
S1	Discussions during a live-stream showcasing the programming a video game during a game programming contest

that also relate to computer science in some way. Through inductive content analysis based on the titles and contents of the videos we derived the following set of categories from this initial group of videos:

- **Games Enhanced by Programming** Videos featuring games that contain CS
- **Game Programming Tutorial** Tutorial videos teaching aspects of game programming
- **Modding Tutorials** Videos showing how to modify (mod) a particular game
- **Game Programming Discussion** Videos discussing on a general level what the work of a game programmer entails
- **Other** The small number of videos that did not fit the above categories

For this categorization, we selected videos by querying `https://gaming.youtube.com` with the search term ‘programming’ and selecting the first 400 results[2]. YouTube’s gaming-oriented site is essentially a dedicated video interface that also contains some extra information (e.g. the games featured in the videos are shown separately in the interface, facilitating

[2]To make to categorization practical we limited the number of videos to 400.

searching similar content). It also limits the search queries to gaming-related videos.

The videos were coded into the categories separate by the authors of **PUB I**. The individual coding reached an agreement of Cohen's $\kappa = 0.619$. The conflicts in codings were resolved through discussion to reach an agreement. Table 4.2 shows the final categorization of videos, and as can be seen, the game programming videos were the most numerous with more than half of the videos falling into that category. About 10 % of the videos belonged to the category of gameplay enhanced by programming which were then further studied by looking at the content of selected videos, as well as, the discussions happening in the comment section.

Videos in the *other* category were not directly related to programming and were thus excluded from further analysis. They covered such topics as configuring a flight stick (e.g. *"Programming The Saitek X55 Slider"*) and sometimes were not even directly related to games (e.g. *"Microsoft Surface Pro 3 - The Nerdiest Review"*.

Category	#	%
Games enhanced by Programming	38	10
Game programming tutorial	242	61
Modding tutorials	7	2
Game programming discussion	56	14
Other	57	14

Table 4.2. The distribution of the videos (N=400) from **PUB I**

Two different type of videos were selected for investigating the discussions in the comments. The first video[3] (V1) selected features the author describing how he implemented BASIC interpreter inside Minecraft, as seen in Figure 4.3. He explains how the interpreter works, and he also showcases how this language can be used to automate tasks inside the game.

The second video[4] (V2), shown in Figure 4.4, features Kerbal Space Program (KSP) – a game of building rockets and spaceships and then controlling them in missions around the solar system. The video focuses on an add-on to the game, called kOS, which enables the player to control the

[3] https://www.youtube.com/watch?v=t4e7PjRygt0
[4] https://www.youtube.com/watch?v=FPDPzsnlHOI

Figure 4.3. Frame from V1 showing a code snippet inside Minecraft

vehicles in the game through a programming language created for that purpose. The author demonstrates a script and explains its operation and how it affects the rocket. He also goes on to discuss the automation in a broader context and the similarities to programming real-life rockets and rovers.

The first (by the default sorting of YouTube) 350 comments from both videos were retrieved through a script. The first step to analyzing the discussion was to trim it down to the comments that had some relevance to computer science. That is to say, comments that were irrelevant to the video, jokes, spam, and such were removed. This meant that there were 139 comments for V1 and 73 comments for V2 remaining.

We (the authors of **PUB I**) started open coding the remaining comments. Open coding is the first step in Grounded Theory (GT) – a qualitative research method that has also been applied in computer science education research [75]. The core idea is to build theories that are grounded in the

Figure 4.4. Frame from V2 showing the space ship in KSP with the code on the left

qualitative data. The second stage of GT consists of axial coding in which the initial open codes are abstracted into categories and themes.

Based on the open codes, the following themes emerged in the axial coding: Programming Languages, Efficiency, and Learning Experience. *Programming Languages* were discussed mainly in comments for V2 since it featured a custom language. Overall the topic of authenticity was discussed regarding the languages. *Efficiency* was a topic visited in both videos. In V2 it was primarily discussed in regard to some parameters of the code which could be optimized. V1 featured a short script to find prime numbers, which could also be optimized as was pointed out in many comments.

The final category, *learning experience*, consisted of comments describing the experience of learning programming. Some of the comments were more general, such as *"To be honest, logical-block-style programming is* *taught very very early in some places now. Because it teaches logic without needing to learn a programming language syntax." (user1/V2)*. While other discussion recounted more personal experiences: *"the main reason it was hard for me to get into programming was because the IDE was made for more experienced people and was packed with features" (user2/V1)*.

4.2.2 Discussions During a Game Programming Live-stream Event

Live-streaming and the online gaming communities related to that were the focus of **PUB II**. To understand the phenomenon of live-streaming and how it relates to computer science education, chat-logs were collected and analyzed during a competitive game programming event from a single streamer participating in it.

Ludum Dare is an event in which *"developers create games from scratch in a weekend based on a theme suggested by the community"* [84]. During the 37th Ludum Dare, held December 9th–11th, 2016, one stream and the chat channel related to it was recorded. The streamer had three distinct streams during the weekend lasting 6, 12, and 14 hours. The streams reached respectively 3600, 1700, and 1100 viewers. During the streams, the community typed about 39,000 comments to the chat.

It is worth noting that the content creator typically streams regular games and gameplay instead of streaming programming or game development. Thus he has an audience that is accustomed to seeing gaming content, though at least in some cases they do turn up for the programming streams as well.

Since the number of comments during the weekend was so large, the dataset was reduced in size by filtering the comments with relevant keywords, such as names of programming languages, concepts, and constructs. Similarly to **PUB I**, the relevant comments were first open coded and from those, the emergent themes were abstracted during axial coding. The final categories of the comments were: Interest in Programming, Questions & Answers, Interaction With Streamer, and General Computer Science Discussion. In the coming sections, these comments are marked with S1.

Questions and answers featured prominently in the CS related discussions. The questions ranged from very broad ones, such as asking tips on participating to Ludum dare events, to very specific ones, inquiring about particular features of a particular programming language. A sense of community is evident in all of the categories. However, it is particularly clear in the questions and answers because some of the questions were directly aimed at the streamer but they were also answered by other members of the community.

Interaction with the streamer is what really sets streaming apart from recordings on YouTube and similar platforms. This interaction is not just limited to questions but it can, for example, be appreciative of the streamers work and recognizing their role in a learning process: *"Hey [streamer], I really enjoyed watching your hell wars game from ludum dare 33, I learned a lot about c# delegates through it." (user3/S1)*. These interactions also make the streams more collaborative and the viewers can and do contribute to the process: *"try removing your clamp code that moves the model downward" (user4/S1)*.

Finally, there was general *computer science discussion* which did not necessarily relate to the current stream in any way. Participants discussed features in different programming languages. Similarly to **PUB I**, there were also discussions regarding the experience of learning to program: *"yep python was the first language i learned and is a great one to start with" (user5/S1)*.

PUB II also reflects on the potential uses of live-streaming in more formal education. Three distinct approaches to utilizing streams can be seen:

Outreach – Streams are in a unique position to showcase what the work of a programmer or a developer might look like to an audience that might not encounter it otherwise. Potentially, this can work as an outreach program to expose people to computer science. There were also indications of this happening with the discussions the viewers had during the streams.

Teaching Method – With the popularity of various programming MOOCs (Massive Open Online Course), streaming could be an additional teaching method. The live interaction allows for questions that directly relate to the code being currently worked on. Importantly, other participants can also answer questions as well – something which is not possible during traditional lectures.

Extra-curricular material – Finally, live-streams and recordings of live-streams could be used as an additional resource for courses. Many streams are stored and published as VODs (video on demand) directly on Twitch.tv or later on YouTube. These allow the viewer to see a longer and more complex programs being created. Additionally, the VODs might not be relevant to interesting to all students but they could be a valuable source of learning for those interested in a topic that is featured on that particular stream.

4.2.3 Combined Analysis of Comment Data from YouTube and Twitch.tv

In order to look at the phenomenon of online gaming communities and learning computer science concepts, **PUB III** looks at the combined datasets from **PUB I** and **PUB II**. The datasets in this publication are referred to with the same identifiers as in the previous sections. Namely, V1 and V2 refer to the videos originally investigated in **PUB I** and S1 to the livestream analyzed in **PUB II**.

Doing a combined analysis using both datasets improves the categorization of different types of discussions that relate to CS that are happening in these communities. Since the context of the conversations and the communities in which they happened were fairly different, our aim was to produce a categorization that would apply to many different types of online gaming communities.

The filtered list of comments regarding V1 and V2 were kept the same as in **PUB I**. However, for S1 we redid the keyword filtering of the chat comments to include as many comments relevant to computer science as possible. The filtering of the original 38,694 comments to the chat resulted in 2079 user interactions. These were then read through by both authors of **PUB III** to ensure the relevance of the comments. This process reduced the dataset to 365 comments that were then used with the relevant comments from V1 and V2 to start the open coding. In the axial coding, three major themes emerged from the datasets: Learning, Programming Experience, and Community. Similar comments were grouped into subcategories within the themes.

Learning – The range of viewers with varied backgrounds to YouTube and Twitch seemed to lead to a host of discussions regarding learning computer science. Some of the comments implied a *desire to learn* CS, such as *"I keep telling myself I want to learn programming. Watching this both inspires me and terrifies me " (user6/S1)*. Though *poor self-efficacy* was not as common, it was still present in the comments, e.g. *"programming seems totally impossible to me" (user7/S1)*. Due to the context, it is not surprising that *game development* was a motivation for learning programming. There were also thoughts shared on the *resources to learn*, as well as, as well as enthusiasm for programming *"im 13 and learning c# and uss unity and blender so I'm starting young but think I may be able to make it.... and even better I find coding fun XD" (user8/S1)*.

Programming Experience – Discussions within learning theme were generally had by those who were seemingly less familiar with programming. But the theme of programming experience was naturally discussed more by those with more prior CS experience. A common topic of discussion under this category was *programming languages and paradigms* where the discussion ranged from considering features of individual languages to comparing paradigms in programming. As was the case in **PUB I**, *efficiency* was discussed in S1 as well, where discussion also covered aspects of memory management. *Debugging* was a topic that was only seen in S1, probably due to the fact that the events happened live thus making audience participation easier. Additionally, code that is featured in videos is presumably more polished than one that is being written at the moment. Especially those who seemed to have an undergraduate background also discussed CS in terms of a *professional career*.

Community – The sense of community and people coming together to spend time together was definitely more noticeable in discussion happening during S1. This is understandable since the streamer has regulars in the audience – something which is not often achieved in YouTube comments. Two subcategories under this theme were particularly noticeable. Firstly, there was a strong sense of *encouragement* for newcomers to learn to program, which were seen in replies to newcomers such as *"Programming is one of the most rewarding things in my life. If you're enjoying this stream, you should try to dabble in the basics. I think you'll like that as well." (user9/S1)*.

