DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
50

RAIMOND-HENDRIK TUNNEL

Video Game Design and Development Bachelor's Curriculum for Estonia

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
50

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
50

RAIMOND-HENDRIK TUNNEL

Video Game Design and Development Bachelor's Curriculum for Estonia

UNIVERSITY OF TARTU
Press
1632

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in computer science on 7 February, 2024 by the Council of the Institute of Computer Science, University of Tartu.

Supervisor: Ulrich Norbisrath, PhD
University of Tartu
Estonia

Opponents: Martin Sillaots, PhD
Tallinn University
Estonia

Jessica Bayliss, PhD
Rochester Institute of Technology
United States

Defense: The public defense will take place on 4 March, 2024 at 14.15 in Narva mnt 18–1021, Tartu, Estonia.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print)
ISBN 978-9916-27-472-9 (print)
ISSN 2806-2345 (pdf)
ISBN 978-9916-27-473-6 (pdf)

University of Tartu Press
www.tyk.ee

Educating the mind without educating the heart is no education at all.
– Aristotle

ABSTRACT

Teaching video game design and development in a higher education institution program is not a straightforward task. Due to the interdisciplinary nature and rapidly advancing development ecosystem of video games, higher education institutions need to make difficult choices when designing such curricula.

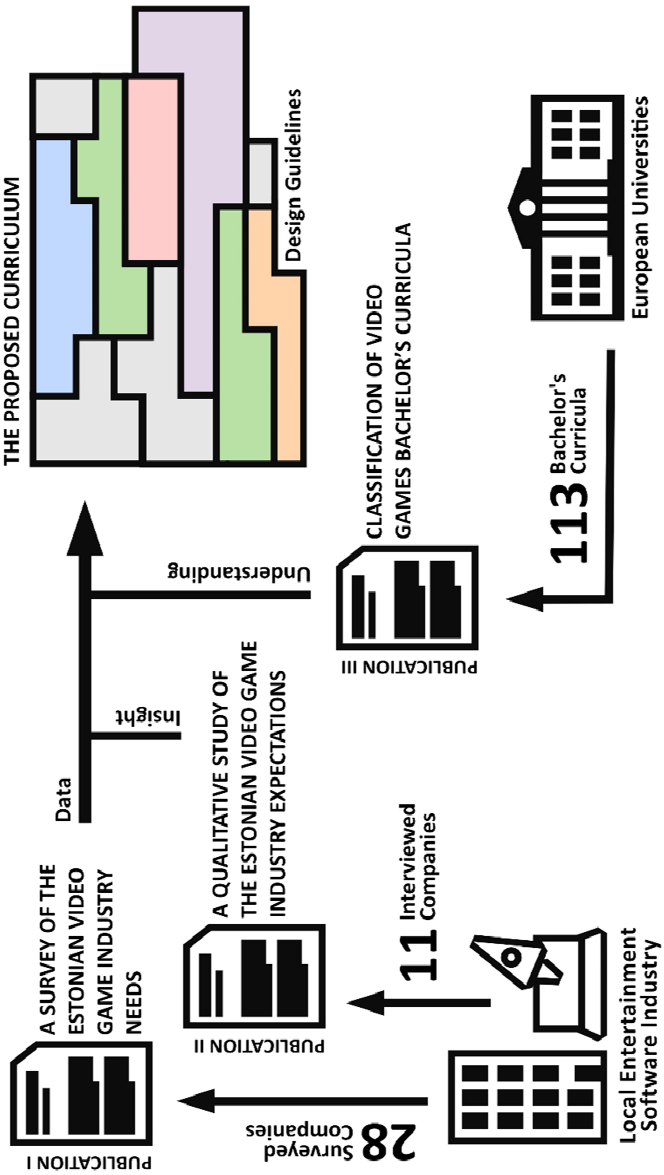
The country of Estonia is known for its technology startups, but the field of video games is largely underdeveloped when compared to nearby countries of Lithuania, Finland, and Sweden. However, due to the country's startup-friendly financial and legal policies and the high level of technological competency in the workforce, there is much potential here. In Estonia's biggest university, the University of Tartu, we wanted to expand our education to cover the needs of that growing sector better.