The second noticeable topic of discussion was that of seeing the *streamer as a role model*. This was reflected in the comments and questions inquiring about his background, for example "does [streamer] have a background in computer science?" (user10/S1). Many comments also reflected appreciation towards the work that the streamer does. This included the programming live-streams, as well as, the programming tutorials he posts on YouTube: *"[streamer] Your FPS Multiplayer tutorial was the one that opened the door to Unity3d programming for me." (user11/S1).*

In addition to the themes present in the discussions, **PUB III** also outlines perspectives to online gaming communities to aid in future research. These perspectives aim at capturing the differences in online communities when it comes to computer science content. The four perspectives we identified were: engagement, interactivity, exposure, and technical detail.

Engagement and *interactivity* are essentially two sides of the same coin. When platforms afford more engagement it means that the viewers are able to contribute through *legitimate peripheral participation* (see Section 2.2). With live-streaming, this process is facilitated by interactivity that is possible due to the platform. Video sharing sites typically have lower, asynchronous interactivity in the form of leaving comments to the video while streaming services offer synchronous interactivity through real-time chats with the streamer, as well as, amongst the viewers.

Exposure is the potential reach that the platform has. In videos like V1 and V2, the games themselves are already popular and they attract more viewers due to the games featured in them and the authors creating those videos. That can be contrasted to S1 where the viewers need to know when the live-stream is happening in order to tune in.

Technical detail describes the level of detail that technical and more complicated aspects of programming and computer science that are discussed. Short ten minute videos, such as V1 and V2, which were aimed at the gaming community at large, naturally cannot go into much detail. Correspondingly, longer (which is typically the case) live-streams can go into much more detail but at the same time, they might not be as accessible to someone not already familiar with programming.

4.3 Exposure to Games and Studying Computer Science

Where **PUB I**, **PUB II**, and **PUB III** investigated online gaming communities and how exposure to computer science is connected to them, **PUB IV** explored how university students in a computer science program reflected about their experiences with games. The aim of this study was to gain insights on how exposure to games might relate to choosing their field of study.

The datasets were originally gathered for a separate research project aimed at investigating the CS students background more broadly. However, since it contained a considerable portion of details about past gaming experiences that came up unprompted, we used the datasets in **PUB IV** as well. The qualitative approach in **PUB IV** relied on two different datasets gathered from students majoring in computer science:

- **Essays** were written as an assignment on an introductory course that was mandatory for the first-year students in computer science. Students were asked to reflect on the some or all of following topics: “time before university studies”, “reasons for choosing this university”, “why you chose CS major”, “expectations regarding studies at the university”, “what interests you in their future studies”, “what does not interest you in their future studies”, “strengths as a student”, “weaknesses as a student”, and “career plans’ which had been used in similar research in similar settings [100]. There were 69 participants on the course and thus 69 essays. It is worth pointing out, that students were not explicitly prompted to write about topics related to games.

- **Interviews** provided a second data source to students’ current and past game playing. This time the participants (N=9) were from a data structures and algorithms course. The semi-structured interviews were a part of another research project related to students experiences with online learning and gamification. The participants were selected to represent different goal orientations [98].

The essays were written in Finnish and the interviews were conducted in Finnish as well. Thus the analysis was done entirely in Finnish and the quotes presented from these datasets have been translated. The analysis began with the essays, of approximately 500-750 words, which were read through. Based on the initial reading three of the categories contained themes related to games: the time before studies, reasons for choosing CS as a major, and what interests the students in future studies. Almost a third (22/69) of the essays were considered relevant, in other words, games were present in some form in the essay. These were then selected for further study and read through multiple times.

The pertinent sections were then categorized, similarly to approaches in **PUB III** and **PUB I**. Based on the essays the relevant themes emerging from the data were: *Games as interest Triggers for Computing*, *Programming (in) Games*, and *Game Development as Career*. We left the option open to expand these categories based on the interview data. However, during analysis of the interviews no new themes emerged despite the fact that all nine interviewees recounted either playing games currently or in the past.

Games as interest Triggers for Computing

Under this category, the games did not necessarily provide the impetus to learn computer science yet. However, what they did provide was the motivation to understand computers, for instance, how to turn them on without older siblings. Some students directly credited this for their interest in the field:

"... different games got me interested in computers and how they work. At the moment, I don't play nearly as much as I did in my childhood but the interested in information technology has stayed." (essay, translated)

Programming (in) Games

There were two types accounts when it came to students' past experiences regarding games and programming. On one hand, some remarked that they had the desire to learn to program in order to create games. On the other hand, games that incorporated programming, as discussed in Section 3.4, were the spark to learn to program:

"[Programming] was just a thing I encountered in a game which required a little bit of programming. Some games are like that where you can de-

velop in them a bit. Then I realized I didn't know anything about programming and it would be nice to know." (interview, translated)

Game Development as Career

When it came to future plans, working in game development was seen as an appealing option. Some saw the industry as a growing one with an increasing need for developers, while others saw it as an opportunity for creative work, for example:

"I've always been interested in the game development industry and the possibilities it affords to apply one's creativity. As such, computer science was clearly my first choice from the beginning" (essay, translated)

4.4 Conclusions

This chapter has looked into exposure and learning CS through computer games. Firstly, the concept of watching other people play and its significance in modern online culture was discussed in general. Afterward, online gaming communities and exposure to computer science were linked and the research conducted in **PUB I**, **PUB II**, and **PUB III** was discussed. Finally, exposure to games was discussed within the context of career selection and how computer science majors reflected on their own gaming past and its influence on their career selection.

Due to differences in online gaming communities and different affordances that their chosen platform offers, there are differences in what is possible when it comes to exposure to computer science. *Engagement and interactivity* are naturally much higher in cases such as S1. This is due to the community consisting of regulars, who turn up to view the streams and interact with other members of the community frequently. This can be contrasted to video platforms, such as YouTube, where the viewers are part of a much larger audience and do not necessarily interact with the content creator or other viewers to a great degree.

Similar differences apply to *exposure* and the level of *technical detail* that separate online communities can achieve. For example, the content creator in S1 usually streams playing games instead of coding them, and as such, his audience typically tunes in for that. When there are streams that involve game programming, it is then feasible that at least some of the audience shows up and is exposed to programming in a way that would typically not happen. Similarly, the level of technical detail is different, as was the case in V1 and V2. They are meant to appeal to a wider audience and are shorter, minutes instead of hours, in form thus limiting the complexity of discussion.

PUB IV related past gaming experiences to choosing CS as a major. Overall, we found three themes in students' recollections of their background with games: how games got them interested in computers, how they encountered programming in games and how they also programmed games themselves, and their interests in pursuing a career in the game industry. Overall, about a third of the freshmen students wrote about games and gaming hobbies in their essays describing their reasons for choosing CS, even though it was not explicitly asked about. When prompted during interviews all nine students had played or were still playing games.

5. Gameful Approaches in Formal Computer Science Education

This chapter looks at gamification and badge systems from two perspectives. Firstly, badge systems are viewed not as a motivation or engagement enhancing reward scheme but as a technical software system that needs to be implemented. The focus is especially on solving the problem of interoperability: gamification systems need to interact with other systems related to learning in a modern online environment. In the latter part of the chapter, we focus on the students' reactions to these systems and utilizations of badges on a course.

First, relevant concepts such as Learning Management Systems (LMS), Smart Learning content (SLC), and online learning activities are introduced in Section 5.1. The section also discusses the concept of interoperability and relates it to online learning. A brief overview of different types of learning managements systems is given.

Section 5.2 introduces two different software systems for supporting gameful approaches in computer science education. *Acos* is introduced as a interoperability platform that hosts different types of online learning activities. *Daechschen*, the other system, is more focused on implementing gamification through achievement badges to a single LMS.

After introducing the technical aspects of Daechschen, Section 5.3 discusses an experiment on adding achievement badges on a data structures and algorithms course. The feedback from students is looked at both qualitatively and quantitatively. In addition to feedback, the student behavior was also analyzed utilizing log data from Daechschen.

Finally, Section 5.4 concludes the chapter by discussing about the different software-based solutions to adding gameful approaches to CSE. Additionally, the importance of interoperability in modern online learning environments is discussed.

5.1 Learning Management Systems and Smart Learning Content

Currently computer science education uses and leverages automatically assessed exercises that are typically accessed via Learning Management Systems (LMS). Common examples of such systems are, for example, Moodle [1], Blackboard [2], or various MOOC systems such as Open edX [3].

Sophisticated exercises in these systems can be thought of as Smart Learning Content (SLC). Brusilovsky et al. [21] outlined the SLC as as having *"some form of interactivity is a central aspect"*. In order for learning content to be considered 'smart' they argue that it should have three qualities: *input*, *process*, and *output*.

Input refers to the data that the learner inputs to the SLC. It varies between pre-specified (e.g. selecting a particular option) or free-form (e.g. writing code that is run or otherwise checked by the SLC). The SLC takes the learner's input and *processes* it. The processing is also on a continuum from a fully computational (SLC computes the output completely) to not computational (e.g. the SLC just facilitates communication between two learners). Finally, in the *output* the SLC provides feedback to the learner, which ranges from generic (feedback doesn't adapt to that particular learner or context) to customized (e.g. feedback is different due to the learner's incorrect solution in two previous attempts).

However, in this dissertation the scope is broader and SLC is considered to be one category of online learning activity. In **PUB V** it was defined as *"We have used the term online learning activity to describe any learning activity that a student can work within a browser"*.

Interoperability

Though many of the technical challenges lie outside the scope of present work, the concept of interoperability is of importance. Interoperability refers to *"The ability of two or more systems or elements to exchange information and to use the information that has been exchanged"*[4]. In practice, in the ideal case, this would mean that various different exercises from different sources using various protocols could be accessed from an interoperable LMS.

Dagger et al. [29] have divided LMSs to three categories depending on their level of interoperability. In the first category systems are completely *monolithic* and the only way of extending their capabilities in terms of gamification or new SLC is to modify the system itself. The second category, to which Dagger et al. refer as *modular*, are systems that are extendable. This means that they can be extended in some way without touching the core system. In practice, this typically means that the system has a plug-in system which can extend the capabilities of the core system. It should be noted that this generally means that the system can only be extended using the same programming language it was created in.

The final category that Dagger et al. identified are the *service-oriented* systems. With these systems it is possible to incorporate SLC from various sources, utilizing one or more interoperability protocols, such as Learning Tool Interoperability (LTI) [66]. With these systems it is also possible to incorporate other types of content beyond SLC. With Interoperable systems including new elements such as visualization tools or gamification systems becomes much simpler compared to monolithic or even modular systems.

In service-oriented architectures the systems typically communicate using Representational State Transfer (REST) [44]. In REST the requests between systems use a stateless protocol, as well as, standard operations (e.g. GET, POST, DELETE in HTTP). This approach results in systems that can be expanded by incorporating new components into existing systems. Educational research benefits from this approach due to the fact that new tools and learning content can be tested with less effort than in monolithic systems.

5.2 Systems Supporting Gamification Schemes

This dissertation presents two complementary approaches to supporting gameful approaches in online learning management systems: Daechschen, presented in **PUB VI**, and Acos, described in **PUB V**. Before a more detail look into the systems, the core differences are discussed here first.

Both of the systems were designed to work with the service-oriented A+ [69] LMS. The core idea of A+ is to separate the SLC to external services which are then accessed through an Application Programming Interface (API). Additionally, Acos supports multiple protocols so SLC designed to work with Acos can be integrated into multiple different LMSs, such as Moodle [1] or systems using Adapt2-protocol [20].

A+ was originally developed at Aalto University with the goal of reducing different systems that students would have to interact with on CS courses. Previously, there were many different systems hosting different types of exercises that would have to be accessed from different online environments. To combat this A+ was specifically designed with interoperability in mind. Presently, it is used annually by approximately 3000 students on 20 different courses at Aalto University and recently it has also been used and developed other universities as well.

While both systems were designed to support gameful approaches to online learning activities their approach to gamefulness differs. Daechschen was designed to be a complementary system for A+, only focusing on implementing achievement badges to be used on courses in A+. Whereas, Acos was designed as a more interoperable system where learning activities (both with or without gameful approaches) could be incorporated into multiple different LMSs using multiple different protocols.

Other software for supporting gamification and badge systems has been developed as well. Perhaps one of the better known ones is the Open Badges [107] which was started by Mozilla Foundation. It aims to: *"provide a flexible way to recognize learning wherever it happens, in and out of formal education and the workplace."*. Institutions and organizations can award badges on this infrastructure and the recipients can showcase their badges through this system. Originally, Open Badges were considered as a platform for the research in **PUB VII**. However, the process of becoming a badge issuer is administratively cumbersome for a small research project. Furthermore, the completion logic for the badges still needs to be incorporated into a system that utilizes Open Badges. In practice, this means that a new system still needs to be implemented, such as the one described by Wüster & Ebner [133].

5.2.1 Acos

Acos is a solution to hosting online learning activities in an interoperable manner, making it easier to integrate existing learning activities to different platforms. This is achieved by decoupling communication protocols from the learning activities thus providing a uniform interface for developers of learning activities. When it comes to the design of Acos, there were four key considerations that needed to be taken into account: interoperability and reusability, extensibility, scalability, and discoverability.

The overall architecture of Acos is shown in Figure 5.1. It describes the main idea of Acos: it provides a uniform interface for all learning activities which enables using the same activities with different interoperability protocols (e.g. LTI [66] or Adapt2 [20]). The support for different protocols enables integrating learning activities into different LMS environments without modification to the learning activities themselves.

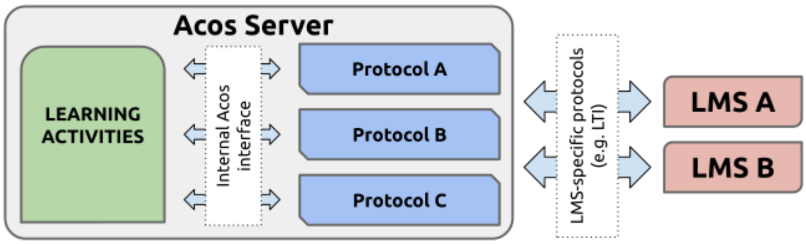

Figure 5.1. Overall Architecture of Acos

Acos has a modular architecture where support for new learning activities and protocols can be added without it affecting any of the existing functionalities. Internally, Acos has three types of components: protocols, content types, and content packages (see Figure 5.2). *Protocol* are completely separated from the content types and content packages, and thus adding a new protocol package means that all the existing content can be used through that protocol as well. *Content types* implement the basic functionality for a given learning activity (e.g. js-parsons [64]). *Content packages* implement the actual learning activities based on the functionalities offered by a specific content type (e.g. js-parsons-python[1]). For some of the content types, prototypes for gamified learning activities have been developed.

Additionally, Acos has a package type for *tools*. These are general purpose packages that implement tool-like functionality that is not directly related to a learning activity, but for example, can assist in creating one.

[1] `https://github.com/acos-server/acos-jsparsons-python`

An example of a tool, is acos-jsparsons-generator[2] which allows creating new js-parsons exercises in a browser without the need to edit existing content packages.

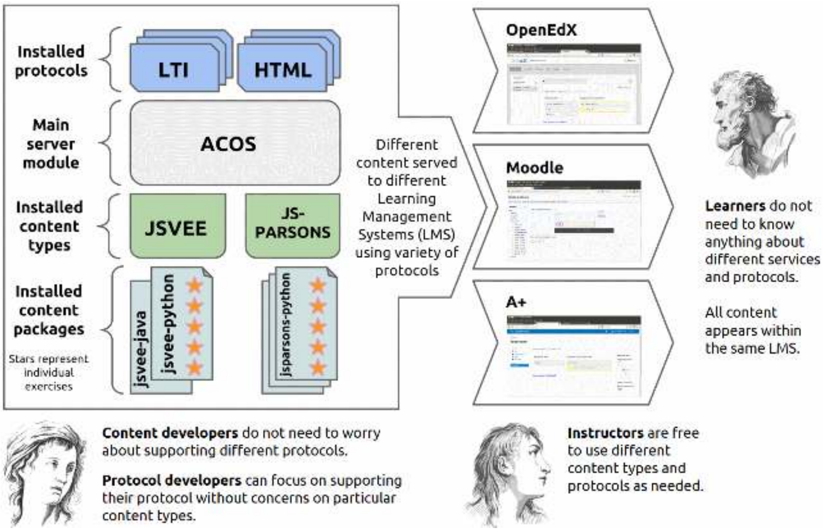

Figure 5.2. A high-level view of different software components in Acos and description of benefits of Acos for different stakeholders.

Figure 5.2 also shows different roles that people have when it comes to Acos. *Content developers* are free to focus on developing content and new content types without the need to worry about implementation details how those learning activities are incorporated into online environments. Correspondingly, *protocol developers* can focus on supporting interoperability protocols without concerns regarding content types. For *instructors*, Acos can be thought of as a repository of learning activities from which one can choose the most suitable ones on a pedagogical basis. Finally, *learners* should be completely oblivious to the existence of Acos, since the learning activities are incorporated into their usual LMS.

Evaluating Acos

Acos was evaluated against the four core design considerations: interoperability and reusability, extensibility, scalability, and discoverability.

Interoperability and reusability of learning activities is greatly enhanced by the fact that different interoperability protocols are supported and all

[2] https://github.com/acos-server/acos-jsparsons-generator

the existing content can be used with any of the protocols. Table 5.1 lists the the protocols currently supported by Acos.

Package name	**Description**
acos-aplus	A+ protocol support
acos-html	A protocol to support embedding online activities as simple HTML pages without communication to LMSs
acos-lti	LTI protocol support
acos-pitt	Support for communication with ADAPT2-protocol

Table 5.1. Interoperability protocols currently supported by Acos

Acos has been tested to work with Moodle [1], as well as, Open edX [3] through the LTI protocol. Additionally, Acos has already been used in multiple universities in two countries using different learning management systems with two different protocols.

Extensibility is mainly solved through the architecture of Acos. Modular structure means that new functionalities (such as content types, protocols, or tools) can be implemented without affecting other modules. In addition to this, Acos is written as a **node.js** application and it uses the standard solution of `npm` package manager to distribute its packages. Finally, the server itself, currently supported protocols, content types and packages, and available tools are published as open source and are available from: `https://github.com/acos-server/acos-server`.

Scalability was tested with Apache Benchmark by doing three concurrent requests to the production server until 10,000 requests had been made. 99% of the requests were served under 80 ms. The scalability follows from a stateless architecture, where nothing needs to be fetched from a datastore and, depending on the amount of learning activities, everything or most of the activities can be kept in memory. Furthermore, the stateless architecture makes it easy to install multiple servers behind a load balancer.

Discoverability is improved by having a single source for educators to browse the available learning activities. Additionally, Acos has a content brokering API, which can be used to search and browse the available content programmatically. This API could be also incorporated into an

LMS which would mean that educators could browse the available content within their LMS.

We also evaluated Acos by collecting feedback from developers and instructors (student feedback was omitted, since generally students should not be aware of existence of Acos). The feedback from collaborators has been positive, though no formal user evaluation has been conducted. For example, when it came to utilizing existing technologies as one content developer wrote in an email: *"[Acos] follows a standard architecture which rests on a popular Node.Js framework. This simplifies the content creation to a great extent."*.

There are also limitations that stem from the design principles of Acos. At the moment, all the learning activities are checked in the learner's browser, making cheating via technical means easier. However, Acos was designed to disseminate small learning activities easily rather than to support summative assessment. One severe limitation that stems mostly due to the stateless architecture, is that collaborative learning activities and working in groups is difficult to support.

5.2.2 Daechschen

Daechschen is a server side program designed to enable achievement badges in A+ learning management system. It is designed to interact with A+ through an API. When a page containing an exercise is loaded or when a learner submits a solution Daechschen fetches that learner's exercise points on the course. Then it checks whether a criterion to award a badge is fulfilled. And based on this it either awards a badge and shows it to the user or shows the badge that was last awarded.

Daechschen supports three categories of badges that aim at improving earliness of submissions, carefulness, and completion. The specific criteria can be adjusted to exercise rounds, e.g. complete X % of exercises before Y days of deadline. In addition, it supports meta-badges that are awarded when a set of other predefined badges are awarded.

Figure 5.3 shows how A+ and Daechschen communicate to show the student the learning activity and the badge information. When a student requests a page with an exercise (Step 1), A+ requests it from an exercise service and shows it to the student (Step 2). At the same time in Step 3, the badge widget is requested from Daechschen. In order to return the widget, Daechschen requests the students exercise data (completed exercises, submission counts, etc.) from A+ API (Steps 3.1 and 3.2). Based on the information received Daechschen either awards a new badge or shows the previously awarded badge and this view is returned to the student in Step 4. The summary view works in the same manner, except that there is no exercise to request and since no new badges need to be awarded Steps 3.1 and 3.2 can be omitted.

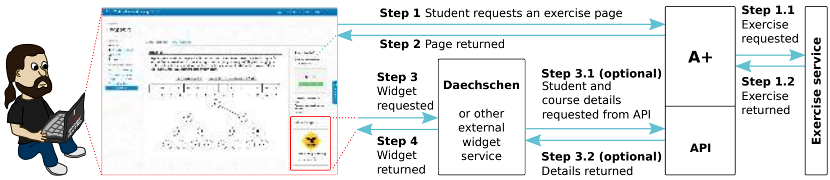

Figure 5.3. Architecture of Daechschen. From **PUB VI**

Based on the experiences of implementing Daechschen, and the earlier achievement badge system embedded in TRAKLA2 [57], we outlined six design recommendations for future badge systems in **PUB VI**. Naturally, the systems should fulfill *functional* requirements, while taking into account the *security* of the system and the *privacy* of the users. But additionally, the systems should be *interoperable* so they could be utilized from multiple different LMSs and the *usability* shouldn't be compromised even if the badge system is incorporated into different environments. Finally, the systems should be *flexible* enough to they could be adapted into multiple different courses, where the criteria for awarding badges might differ to a great extent.