This dissertation starts by providing a brief overview of the current situation at the University of Tartu and outlining our motivation for introducing a new Bachelor's curriculum. Subsequently, we present three research publications. The first two explore the needs of the entertainment technology industry in Estonia. We surveyed 28 local companies that work in designing and developing video games or closely related products. The survey identifies the current industry needs for programming language use, knowledge areas, soft skills, and contextual fluencies, as well as identifying skills in specific software tools and environments the companies need. The dissertation continues with the second publication, which delves deeper into the industry via semi-structured interviews. The dissertation presents the insights gained in terms of specific expectations companies have for candidates applying for video game designer or developer positions.

The third publication presented in this dissertation explores other Bachelor's curricula around Europe. In a comprehensive study, 113 such curricula were analyzed through profiling, hierarchical clustering, and principal component analysis. As a result, the dissertation presents three clusters of curricula, each with its specific emphasis: video game art, video game programming, and interdisciplinary design and development of video games. The curriculum profiles developed in this dissertation offer insights for curriculum designers to position their own programs in a larger landscape of video game curricula.

Lastly, we have designed a 180 ECTS credits Bachelor's program spanning six semesters. Through designing that program, five guidelines were formulated, serving as fundamental principles for our curriculum. These are presented in this dissertation as potential guides that curriculum designers can consider to make informed decisions and create industry-aligned programs.

VISUAL ABSTRACT



CONTENTS

LIST OF PUBLICATIONS

This dissertation is based on the following publications:

I Raimond-Hendrik Tunnel and Ulrich Norbisrath. 2022. A Survey of Estonian Video Game Industry Needs. J. Educ. Learn. 11, 5 (July 2022), p183. DOI:https://doi.org/10.5539/jel.v11n5p183

II Raimond-Hendrik Tunnel and Ulrich Norbisrath. 2023. A Qualitative Study of the Estonian Video Game Industry Expectations. J. Educ. Learn. 12, 2 (February 2023), p1. DOI:https://doi.org/10.5539/jel.v12n2p1

III Raimond-Hendrik Tunnel and Ulrich Norbisrath. 2023. Classification of Video Games Bachelor's Curricula. J. Educ. Learn. 12, 2 (February 2023), p39. DOI:https://doi.org/10.5539/jel.v12n2p39

AUTHOR'S CONTRIBUTION TO THE PUBLICATIONS

I–III The author of this dissertation is the first author and main contributor of all these publications. Other authors contributed advice, reviewing, and minor content adjustments.

1. INTRODUCTION

The field of video games is very interdisciplinary. Ranging from pure video game art studies through game design, development, and product management curricula to very specific programming and computer science areas. Building quality video games requires all these disciplines to work together, prompting video game creators to interdisciplinary design and understanding. The renowned video game designer and author Jesse Schell really accentuates that point when he answers the question about what skills video game designers need with a simple answer – *all of them* [45].

Game design and development at the University of Tartu, an Estonian national university, as well as in numerous other universities [4,6,8,15,23], is closely aligned with computer science, given its technical nature. Over the past seven years, the Institute of Computer Science at the University of Tartu has been actively creating a range of elective courses focused on video games, computer graphics, and virtual reality (see Figure 1) [33,40,50].

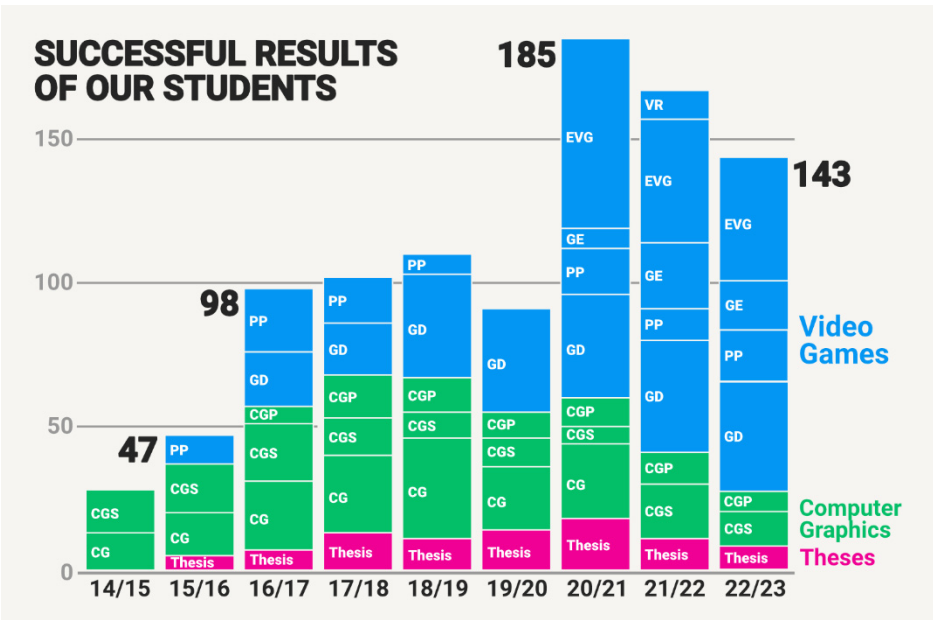


Figure 1. University of Tartu's Computer Graphics and Virtual Reality Study Lab's developed courses and the number of students passing each course. CG – Computer Graphics, CGS – Computer Graphics Seminar, PP – Programming Patterns in Computer Games, CGP – Computer Graphics Project, GD – Computer Game Development and Design, EVG – Evolution of Video Games, VR – Virtual Reality, Theses – both BSc and MSc theses.

These specialized and general elective courses have been developed and offered by the institute's Computer Graphics and Virtual Reality Study Lab, which was founded in 2015. However, the lab's staff has observed that individual elective

courses alone do not adequately prepare our computer science graduates for the competitive entertainment technology job market. Due to time constraints, students often have the capacity to enroll in only one or two of our electives, which results in a deficiency in their game design and development education. Moreover, the limited number of courses restricts our ability to delve deeply into various video game topics. Another important consideration is that students pursuing computer science studies typically possess different interests and motivations compared to those enrolling in a video game design and development curriculum. Although Bayliss and Bierre [6] briefly investigated this aspect, their study was conducted in 2008, and the landscape of video game development has since undergone significant changes in terms of tools and required skills.

Because of these reasons, there has been interest among our lab staff and students for creating a Bachelor's curriculum that would be specifically about video game design and development. Out of this interest grew this dissertation that describes the iterative design process and research using the mixed-methods survey methodology to answer the questions we had when designing a video game design and development Bachelor's curriculum proposal for the University of Tartu.

1.1 Research Questions

When designing a video game design and development Bachelor's curriculum, we had the following research questions:

1. **What are the industry needs?**
   Our curricula should serve the actual needs regarding the knowledge, skills, abilities, contextual fluencies required in the industry.
2. **What are the company-specific expectations?**
   There can be significant differences in what specific companies expect from such a program or its graduates. It is important to know what these expectations are and how they could affect the employability of graduates.
3. **What is the content of the similar curricula?**
   As there are existing programs developed in other institutions, it is important to know what content they include.
4. **What is the stakeholders' feedback to the suggested curricula?**
   As we chose the iterative design process as our method of developing the curriculum proposal, it is necessary to gather feedback from the stakeholders and iterate based on the feedback.

These four questions along with the curriculum design process serve as an outline for this dissertation.

1.2 Research Goals

For each of the four areas defined by the research questions, we had several goals that we needed to achieve to improve our proposed curriculum design.

1.2.1 Local Industry Needs (Publication I)

Our primary goal of the proposed curriculum was to design it to meet the local and potentially global video game industry needs. Without meeting such needs, the employability of the graduates would be poor, and thus, the curriculum would not serve its main purpose. According to a 2022 study done by the Estonian Institute of Economic Research [12], the growth of the local entertainment software industry is hindered by the lack of a talented workforce. The survey does not provide specifics for the required talent but states that the lack of it is predicted to slow down the development of the entertainment software sector in the next 3–5 years.