5.3 Utilizing Achievement Badges on a Data Structures and Algorithms Course

To study the effects of achievement badges in an online learning management system a within subject experiment was carried on an a Data Structures and Algorithms (DSA) course at Aalto University in 2013. The course had eight exercise rounds covering topics such as sorting algorithms and graph algorithms. DSA is a bachelor level course which consists of online exercises, a larger project, and a final exam.

The course used A+ [69] LMS to host its online exercises. For this iteration of the course, achievement badges were implemented with Daechschen as an external service. Daechschen provided two views to badges: a **sidebar view** to the exercises that showed the latest achieved badge (see Figure 5.4) and **summary view** showing students progress (see Figure 5.5). The badges were enabled in the middle of the course from the beginning of exercise round five. Altogether, there were eight exercise rounds.

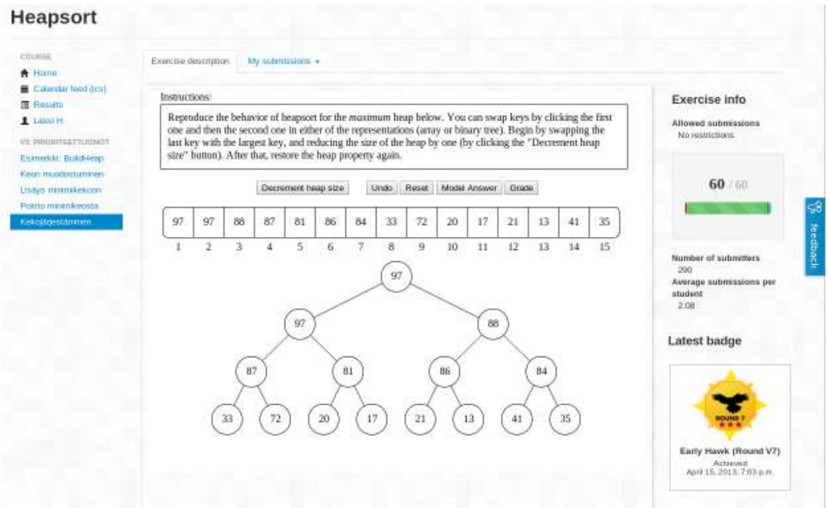

Figure 5.4. Achievement badges in A+ during the course. From **PUB VII**

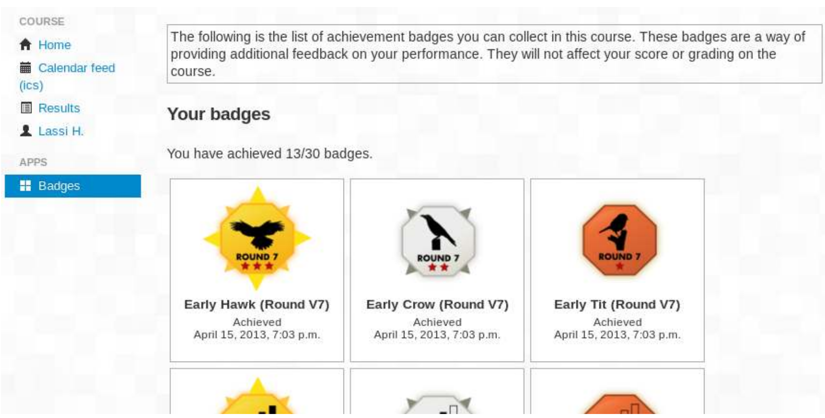

Figure 5.5. Summary view showing showing achieved badges in A+.From **PUB VII**

With three different categories of badges available there were three aspects of studying that they targeted: time management (completing exercises early), carefulness (avoiding trial-and-error approaches), and completion (achieving a percentage of maximum score from the round). The badges in each of these categories had three levels as well, bronze, silver, and gold, reflecting the difficulty of achieving the badge. For instance, in the completion category, the requirement was to complete the exercise round with 50, 75, or 100 % of the points.

There were two types of data collected for this study: log data and a questionnaire. Daechschen stored data on all of the badges achieved. In addition to that, the log data was also used to analyze how often students visited the summary page, that showed all the badges they had achieved. The students were also sent a questionnaire after the course, to which a total of 162 students responded out of the 306 students who had registered to the course. The questionnaire contained Likert-scale questions, of which six were related to badges. In addition, there was an open text question instructing to *"Please give additional comments about the badges"* to which 88 students wrote something.

Table 5.2 summarizes the the responses to the questions that related to achievement badges on the course. The scale in the questionnaire ranged from strongly disagree (0) to strongly agree (4) with 2 being a neutral option.

As can be seen from the table, similar number of students found badges somewhat motivating or not-motivating. Overall, the answers regarding motivational aspect of the badges were spread out. Only a few found the badges disturbing and the majority did not. Overall, it seems that there was a small population of students for whom badge affected their behavior, at least to a degree that they self reported so. Even though only 11 % strongly agreed that the badges affected their behavior, over half of the students agreed or strongly agreed that badges should be kept in the future installations of the course.

The badges were enabled for round 5-8 on the course, during these rounds there are fewer students who continue to submit solutions to the exercises. This is partly due to the fact that students can earn a passing grade earlier on and thus those that do not seek higher grades stop doing the exercises. We compared the effect of the badges by calculating

Table 5.2. Feedback regarding badges on the course (N=162). Most common answers are in bold. 0 = I completely disagree, 4 = I completely agree

Question	0	1	2	3	4
I found the badges motivating	20 %	**24** %	17 %	**24** %	14 %
Badges disturbed my work	**61** %	15 %	15 %	6 %	3 %
Trying to achieve badges had an effect on my behavior	**40** %	21 %	13 %	15 %	11 %
Visual look of the badges was good	3 %	6 %	22 %	**42** %	27 %
I was satisfied with the criteria for awarding badges	3 %	8 %	**42** %	29 %	18 %
I think that badges should be used in A+ for the next year's course as well.	8 %	12 %	28 %	22 %	**30** %

the amount of badges that students would have achieved on rounds 1-4 and compared that to the actual badges they achieved on round 5. There was no statistically significant (Wilcoxon signed-rank test) improvement in the number of badges earned, though we observed a smaller population of students for whom the amount of badges earned increased.

Despite the lack of statistically significant improvement in the amount of badges collected, we were able to show a correlation between interest in the badges and the number of badges achieved. Based on the log data from Daechschen, those who visited the summary view for the badges also earned more of them. Similar effect, though not in a CS context, was also noted by Denny [32].

5.4 Conclusions

This chapter introduced two different systems for supporting gameful approaches for computer science education detailed in Section 5.2. *Acos*, discussed in **PUB V**, is a modular service-oriented architecture for hosting online learning activities supporting multiple interoperability protocols. Its strength lies in the fact that learning activity developers, whether creating gamified activities or not, do not need to concern themselves with supporting multiple protocols. On the other hand, *Daechschen*, introduced in **PUB VI**, is aimed at integrating achievement badges for A+. It too has a service-oriented architecture that works with the APIs offered by A+.

Section 5.3 discussed the empirical investigation in using achievement badges in a data structures and algorithms course utilizing Daechschen. Overall, the students were somewhat positive towards the use of badges. We were unable to show statistically significant improvement in their studying behavior, although we saw a positive correlation with the number of badges achieved and the amount of times students viewed their badge collection.

Achievement badges were first tried on the course a year earlier [57] where they saw statistically significant changes in the students' behaviour. Yet, in **PUB VII** the observed changes were not significant. While some of this might be due to differences in the experimental setup, there was also difference in the badges that could be achieved, the LMS was changed between the experiments, and how badges were incorporated into the LMS. It seems, that gamification or badge systems are affected by many of the implementation details. This could also, in part, explain why there are differing results in the efficacy of using achievement badges (e.g. [5]).

6. Discussion

Using game-related approaches in computer science education is a broad topic that can be approached in many ways. One option is to use games as a context for teaching and thus, presumably, make learning CS more relevant and motivating for students. Alternatively, serious games[92] can be constructed in the hopes that just playing these games will lead to learning outcomes. Gamification [34] can also be used to try to improve motivation in educational settings by implementing various game-mechanics, such as badges, into learning environments. Finally, just by playing, or watching someone else play, video games might expose potential future students to CS.

The work conducted in this dissertation highlights two different game-related approaches in CSE. On one hand, there is the exposure to computers and computer science that happens in relation to games, and especially presently, the online gaming communities which was the focus of Chapter 4. On the other hand, game-related approaches can be used in formal education as well, which was explored via gamification and the systems to support gamification in university-level education and discussed in Chapter 5.

This chapter concludes the dissertation by first discussing the contributions made in this work in Section 6.1. The different publications are considered as a whole and conclusions are discussed. In Sections 6.2 and 6.3, the validity and trustworthiness of the findings are discussed regarding the approaches taken in the different publications. Section 6.4 considers the dissertation from an ethical point of view. Finally, in Section 6.5, potentially interesting lines of future research based on the present work.

6.1 Conclusions and Contributions

Within the informal learning context, the exposure to CS through online gaming communities was found to be interesting from an outreach point of view. While the work in the formal learning context focused on gamification, and in particular, tackling the technical challenges in implementing gamification in modern online environments. Additionally, we aimed at filling the gap in research when it comes to within-subject studies evaluating gamification in a CSE setting.

This work adds to the growing body of literature regarding game-related approaches in computer science education. The aim has been to investigate the utilization of games in CSE and exposure from multiple perspectives using different methods. The chronologically older work with gamification and systems supporting gamification was more focused on concrete systems and experiments. The contributions related to this have resulted in two open source platforms and their subsequent evaluation. The newer work regarding online gaming communities highlights the possibilities for exposure and outreach – enabled by the evolution of online gaming culture.

From a broader perspective, one of the contributions is the new way of categorizing games and learning computer science. The categorization, discussed in Chapter 3 and summarized in Figure 3.1, is not just a generic categorization of using game-related approaches in education but specifically focuses on the relationship between games and CS/CSE. It can be seen as an extension to the categorization by Wallace et al. [128] and to the taxonomy by Kelleher & Pausch [72]. An abridged version of the figure is replicated in Figure 6.1 showing how the presented categorization relates to the works of Kelleher & Paush and Wallace et al.

Wallace et al. [128] considered learning CS through games with through approaches either by programming games or by playing serious games. We broadened these approaches with gamification, watching someone else program games via live-streams, and watching as well as playing entertainment games with CS content.