Thus, to build a higher education program that would tackle that issue, we would first need to know what talent exactly the entertainment software industry needs. From previous studies, a direct answer to that is given by McGill in their 2009 survey of video game studios in the US and Canada [37]. While their research was about identifying the expectations gap between what academia provides and the industry needs, the mapping of the latter provides good information about what talent is sought after by the companies.

Unfortunately, the survey by McGill is more than a decade old, and the technologies and tools used by the industry have most likely changed. For example, their survey, with possible exceptions of Director and Flash, does not cover game engines at all. This is because third-party game engines started to gain popularity only in the mid-2010s [2]. Based on our own experience and from discussions with the industry representatives, we gathered that other needs might have also changed and, thus, a new survey was required. Thus, the goal was to gather enough data ourselves about the industry's needs to make informed choices regarding our curriculum design.

1.2.2 The Company-Specific Expectations (Publication II)

In addition to determining the needs of the local industry, we wanted to get more insight into the individual Estonian video game companies. Our goal was to listen to what the individual companies deemed important when hiring new employees, what roles and role overlaps they have internally, and what they think about the iterations of the proposed curriculum design.

1.2.3 The Content of Similar Curricula (Publication III)

We needed to investigate what contents such a curriculum could have, or, even more importantly, are there different types of video game curricula that already exist. The goal for such an investigation was to gain a better understanding of

what successful programs have been designed, what are their differences, and figure out how our program would be similar to or different from them.

One specific subgoal was how to decide if our proposed program should reward a Bachelor of Science or a Bachelor of Arts degree.

1.2.4 Stakeholder Feedback

A very important goal for us was to gather stakeholder feedback on the proposed curriculum draft and make suitable changes based on that. There are different stakeholder groups like the industry, educators, program managers, and the institute's administration.

An essential stakeholder group for a curriculum is its potential students. Thus, one of our goals was to find out if our proposed curriculum would appeal to potential new students.

We also wanted to somewhat alleviate the bias of only gathering feedback from a close circle of stakeholders. So, one of the goals became reaching more people through presentations at public events.

1.2.5 The Proposed Curriculum

These previous goals all served the larger goal of iteratively designing a curriculum proposal for the University of Tartu's Institute of Computer Science. The proposal needed to be shown to meet the industry needs, take into account the specific expectations of the companies, be comparable to similar already existing video game programs, and done in collaboration with the different stakeholder groups.

2. THEORETICAL BACKGROUND

When designing a video game higher education program, it is first useful to understand how developing video games has evolved through time. This chapter gives a brief introduction to both early and contemporary video game development. Then, it proceeds with the introduction of the IGDA Curriculum Framework – an approach to characterize the different areas of video game development and serve as a guide for developing video game curricula. Lastly, this chapter includes descriptions of two research papers that were influential to this dissertation.

2.1 Brief History of Early Video Games

The design and development of video games started in the early 1970s. The development of cheap consumer-grade electronics allowed for building and mass-producing human-sized cabinets, which were built for housing usually a single game. Such games were called arcade games and were playable in local gathering spaces like malls, bars, and amusement parks [62]. Building such video games required a good understanding of electronics. The founders of Atari, one of the most influential arcade companies at the time, Nolan Kay Bushnell and Samuel Frederick Dabney Jr., were electrical engineers.

The late 70s and early 80s saw the development of home consoles. Video games became not only something played in dark video game arcade locations but were present in family living rooms. These consoles were capable of playing many different games that could be purchased on game cartridges. The development of such games still needed a good grasp of electronics and low-level programming.

However, the design of these video games became more varied. The early arcade games featured quick and reaction-based gameplay due to their coin-based competitive nature. As home consoles allowed for longer and more isolated play, the design could focus more on exploration of the virtual world and narrative-based adventures. For example, the video game *Adventure* (1980), developed by Warren Robinett for the Atari 2600 home console, included exploration of a maze-like environment, environmental puzzles, and item discovery [5].

Together with the rise of home consoles came the popularization of the personal computer (PC). The video games played on PCs are usually called computer games (as opposed to console games). PCs not only allowed people to play games but also build games themselves with programming languages like BASIC and Fortran. To get the most out of the hardware capability, sometimes developers had to write directly in the machine's assembly language [62].

One of the first PCs was the Apple II computer. The designer of that PC, Stephen Gary Wozniak, stated in an interview that he had previously built the game *Breakout* (1976) for Atari in hardware, but with Apple II, he wanted to build the same game in software using the BASIC language. Many of the Apple

II features, like the inclusion of a speaker, are because Wozniak needed them for implementing the *Breakout* game [56].

Unfortunately, the rising video games market in the early 1980s became oversaturated with poorly made video games. These games had bad design and bad programming, were made in a rush, and often were just clones of already existing games [52]. This decline in revenue showed that players were expecting a certain level of quality from the games. In fact, these expectations for both the design and implementation of video games have been rising ever since. The early video games showed that there was a market for them. The video games market crash in 1983-1985 (see Figure 2) showed [52] that successful games have to be made with thoughtful design, focus, care, and innovation in mind.

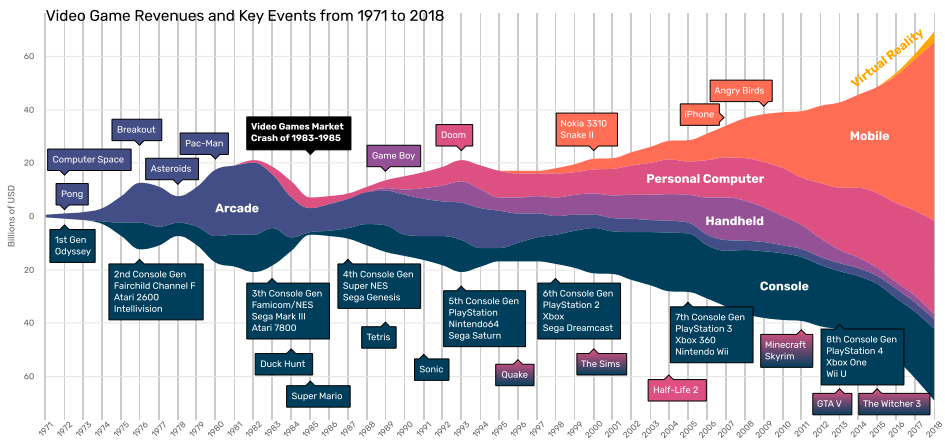


Figure 2. The video games sector revenues and key events from 1971 to 2018.

While early video games could have been made by a few talented electrical engineers or computer programmers, the rising player expectations and the increasing technical capabilities of computers required more inter- and multi-disciplinary people to collaborate and innovate. One example of such is the company id Software [29]. It was founded by an exceptional programmer, John Carmack, a programmer and well-versed game designer, John Romero, a game and level designer, Tom Hall, and an artist, Adrian Carmack. The company produced in the early 90s many influential, fast-paced action games, each breaking new ground in terms of game design and technical capabilities. Those were *Wolfestein 3D* and the *Doom* and *Quake* series. The company later employed more key artists, designers, and programmers. However, the initial composition and skillsets of the founders could be considered instrumental to their success.

So, as the video game industry initially grew out of technically minded computer engineers who wanted to experiment with the early hardware capabilities [44], it very soon needed people with a solid understanding of video games and gameplay, as well as digital artists, sound composers, narrative and world

writers, and much more. Video games became a separate medium that needed its own inter- and multidisciplinary expertise to be successful with it.

2.2 Modern Game Development

As games became more popular and varied, people began to classify them with genres. An influential classification was made by Mark Wolf in 2001, where he categorized games based on their mode of interactivity [54]. This was in contrast to cinematography genres, which were based on iconography (the theme). It became clear that for video games, the way players interact with the game – how the game is played, is a more meaningful way to classify them.