Kelleher & Pausch [72] introduced their taxonomy for systems supporting novice programmers. In it they included *Empowering systems* under which they categorized *activities that are enhanced by programming* that

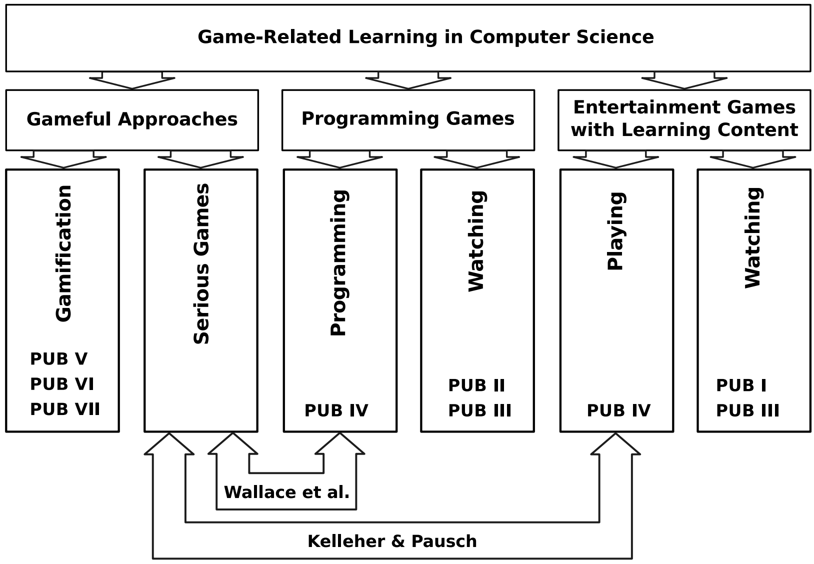

Figure 6.1. The categorization of game-related approaches with publications in this dissertation listed as well. Categories presented by earlier work of Wallace et al. [128] and by Kelleher & Paush [72] are also shown.

contains two sub-categories: entertainment and education. Within the entertainment category, they list games as an activity that can be enhanced by programming and they provide some examples of games, such as The Incredible Machine. We broadened this view by considering entertainment games that actually require some programming knowledge either as a part of the game or an optional addition. We also considered the activity of watching someone else these games as a separate category.

In addition to the categorization, one of the foremost goals of this work was to increase awareness of certain topics that have not been discussed in the CER community. Chief amongst these are the themes regarding online gaming communities and exposure to CS. Since these forms of media are relatively new, many in the research community might not be aware of them. Based on a multitude of informal conversations with colleagues and other researchers, the types of online communities described in this work and how they relate to computer science education are not yet widely known. Though there has been increased interest in researching online gaming communities in traditional game studies and broader sociological research [115, 59, 61].

Game-related approaches for CS education have been studied for decades now. Yet, as new approaches are discovered, more research is needed. Using gamification-based interventions has its challenges, both technical and motivational, but for some they provide an extra incentive to study harder. Within the informal context, there is an ever-increasing number of games available in digital marketplaces and correspondingly there are more entertainment games which provide more opportunities for CS exposure through their content, mechanics, and associated online communities.

The rest of this section is focused on relating the publications to the research questions introduced in Section 1.2. Contributions from each publication are considered under the respective research questions. The results are discussed and contrasted to prior literature. In addition, conclusions are discussed based on the work presented.

6.1.1 RQ1 How do game-related approaches and communities relate to exposure to computer science in informal settings?

The first theme of how game-related approaches and communities are related to CS was explored by looking at computer science students' past experiences with games, as well as, looking at contemporary online gaming communities and how CS is discussed in them. In the case of students' histories, they described the role of games as important when it came to becoming interested in computers and programming. In gaming communities, we found discussions between experienced programmers, but more importantly, newcomers and people who showed interest in learning to program also participated in these communities.

RQ1.1 *How do online gaming communities facilitate exposure to computer science and learning programming?*

Online gaming communities have received a great deal of research interest over the past few decades – including research related to learning [38, 123, 47]. However, it seems that the role of online gaming communities and exposure to CS has not been previously researched. As such, publications **PUB I**, **PUB II**, and **PUB III** aim at filling in this gap. The categorization of discussions regarding CS topics are the primary contributions in these articles. However, almost as important is the recognition that these gaming communities exist and people who might not necessarily seek out CS content are exposed to programming and related concepts.

PUB I was our first venture into looking at how gaming-related YouTube videos contain CS content in them. We found that there is a category of videos that mix normal (non-educational) games with CS content. Additionally, we examined the interactions that viewers had in the comment section and found that there were discussions related to CS, including discussing the experiences of learning to program.

In **PUB II** the goal was to look at streaming communities and how CS content is discussed in them. We found that particular streamers can gather audiences of thousands of viewers who watch game programming happening live during game development contests. Additionally, viewers participate in the programming process by providing feedback, suggestions, and help with debugging. Because of the active role the audience (or some members of the audience) it can be seen as *legitimate peripheral participation* [81].

Similarly, these communities can be seen as *virtual communities of interest* [60] where the audience and the streamer are gathered around a shared interest. Though it is important to note that not all those who watched had programming experience and some of the discussions were focused on those who wanted to learn to program or had just recently started. Due to the interest in learning to program, an argument could also be made that the audience also a *learners' community* [60]. This is also supported by research concluding that the number of hours watched is associated with information seeking and that viewers might receive cognitive gratification from live-streams [115].

These live-streams featuring programming are interesting from an exposure point of view since the people tuning in often do so to watch the streamer. In cases such as in **PUB II**, where the streamer does traditional gaming content as well as programming, viewers who might not have been exposed to programming otherwise have an opportunity to be exposed to programming and CS. Moreover, the community has other experienced programmers who give guidance and encouragement to the newcomers.

As a side note, content creators often do sponsored deals where they play a particular game to promote it. Using popular content creators and streamers who are able to program or are interested in playing games that feature programming might be an interesting approach to CS outreach.

Seemingly, this type of phenomenon has not been investigated in CER – though utilizing games in formal contexts in CS education has been studied for decades. The closest parallel we were able to find is Mohanty & Cantu's work on using commercial games in introductory physics and their note on the games having game-play video available on YouTube and that the games allow self-motivated learning outside classrooms [93].

To gain a deeper understanding of the phenomenon we looked at both datasets from **PUB I** and **PUB II** in **PUB III**. This allowed us to form a categorization of different types of online gaming communities regarding how they discuss CS topics and programming. We categorized the themes discussed into three groups: learning, programming experience, and community. Within each category, we further designated more specific sub-themes. For instance, we further divided learning into *Learning* sub-themes such as *desire to learn CS*, *poor self-efficacy*, *game development*, *learning resources*, *what programming language to learn first*, and *different learning experiences*.

It is interesting to contrast these categories to the experiences in organized game programming outreach by Lakanen et al [80]. They clustered participants of a game programming outreach to five different groups, such as enthusiasts and newbies. They noted differences in the topics that these groups discuss. Similarly, some of the themes that we found are more pertinent to newcomers (e.g. "which programming language should I learn first?") than to more experienced programmers (e.g. optimization and efficiency).

In summary, there are online gaming communities that do discuss CS content and programming. The CS topics discussed are relevant for the content produced and the participants in these communities discuss CS and programming. Moreover, since newcomers and complete beginners when it comes to programming participate in these communities, these new types of media might be particularly interesting when it comes to outreach in CS.

RQ1.2 *How playing games in adolescence is related to choosing to major in computer science?*

PUB IV contributes to the area of career selection and especially choosing CS as a major. The analysis of broad essays regarding the students' background with gaming combined with the more targeted interviews allowed us to have an understanding of how gaming habits and career selection in CS are related. This work provides evidence for the idea presented in earlier literature that studying CS and playing games early on seem to be linked [77, 76].

In addition, we provide three categories describing the various ways in which students saw games as relevant to their CS education. Firstly, games triggered interest in computing in general, e.g. to be able to operate computers and play games independently. Secondly, students had encountered programming in games, as well as, had programmed games themselves. Finally, games were seen as an appealing option for future employment and several students showed interest in game development as a career.

Given that the number of people playing video games is increasing, understanding the relationship between games and choosing CS can be especially beneficial when planning new outreach activities.

6.1.2 RQ2 How can game-related approaches be utilized in formal CS education?

The investigation into how game-related approaches can be used in CSE was conducted using two different but complementary approaches. Two different server systems were built in order to explore and investigate how gamification and learning systems should be constructed. In essence, the design of the systems and the way they interact with existing online learning tools is important consequently interoperability in gamification tools is crucial.

In addition to the systems, we also investigated their use in online learning activities. More specifically, we implemented badges on a course on algorithms and data structures and polled students' opinion of the badges as well as looked at the log data produced during the course. Though we noticed a positive correlation in interest towards badges and the number of badges collected, overall the effect was small. This was echoed in the statements and thoughts students had about the badge system, where large portion of them felt indifferent towards badges. However, generally they felt that badge systems should be kept in and a small population of students were enthusiastic regarding badges.

***RQ2.1** What kinds of technical solutions are needed to support gamification in computer science education?*

Modern online learning environments are complex systems and careful consideration is needed in order to incorporate new elements to these environments. **PUB V** and **PUB VI** contribute a better understanding of how to do this in practice. **PUB VI** had a much more limited scope in that it only had to support a single system, A+. However, it can be seen as a prototype for the modular architecture that was later on heavily utilized when designing and implement Acos, the system introduced in **PUB V**.

PUB V and **PUB VI** contain a high-level description of how these systems work and also focus on the evaluation of the systems. But in addition to this, both Daechschen and Acos are open source solutions providing additional benefits on top of the descriptions in the publications. First, since they are freely available, instances of both programs can be installed and utilized as deemed useful. In particular, Acos comes with hundreds of online learning activities which can be incorporated into existing course materials. And secondly, studying the structure and source code may offer insights in designing similar systems.

Interoperability and service-oriented architectures have received research interest in the educational context, for example in developing LMSs and various interoperability protocols [91, 29, 66, 20]. However, we were only able to find one other system, presented by Wüster & Ebner [133], that described integrating a badge system to an online learning environment. They provide a very high-level description of the system in their article but the system itself is not documented in detail or available publicly. In addition, based on their description the system seems to be tied to a particular technology (Java Servlets and JSP). Compared to this, Daechschen offers an open source solution that is stateless and can be utilized with any programming language or framework. However, presently it only supports interoperability through A+ protocol.

Interoperability has been identified as one of the key issues in increasing the adoption of online learning activities [21]. As such, an important contribution of **PUB V** is the interoperable architecture that is the key feature of Acos. This has not only meant that the system has been utilized in multiple universities spanning multiple continents but in addition to that, more content has been added to Acos which has broadened the use of Acos beyond CS. Recently, content types for Acos have been expanded to include small language exercises for English courses[1].

RQ2.2 *How do students react to adding achievement badges to online exercises?*

Based on the reactions that students had on the achievement badges we concluded in **PUB VII** with some recommendations for other educators and researchers using badge systems. First of all, we found that there was a small portion of students who did not like the badges at all and hence opting out of them should be made possible. Secondly, if there are already point systems in online learning environment, creating badges that are awarded based on collecting points offers little novelty. Because of this, any badges created should target other aspects of learning (e.g. carefulness) that are not covered by the point systems. The third recommendation we had is to clarify the gamification scheme so that students are aware of the relationship, if any, between earning badges and the course grading schemes (e.g. typically badge schemes do not affect the final grade). And finally, great care should be taken that badges do not hinder the usability of the online learning system in any way and also that the badge systems do not impede the learning experience.

Additionally, we were able to show a positive correlation between students' interest in the badges (viewing their collected badges) and the total number of badges they achieved. Previously, the positive correlation between collected badges and summary views has been discussed on a population health course [32]. We were able to show that a similar correlation exists in a CS context.

Overall, these findings are similar to prior research on badges even though we were not able to show statistically significant changes in behavior. This is somewhat surprising since Hamari et al. reported in their literature review that the quantitative studies they included all showed

[1]More information is available at `https://apluslms.github.io/events/1st-a-plus-con/acos-point-click-drag-drop.pdf`

either all or part of the tests positive [5]. One possible explanation for this could be that the specific badges and the environment into which they were implemented enticed students with different goal-orientations differently. As goal-orientation has been shown to affect interest in achievement badges [12, 6].

It is interesting to compare the results from **PUB VII** to those of Hakulinen et al. [57], which was included in the literature review by Hamari et al [5]. Hakulinen et al. did notice statistically significant effects on student behavior with a between-subject setup. Whereas the experiment in **PUB VII** was within-subject. To our knowledge at the time, no prior within-subject experiment in CS education context with badges had been conducted. The comparison is particularly interesting since the experiment in [57] was conducted on the same course a year earlier as the experiment in **PUB VII**. However, the badges were implemented into a different LMS and the badge design itself was different as well.

6.2 Validity and Trustworthiness

This section focuses on the validity and trustworthiness of the work presented. We begin with discussing the qualitative work in **PUB I**, **PUB II**, **PUB III**, **PUB IV**, and **PUB VII**. Then we go on to discuss other remarks regarding the validity of the work.

Regarding the representativeness of the data we, have followed the guidelines recommended by Sandelowski: *"An adequate sample size in qualitative research is one that permits – by virtue of not being too large – the deep, case-oriented analysis that is a hallmark of all qualitative inquiry, and that results in – by virtue of not being too small – a new and richly textured understanding of experience."* [112]

Qualitative Research

When it comes to qualitative research, Guba & Lincoln have outlined four trustworthiness criteria for evaluating the work: credibility, transferability, dependability, and confirmability [51].

To increase *credibility* – which can be understood roughly as internal validity of the research – we have sought to collect data continuously along the process and refine our perception of the phenomenon of gaming communities incrementally. Credibility could have been further enhanced by interviewing the participants in these communities and asking for their perspectives on the categories of discussions that we analyzed. We had plans to interview participants in these communities but ultimately were unable to do so due to live-streamers not responding to collaboration requests. However, it would be particularly interesting to do so in future work.

Credibility can also be thought of as how well the results of the research actually match reality. There is a threat that our analysis and conclusions do not accurately describe what is truly going on. To combat this, we have used different data sources in **PUB III** and **PUB IV** to better triangulate our findings. Similarly, in **PUB I**, **PUB III**, and **PUB IV** the results were concluded with multiple analysts.

Transferability is generally considered to be the responsibility of the one who is trying to generalize or transfer results into another context. To aid in this, the analysis methods have been described in the relevant publications. The results in **PUB IV** are perhaps most tied to the local context since the students had a Finnish background and attended a Finnish university. Similarly, the qualitative work in **PUB VII** is tied to the context of a CS course in a Finnish university with Finnish students, though it is possible that the results could transfer to other similar contexts.

Research in **PUB I**, **PUB II**, and **PUB III** we conducted with a global context in mind, at least when it comes to English speaking online gaming communities. Whether the findings of these publications apply to gaming communities with other languages and cultural context is unclear and should be considered carefully before assuming so.

Dependability in qualitative research can be considered to be roughly parallel to reliability in quantitative research. That is to say, how accurate are the results and whether the research could be repeated with similar results, with the caveat that qualitative research cannot truly be replicated or repeated in the same way that is possible with quantitative approaches.

We have aimed at being transparent and tried to explain our procedures in sufficient detail to improve the dependability of the work. Additionally, we have used standard measures of reliability where it was possible, e.g.

in **PUB I** in we used and reported inter-rater reliability. We have also tried to provide ample examples from the datasets used. In addition to these example snippets being illustrative, we hope that they show our reasoning and that the reader is able to assess the dependability of the work based on them.

To enhance *confirmability* – how well our results and work can be traced back to the initial data – we have aimed at describing our procedures during the data collection and analysis with sufficient detail in each of the publications. Our goal has been to enable further research, as well as any studies, wishing to look into the same phenomenon. However, there are some aspects which lower the confirmability of some studies.

Ideally, the raw corpus for each study would be available which would enable other researchers to see whether they would arrive at the same conclusions as we did. However, when it came to texts and interviews with our students we have not released this data for privacy concerns. This is the case for the data used in **PUB IV** and **PUB VII** which lowers the confirmability of those works. This may be especially true for **PUB IV**, in which the open-ended essay question led to many game-related anecdotes, which may not be the case generally.

Naturally, the data used in **PUB I**, **PUB II**, and **PUB III** originated from public sources. In practice, there might be some differences in accessing the data again, for example, due to changes in the search algorithms and how they rank content. This would mean that the search results might not be identical to those that we gathered while working on **PUB I**. Even though there will inevitably be some variation, we expect the publicly accessible data to be very similar to the one we used.

Other Considerations

One significant factor reducing the validity of the work based on publicly available online comments, namely **PUB I**, **PUB II**, and **PUB III**, is the fact that we have no way of knowing who the people participating in these online communities are and what their backgrounds are really like. In short, we have no way of evaluating whether the comments we analyzed were truthful. However, we do not see major incentives for lying in the types of conversations that the participants had. Again, future research including interviews would be particularly interesting.

When it comes to the findings in **PUB IV** it should be noted that the past gaming experiences that the students described happened approximately in the late 1990s and early 2000s. Though there were online gaming communities during those times as well, live-streaming services did not exist and YouTube itself was launched in 2006. As such, the types of communities investigated in **PUB I**, **PUB II**, and **PUB III** did not exist. All this is to say that the work might not necessarily apply to coming generations who are growing up and experiencing games not only by playing them but also by watching other people play.

Research in **PUB VII** was conducted using a within-subject experiment. This might have impacted the results and could explain why there was no statistical improvement in student behaviors, even though we noticed a correlation between achieved badges and interest in badges. Another factor affecting **PUB VII** regarding the students' perceptions towards the achievement badges to keep in mind is that we surveyed the students after the course and only about half of the students responded. This naturally introduces bias in the answers. It is also possible that some students might have disengaged from the course partly due to the gamification and might not have answered to questionnaire at all, even though it was sent to all students registered on the course.

6.3 Constructive Research

Though both Acos and Daechschen have been functional in the tasks that they were designed for, the validity of the research involved needs to be addressed. The work presented in **PUB VI** concluded with guiding principles for designing new badge systems. These guidelines were established based on our experiences with implementing two different iterations of systems supporting gamification. Though the statements were based on the experiences of not just implementing badge systems but also all the prior software development work done by the authors, the statements were not tested in practice at this stage nor were they subject to any systematic empirical investigation.

However, these principles were useful in the design process leading up to Acos and can be seen as a form of triangulation to find crucial design principles to implement interoperable systems in a CS context. And functionally, Acos has operated as hoped and continues to do so.

A small-scale evaluation of the architecture and software was presented in **PUB V** but there are concerns regarding the evaluation that need to be addressed. First, since we gathered opinions regarding the system from those who had been using or developing for the platform already, there is an inherent bias present. Secondly, for the same reason, the number of people interviewed was relatively small, so it is hard to assess whether the findings are generalizable. However, new content by developers previously unfamiliar with Acos has since been created so it seems that our goal of ease of extendability was achieved.

6.4 Ethical Considerations

Overall, the research conducted in this dissertation has been done following good ethical practices in educational research [11]. Special care has been taken to not put any group of students at a particular advantage or disadvantage. Similarly, students' anonymity and privacy have been considered throughout the research.

The data used and analyzed in **PUB I**, **PUB II**, and **PUB III** was gathered from public sources. However, we decided to anonymize even the usernames of the comments we used as examples in the publications and in this dissertation. The students' data was anonymized before analysis as were the written feedback in **PUB VII** and the transcribed interviews and reflections essays in **PUB IV**. Naturally, students could have opted out from all of these studies without it affecting their grades. When it comes to the experiment in **PUB VII** all students were treated equally since it was a within-subject experiment and all students saw achievement badges in the same manner. Similarly, all of the online exercises were same for all students.

Regarding the software in **PUB V** and **PUB VI**, the primary ethical consideration was related to the privacy of the students and student data. Daechschen was designed in such a way that the identity of the student earning badges was not known to the server and the students were only identified by anonymized numbers. In Acos, the content developer may choose which information is logged when someone interacts with an online learning activity. It is possible to log nothing at all or all interactions on a given session.

6.5 Future Work

Because gamification was a much-hyped trend, in education and elsewhere, it has been studied in a variety of contexts. Hence there have been a multitude of studies since **PUB VII** and presumably this line of inquiry to enhance CSE will be further studied. However, the online gaming communities and the exposure to CS is a relatively new phenomenon when it comes to research. Correspondingly, the possible future research outlined here focuses on games and gaming communities and their potential impact on CSE and outreach.

Modding Communities

One game-related approach that was not discussed in this work is modifying games or *modding* as it is colloquially known. Modern games often offer different content editors and utilities that can be used to customize games and create new content for the game. In addition to editors, many games offer APIs which can be used to create more complex behaviors and additions that go beyond adding new graphics or creating new maps in editors. With popular games, communities of *modders* are formed and sometimes the mods themselves make a game more popular than the original base game.

There is some tentative research on learning through modding in formal contexts [41]. However, a deeper look into the modding communities and how CS and programming is relevant for their discussions could be interesting. Traditionally modding communities have tutorials and they are an important part of the larger network of online gaming communities. It would be interesting to look into these communities to see the similarities and differences between the gaming communities discussed in this dissertation.

Backgrounds in the Gaming Communities

This work investigated online gaming communities through the comments and interactions without placing the participants into any particular context or background. It would be interesting to investigate the participants and their background in detail either through more focused interviews or more broadly with surveys and questionnaires. In particular, finding out about perceptions towards CS and attitudes towards programming from the younger members of the audience would be interesting.

In addition to the backgrounds of the participants in gaming communities, it would be interesting to conduct similar research as in done **PUB IV** but in earlier grades. In particular, the participants' possible gaming hobbies and their perceptions towards getting a university degree in CS would be interesting to understand and that information could also be helpful in planning CS outreach activities.

Potential Learning Outcomes from Playing

The work included in this dissertation has been primarily focused on showing the existence of CS discussion in gaming communities and the exposure to CS that they enable. Given that the exposure and discussions are happening, one particularly interesting line of research would be to look into the possible learning outcomes that might happen as a result of playing (or watching someone else play) particular games that feature CS content.

One particularly interesting avenue of inquiry that was not discussed in this dissertation, would be to investigate the potential learning outcomes through implicit learning. Presently, there is some prior research showing that at least in some contexts particular learning outcomes are achieved as a byproduct of playing games [26]. Although this type of investigation is methodologically challenging, it might yield some very interesting results. Additionally, research on informal and implicit CS learning should not necessarily be limited to games themselves but extended to the online communities in which these games are discussed.

References

[1] About Moodle. https://docs.moodle.org/30/en/About_Moodle. Accessed: 2018-01-08.

[2] Blackboard. http://www.blackboard.com/index.html. Accessed: 2018-02-02.