Wolf's original classification was somewhat overspecific, featuring top-level genres like chasing games, dodging games, and Pinball games. Also, as video games have evolved, new genres, like Metroidvania and Battle Royale games, have emerged. In the modern context, a more meaningful and comprehensive system is in Wikipedia [60] and defines ten main genres: action, action-adventure, adventure, puzzle, role-playing, simulation, strategy, sports, MMO, and other games. These also have a lot of subgenres. This system is similar to what is listed by Ernest Adams in his book Fundamentals of Game Design [1], but shooter and platformer games are classified under the larger action game genre, and vehicle and construction simulation games are under a larger simulation genre. It should be noted that games can, and often are, classified under multiple different genres. This is important in study design as, naturally, designing and developing games of different genres is often quite different. Designers would need to understand what makes games of a specific genre appealing to their target audience. Developers of games of a specific genre need to thoroughly understand the requirements of games from that genre.

When making *Doom* (1993), id Software purposefully kept a separation between the general code, for example, graphics rendering, saving and loading, and input parsing, and game-specific code, for example, the specific levels, enemies, and pickups. This separation allowed the general code to be easily reused to make other similar games, like *Doom II* (1994) or numerous Doom clones by other developers [63]. That general code was coined an *engine* or a *game engine*. Other game development companies also built their own engines, such as CryENGINE from Crytek [64], Source from Valve [65], or Creation Engine from Bethesda Softworks [61]. Engines that are used for building games in the same studio and usually have a proprietary license are also called first-party game engines.

Around the late 2000s, more third-party engines entered the market and started to become popular [2,7,53]. Some game engines, like RPG Maker, are dedicated to building games of a specific genre. However, many of the more popular ones allow building almost any known game without the developer having to worry about low-level or even platform-dependent code [19]. It is hard to accurately know how many developers are using specific engines at a given time. A survey from 2019 has found that 72% are using Unity, 27% use

Unreal Engine, and 27% use an in-house proprietary engine in the United Kingdom [46]. The participants of the survey could choose multiple options as some studios use multiple engines. From study design, it is important to understand what tools are needed in the industry to ensure the employability of the graduates. The focus on game engines here is important as they have become popular during a bit more than the last decade. So, when designing a study program, it must be considered that the tools in need at a certain time can easily change after several years.

While Unity was the most popular game engine according to the 2019 survey and likely is such at the time of writing as well, that could be changing due to recent controversial monetization policies [47]. In September of 2023, Unity Technologies announced that they would be charging a download fee for games that are built with their Unity engine. The proposed changes caused many game developers to be disappointed and possibly switch to another engine [27]. Since then, Unity Technologies has redacted the plans, but trust had already been lost. The actual effects of this are yet to be seen. This is an example that even highly popular tools can suddenly lose popularity in the industry, and thus, if a study program is reliant on only a single tool, the value of that program can also suddenly decrease.

Game engines mitigate a lot of the technical challenges that early video game developers faced. Of course, each game features more high-level technical challenges and may need complex technical issues to be solved. The latter is even more true for bigger game studios who are pushing technical innovation. While game engines help a lot with the implementation, a large part of making a successful game is game design. The different components of the game need to function as a whole, providing a cohesive and enjoyable experience. Being able to design a game that provides the desired experience for the player is a very difficult problem that contemporary game studios face [21,48].

2.3 Higher Education Programs

The data about when and how higher education programs start focusing on video games is limited and could potentially be an interesting research topic. Still, we know that a lot of video game education is grown beside or out of an existing computer science program [8,10,13,25,30,31,34,38,42,43]. However, oftentimes, one purpose of integrating video game development into a computer science program is to attract more students to the existing computer science program and potentially help them better learn computer science topics [43,51,55].

It is unclear if learning game design and development through a few elective courses inside a computer science program is generally sufficient to satisfy the needs of the industry. From our experience with that, it seems that it is not. There just is not enough time within their studies for computer science students to learn what they need to in order to be prepared for an entry-level video game designer or developer position in the industry.