[3] Open edX. https://open.edx.org/. Accessed: 2018-03-02.

[4] The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition. *IEEE Std 100-2000*, pages 1–1362, Dec 2000.

[5] Does gamification work?–a literature review of empirical studies on gamification, author=Hamari, Juho and Koivisto, Jonna and Sarsa, Harri. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3025–3034. IEEE, 2014.

[6] Samuel Abramovich, Christian Schunn, and Ross Mitsuo Higashi. Are badges useful in education?: It depends upon the type of badge and expertise of learner. *Educational Technology Research and Development*, 61(2):217–232, 2013.

[7] Clark C Abt. Serious games: The art and science of games that simulate life. *New Yorks Viking*, 6, 1970.

[8] Luis von Ahn. Games with a Purpose. *Computer*, 39(6):92–94, June 2006.

[9] Leigh Alexander. What makes a game?, 2012. Retrieved 2018-01-07. Available online: https://www.gamasutra.com/view/news/164869/.

[10] John R Anderson, Lynne M Reder, and Herbert A Simon. Situated learning and education. *Educational researcher*, 25(4):5–11, 1996.

[11] British Educational Research Association et al. *Revised ethical guidelines for educational research (2004)*. British Educational Research Association, 2004.

[12] Tapio Auvinen, Lasse Hakulinen, and Lauri Malmi. Increasing students' awareness of their behavior in online learning environments with visualizations and achievement badges. *IEEE Transactions on Learning Technologies*, 8(3):261–273, 2015.

[13] Anja Balanskat and Katja Engelhardt. Computing our future : Computer programming and coding Priorities, school curricula and initiatives across Europe. Technical report, Brussels, 2015.

[14] Tiffany Barnes, Eve Powell, Amanda Chaffin, and Heather Lipford. Game2Learn: improving the motivation of CS1 students. In *Proceedings of the 3rd international conference on Game development in computer science education*, pages 1–5. ACM, 2008.

[15] Jessica D Bayliss and Sean Strout. *Games as a Flavor of CS1*, volume 38. ACM, 2006.

[16] Tim Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. Computer science without computers: new outreach methods from old tricks. In *Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications*, 2008.

[17] Mordechai Ben-Ari. Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1):45–73, 2001.

[18] Rick Bonney, Jennifer L Shirk, Tina B Phillips, Andrea Wiggins, Heidi L Ballard, Abraham J Miller-Rushing, and Julia K Parrish. Next steps for citizen science. *Science*, 343(6178):1436–1437, 2014.

[19] Elizabeth A Boyle, Thomas Hainey, Thomas M Connolly, Grant Gray, Jeffrey Earp, Michela Ott, Theodore Lim, Manuel Ninaus, Claudia Ribeiro, and João Pereira. An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers & Education*, 94:178–192, 2016.

[20] Peter Brusilovsky. KnowledgeTree: A distributed architecture for adaptive e-learning. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113. ACM, 2004.

[21] Peter Brusilovsky, Stephen Edwards, Amruth Kumar, Lauri Malmi, Luciana Benotti, Duane Buck, Petri Ihantola, Rikki Prince, Teemu Sirkiä, Sergey Sosnovsky, et al. Increasing adoption of smart learning content for computer science education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, pages 31–57. ACM, 2014.

[22] Keith Burgun. What Makes a Game?, 2012. Retrieved 2017-11-27. Available online: https://www.gamasutra.com/view/feature/167418/.

[23] Roger Caillois. *Man, play, and games*. University of Illinois Press, 1961.

[24] Martin C. Carlisle. Using You Tube to Enhance Student Class Preparation in an Introductory Java Course. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 470–474, New York, NY, USA, 2010. ACM.

[25] Clement Chau. Youtube as a participatory culture. *New directions for youth development*, 2010(128):65–74, 2010.

[26] Chad Ciavarro, Mike Dobson, and David Goodman. Implicit learning as a design strategy for learning games: Alert Hockey. *Computers in Human Behavior*, 24(6):2862–2872, 2008.

[27] Code.org. Every student in every school should have the opportunity to learn computer science, 2014.

[28] Thomas M Connolly, Elizabeth A Boyle, Ewan MacArthur, Thomas Hainey, and James M Boyle. A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, 59(2):661–686, 2012.

[29] Declan Dagger, Alexander O'Connor, Seamus Lawless, Eddie Walsh, and Vincent P Wade. Service-oriented e-learning platforms: From monolithic systems to flexible services. *IEEE Internet Computing*, 11(3), 2007.

[30] Jan De Houwer, Dermot Barnes-Holmes, and Agnes Moors. What is learning? on the nature and merits of a functional definition of learning. *Psychonomic Bulletin & Review*, 20(4):631–642, 2013.

[31] Edward L Deci, Richard Koestner, and Richard M Ryan. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation., 1999.

[32] Paul Denny. The effect of virtual achievements on student engagement. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 763–772. ACM, 2013.

[33] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. Evaluating a new exam question: Parsons problems. In *Proceedings of the fourth international workshop on computing education research*, pages 113–124. ACM, 2008.

[34] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011.

[35] John Dewey. *Democracy and education*. Courier Corporation, 2004.

[36] Damien Djaouti, Julian Alvarez, and Jean-Pierre Jessel. Classifying serious games: the G/P/S model. *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*, 2:118–136, 2011.

[37] Dota 2 Prize Pool Tracker. The International 2017, 2018. `http://dota2.prizetrac.kr/international2017`. Accessed 2018-01-16.

[38] Nicolas Ducheneaut, Nicholas Yee, Eric Nickell, and Robert J Moore. The life and death of online gaming communities: a look at guilds in world of warcraft. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 839–848. ACM, 2007.

[39] Thomas M Duffy and Donald J Cunningham. 7. constructivism: Implications for the design and delivery of instruction. 1996.

[40] European Commission (EC). Making a european area of lifelong learning a reality. 2001.

[41] Magy Seif El-Nasr and Brian K Smith. Learning through game modding. *Computers in Entertainment (CIE)*, 4(1):7, 2006.

[42] Andrew J Elliot, Holly A McGregor, and Shelly Gable. Achievement goals, study strategies, and exam performance: A mediational analysis. *Journal of educational psychology*, 91(3):549, 1999.

[43] Entertainment Software Association. Essential Facts About The Computer and Video Game Industry, 2017. Available Online: `http://www.theesa.com/wp-content/uploads/2017/04/EF2017_FinalDigital.pdf`. Accessed 2018-01-24.

[44] Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*. University of California, Irvine Doctoral dissertation, 2000.

[45] Brian J Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 40. ACM, 2009.

[46] Marylène Gagné and Edward L Deci. Self-determination theory and work motivation. *Journal of Organizational behavior*, 26(4):331–362, 2005.

[47] Lisa L Galarneau. Spontaneous communities of learning: Learning ecosystems in massively multiplayer online gaming environments. 2005.

[48] Riot Games. Worlds 2015 Viewership, 2015. Available online: `http://www.lolesports.com/en_US/articles/worlds-2015-viewership`. Accessed 2018-01-24.

[49] Barney G Glaser and Anselm L Strauss. *Discovery of grounded theory: Strategies for qualitative research*. Routledge, 2017.

[50] Lindsey Ann Gouws, Karen Bradshaw, and Peter Wentworth. Computational Thinking in Educational Activities: An Evaluation of the Educational Game Light-bot. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 10–15, New York, NY, USA, 2013. ACM.

[51] Egon G Guba and Yvonna S Lincoln. Guidelines and checklist for constructivist (aka fourth generation) evaluation. *Retrieved January*, 23:2010, 2001.

[52] Philip J Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.

[53] Lassi Haaranen and Teemu Lehtinen. Teaching git on the side: Version control system as a course platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 87–92. ACM, 2015.

[54] Lasse Hakulinen. Alternate Reality Games for Computer Science Education. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, Koli Calling '13, pages 43–50, New York, NY, USA, 2013. ACM.

[55] Lasse Hakulinen. *Gameful Approaches for Computer Science Education: From Gamification to Alternate Reality Games*. Aalto University publication series DOCTORAL DISSERTATIONS; 52/2015. Aalto University, 2015.

[56] Lasse Hakulinen and Tapio Auvinen. The effect of gamification on students with different achievement goal orientations. In *Teaching and Learning in Computing and Engineering (LaTiCE), 2014 International Conference on*, pages 9–16. IEEE, 2014.

[57] Lasse Hakulinen, Tapio Auvinen, and Ari Korhonen. Empirical study on the effect of achievement badges in TRAKLA2 online learning environment. In *Learning and Teaching in Computing and Engineering (LaTiCE), 2013*, pages 47–54. IEEE, 2013.

[58] Juho Hamari and Veikko Eranti. Framework for Designing and Evaluating Game Achievements. In *Digra Conference*, 2011.

[59] William A Hamilton, Oliver Garretson, and Andruid Kerne. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1315–1324. ACM, 2014.

[60] France Henri and Béatrice Pudelko. Understanding and analysing activity and learning in virtual communities. *Journal of Computer Assisted Learning*, 19(4):474–487, 2003.

[61] Zorah Hilvert-Bruce, James T Neill, Max Sjöblom, and Juho Hamari. Social motivations of live-streaming viewer engagement on Twitch. *Computers in Human Behavior*, 2018.

[62] Kai Huotari and Juho Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th international academic MindTrek conference*, pages 17–22. ACM, 2012.

[63] María-Blanca Ibáñez, Angela Di-Serio, and Carlos Delgado-Kloos. Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, 7(3):291–301, 2014.

[64] Petri Ihantola and Ville Karavirta. Two-dimensional parson's puzzles: The concept, tools, and first observations. *Journal of Information Technology Education*, 10:119–132, 2011.

[65] Wijnand IJsselsteijn, Yvonne de Kort, Cees Midden, Berry Eggen, and Elise van den Hoven. Persuasive technology for human well-being: setting the scene. *Persuasive technology*, pages 1–5, 2006.

[66] IMS Global Learning Consortium. Learning tools interoperability, 2010. `http://www.imsglobal.org/toolsinteroperability2.cfm`. Accessed 2017-12-17.

[67] David Jonassen, Mark Davidson, Mauri Collins, John Campbell, and Brenda Bannan Haag. Constructivism and computer-mediated communication in distance education. *American journal of distance education*, 9(2):7–26, 1995.

[68] Avi Kaplan and Martin L Maehr. The contributions and prospects of goal orientation theory. *Educational psychology review*, 19(2):141–184, 2007.

[69] Ville Karavirta, Petri Ihantola, and Teemu Koskinen. Service-oriented approach to improve interoperability of e-learning systems. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 341–345. IEEE, 2013.

[70] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira, Jr., and Chedy Raïssi. Watch Me Playing, I Am a Professional: A First Study on Video Game Live Streaming. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1181–1188, New York, NY, USA, 2012. ACM.

[71] Cagin Kazimoglu, Mary Kiernan, Liz Bacon, and Lachlan Mackinnon. A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioral Sciences*, 47:1991–1999, 2012.

[72] Caitlin Kelleher and Randy Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2):83–137, 2005.