What is known in recent times is that the number of game programs in higher education institutions has been growing considerably. For example, a ninefold increase in Canada in the last decade, and in the US, there were over 1200 game-related degree programs in 2018, according to the Higher Education Video Game Alliance [66].

2.3.1 The IGDA Curriculum Framework

To help higher education program designers tackle that issue, the International Game Developers Association (IGDA) has developed a curriculum framework [22]. This lists nine core topics that are meant to be used for game-related education programs. Of course, every program will have its own focus and thus use a subset of these topics. However, it is a good comprehensive overview of topics where each topic would benefit a video game higher education program to a certain amount. Every core topic also has several more specific subtopics. The core topics are:

1. Critical Game Studies
2. Games and Society
3. Game Design
4. Game Programming
5. Visual Design
6. Audio Design
7. Interactive Storytelling
8. Game Production
9. Business of Gaming

The Critical Game Studies core topic is about analyzing games within critical frameworks used in literature, film, and other media. It includes Game Criticism, which goes into investigating, describing, and analyzing the different components of a game, from gameplay and narrative to the influence that games have on the player. This core topic also includes Media Studies, which is about game and gaming journalism.

The core topic of Games and Society investigates games as cultural and societal objects. The first sub-topic here analyzes gaming demographics and the different gaming-related subcultures. Then, the subtopic of History covers how games and their audiences have evolved through time. The Experience of Play subtopic investigates the effect of playing on the player. From the human-computer interaction standpoint, as well as how play transcends national borders, how play creates the *magic circle* [20], and more. The last subtopic here is The Construction of Games and Game Technologies. This one covers how the technologies used for games have evolved through time, as well as the different political stances and cultural practices related to games and game development.

The Game Design core topic includes the Conceptual Game Design subtopic, which is all about creating and improving the ideas behind making a certain game. It is about the game rules, the game space, the gameplay, the

narrative, and the desired and actual experience of the player. The framework states that this is the most intrinsic core topic. It has a subtopic of Practical Game Design, which focuses more on integrating the design with the art and implementation, as well as prototyping, testing on the target audience, and iterating on the design. The last subtopic here is Serious Game Design, which focuses on designing play experiences with specific, often educational, or therapeutic, goals in mind.

The next core topic is Game Programming. This features the different computer science and programming concepts that are necessary for games. From the mathematics of linear algebra and Newtonian physics to object-oriented game design patterns, game engine architecture and design, computer graphics, networking, artificial intelligence, game logic, game data analysis, and more.

The Visual Design core topic includes fundamentals of graphic design as well as basic art and animation techniques. Specific to video games, this core topic includes character and interface design, world design, shading, lighting, visual effects, and game art. But it also includes useful topics like data visualization and procedural content generation.

The subsequent core topic is Audio Design. It includes the technical fundamentals of audio recording and manipulation, as well as music composition. Very importantly, there are topics about designing audio scapes that create certain moods and aesthetic effects depending on the in-game situation.

The Interactive Storytelling core topic is about creating and conveying stories in interactive media. It starts with the fundamentals of storytelling in traditional, non-interactive media such as literature, film, and theater. Then, it includes storytelling in interactive situations. These also include examples like tabletop role-playing games and oral storytelling. Within the context of video games, it is important to understand the player's agency within the story and how to combine the story and interactivity to avoid ludonarrative dissonance.

The Game Production core topic is about the management of game development. As games are made in interdisciplinary, diverse teams, and many different components need to be designed and implemented to fit together, game development can be a difficult task to manage well. This core topic lists important things like the game development lifecycle, development workflows, scheduling, time management, and documentation, as well as team composition, group dynamics, and communication skills.

The last core topic is The Business of Gaming, which is about the business side of video games. This includes topics like the business economy, monetization, platform, and distribution channel choices. Very importantly, it covers the role of the publishers and the deals the developers make with them, as well as laws and regulations about intellectual property rights, patents, contracts, and content restrictions.