[73] Firas Khatib, Frank DiMaio, Seth Cooper, Maciej Kazmierczyk, Miroslaw Gilski, Szymon Krzywda, Helena Zabranska, Iva Pichova, James Thompson, Zoran Popović, et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 18(10):1175–1177, 2011.

[74] Eugene Kim. Amazon Buys Twitch For $970 Million In Cash. *Business Insider*, 2014. Available online: `http://www.businessinsider.com/amazon-buys-twitch-2014-8`.

[75] Päivi Kinnunen and Beth Simon. Building Theory About Computing Education Phenomena: A Discussion of Grounded Theory. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, Koli Calling '10, pages 37–42, New York, NY, USA, 2010. ACM.

[76] Maria Knobelsdorf and Ralf Romeike. Creativity as a pathway to computer science. In *ACM SIGCSE Bulletin*, volume 40, pages 286–290. ACM, 2008.

[77] Maria Knobelsdorf and Carsten Schulte. Computer science in context: pathways to computer science. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88*, pages 65–76. Australian Computer Society, Inc., 2007.

[78] David A Kolb. *Experiential learning: Experience as the source of learning and development*. FT press, 2014.

[79] Antti-Jussi Lakanen, Ville Isomöttönen, and Vesa Lappalainen. Life Two Years After a Game Programming Course: Longitudinal Viewpoints on K-12 Outreach. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, pages 481–486, New York, NY, USA, 2012. ACM.

[80] Antti-Jussi Lakanen, Ville Isomöttönen, and Vesa Lappalainen. Five Years of Game Programming Outreach: Understanding Student Differences. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 647–652, New York, NY, USA, 2014. ACM.

[81] Jean Lave and Etienne Wenger. *Situated learning: Legitimate peripheral participation*. Cambridge university press, 1991.

[82] Lightbot Inc. Lightbot, 2008. `http://lightbot.com/`.

[83] Andrés Lucero, Evangelos Karapanos, Juha Arrasvuori, and Hannu Korhonen. Playful or Gameful?: Creating Delightful User Experiences. *interactions*, 21(3):34–39, May 2014.

[84] Ludum Dare. About Ludum Dare, 2018. `https://ldjam.com/about`.

[85] Janice Malcolm, Phil Hodkinson, and Helen Colley. The interrelationships between informal and formal learning. *Journal of workplace learning*, 15(7/8):313–318, 2003.

[86] Lauri Malmi, Judy Sheard, Roman Bednarik, Juha Helminen, Päivi Kinnunen, Ari Korhonen, Niko Myller, Juha Sorva, Ahmad Taherkhani, et al. Theoretical underpinnings of computing education research: what is the evidence? In *Proceedings of the tenth annual conference on International computing education research*, pages 27–34. ACM, 2014.

[87] Victoria J Marsick and Marie Volpe. The nature and need for informal learning. *Advances in developing human resources*, 1(3):1–9, 1999.

[88] Victoria J Marsick and Karen Watkins. *Informal and Incidental Learning in the Workplace (Routledge Revivals)*. Routledge, 2015.

[89] Victoria J Marsick and Karen E Watkins. Informal and incidental learning. *New directions for adult and continuing education*, 2001(89):25–34, 2001.

[90] Jane McGonigal. *Reality is broken: Why games make us better and how they can change the world*. Penguin, 2011.

[91] Christoph Meinel, Michael Totschnig, and Christian Willems. openHPI: Evolution of a MOOC platform from LMS to SOA. In *Proceedings of the 5th International Conference on Computer Supported Education (CSEDU), INSTICC, Aachen, Germany*, volume 5, 2013.

[92] David R Michael and Sandra L Chen. *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade, 2005.

[93] Soumya D Mohanty and Sergio Cantu. Teaching introductory undergraduate physics using commercial video games. *Physics Education*, 46(5):570, 2011.

[94] Mojang. Minecraft, 2009.

[95] Bill Moorier. Introducing Twitch Creative. *Twitch Blog*, 2015. Available online: `https://blog.twitch.tv/introducing-twitch-creative-fbfe23b4a114`.

[96] Mathieu Muratet, Patrice Torguet, Fabienne Viallet, and Jean-Pierre Jessel. Experimental feedback on prog&play: a serious game for programming practice. In *Computer Graphics Forum*, volume 30, pages 61–73. Wiley Online Library, 2011.

[97] Lee J Nelson, Christopher J Cushion, and Paul Potrac. Formal, nonformal and informal coach learning: A holistic conceptualisation. *International Journal of Sports Science & Coaching*, 1(3):247–259, 2006.

[98] Markku Niemivirta. Motivation and performance in context: The influence of goal orientations and instructional setting on situational appraisals and task performance. *Psychologia*, 45(4):250–270, 2002.

[99] Ray Oldenburg. *The great good place: Café, coffee shops, community centers, beauty parlors, general stores, bars, hangouts, and how they get you through the day*. Paragon House Publishers, 1989.

[100] Aura Paloheimo, Kaisa Pohjonen, and Pirjo Putila. Women and higher engineering education—Choosing one's degree program. In *Frontiers in Education Conference (FIE), 2011*, pages T2H–1. IEEE, 2011.

[101] Seymour Papert. Does easy do it? children, games, and learning. *Game developer magazine*, 1988.

[102] Marisssa Payne. These five gamers just won $11 million playing dota 2. *The Washington Post*, 2017. `https://www.washingtonpost.com/news/early-lead/wp/2017/08/12/these-five-gamers-just-won-11-million-playing-dota-2/`. Accessed 2018-01-15.

[103] Maja Pivec, Olga Dziabenko, and Irmgard Schinnerl. Aspects of game-based learning. In *3rd International Conference on Knowledge Management, Graz, Austria*, pages 216–225, 2003.

[104] Elizabeth Poché, Nishant Jha, Grant Williams, Jazmine Staten, Miles Vesper, and Anas Mahmoud. Analyzing user comments on YouTube coding tutorial videos. In *Proceedings of the 25th International Conference on Program Comprehension*, pages 196–206. IEEE Press, 2017.

[105] Leo Porter, Cynthia Bailey Lee, and Beth Simon. Halving fail rates using peer instruction: a study of four computer science courses. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 177–182. ACM, 2013.

[106] Marc Prensky. Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1):21–21, 2003.

[107] Open Badges Project. Issuing Open Badges, 2016. Retrieved 2017-01-06. https://openbadges.org/get-started/issuing-badges/.

[108] Howard Rheingold. *The virtual community: Homesteading on the electronic frontier*. MIT press, 2000.

[109] Riot Games. League of Legends, 2009. https://leagueoflegends.com/.

[110] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[111] Richard M Ryan, C Scott Rigby, and Andrew Przybylski. The motivational pull of video games: A self-determination theory approach. *Motivation and emotion*, 30(4):344–360, 2006.

[112] Margarete Sandelowski. Sample size in qualitative research. *Research in nursing & health*, 18(2):179–183, 1995.

[113] Simon. *Emergence of computing education as a research discipline.* Aalto University publication series DOCTORAL DISSERTATIONS; 150/2015. Aalto University; Aalto-yliopisto, 2015.

[114] Teemu Sirkiä. Jsvee & Kelmu: Creating and tailoring program animations for computing education. *Journal of Software: Evolution and Process*, 30(2):e1924, 2018.

[115] Max Sjöblom and Juho Hamari. Why do people watch others play video games? an empirical study on the motivations of twitch users. *Computers in Human Behavior*, 2016.

[116] Juha Sorva, Ville Karavirta, and Lauri Malmi. A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education (TOCE)*, 13(4):15, 2013.

[117] Squad. Kerbal Space Program, 2015. https://kerbalspaceprogram.com/en.

[118] Jaakko Stenros. The game definition game: A review. *Games and culture*, page 1555412016655679.

[119] Paradox Development Studio. Europa Universalis IV, 2013. https://www.paradoxplaza.com/europa-universalis-iv/EUEU04GSK-MASTER.html.

[120] Bernard Suits. *The Grasshopper-: Games, Life and Utopia*. Broadview Press, 2014.

[121] Grant Tavinor. *The art of videogames*. John Wiley & Sons, 2009.

[122] Clive Thompson. The Minecraft Generation. *The New York Times*, 2016. Available online: http://www.nytimes.com/2016/04/17/magazine/the-minecraft-generation.html.

[123] Steven L Thorne, Rebecca W Black, and Julie M Sykes. Second language use, socialization, and learning in internet interest communities and online gaming. *The modern language journal*, 93:802–821, 2009.

[124] Greg Tito. Computer Built in Minecraft Has RAM, Performs Division. *The Escapist*, 2011. Retrieved 2018-01-09. Available online: http://www.escapistmagazine.com/news/view/109385-Computer-Built-in-Minecraft-Has-RAM-Performs-Division.

[125] Adilson Vahldick, António José Mendes, and Maria José Marcelino. A review of games designed to improve introductory computer programming competencies. In *Frontiers in Education Conference (FIE), 2014 IEEE*, pages 1–7. IEEE, 2014.

[126] Matthew Ventura, Valerie Shute, and Yoon Jeon Kim. Video gameplay, personality and academic performance. *Computers & Education*, 58(4):1260–1266, 2012.

[127] Janice Waldron. User-generated content, YouTube and participatory culture on the Web: Music learning and teaching in two contrasting online communities. *Music Education Research*, 15(3):257–274, 2013.

[128] Scott A Wallace, Robert McCartney, and Ingrid Russell. Games and machine learning: a powerful combination in an artificial intelligence course. *Computer Science Education*, 20(1):17–36, 2010.

[129] Annmarie R Ward. Promoting Strategic STEM Education Outreach Programming Using a Systems-Based STEM-EO Model. *Research Management Review*, 20(2):n2, 2015.

[130] Nicola Whitton. Alternate reality games for developing student autonomy and peer learning. In *Proceedings of the LICK 2008 Symposium*, pages 32–40, 2008.

[131] Emma Witkowski, Brett Hutchins, and Marcus Carter. E-sports on the Rise?: Critical Considerations on the Growth and Erosion of Organized Digital Gaming Competitions. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*, IE '13, pages 43:1–43:2, New York, NY, USA, 2013. ACM.

[132] Ludwig Wittgenstein. *Philosophical investigations*. John Wiley & Sons, 2009.

[133] Mario Wüster and Martin Ebner. How to integrate and automatically issue Open Badges in MOOC platforms. *Proceedings of the European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCS 2016)*, pages 279–286, 2016.

[134] Zachtronics. Shenzhen I/O, 2016. `http://www.zachtronics.com/shenzhen-io/`.

[135] Cong Zhang and Jiangchuan Liu. On crowdsourced interactive live streaming: a Twitch. tv-based measurement study. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 55–60. ACM, 2015.

[136] Christopher Zorn, Chadwick A Wingrave, Emiko Charbonneau, and Joseph J LaViola Jr. Exploring Minecraft as a conduit for increasing interest in programming. In *FDG*, pages 352–359. Citeseer, 2013.

Aalto-DD 13/2019

ISBN 978-952-60-8385-8 (printed)
ISBN 978-952-60-8386-5 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS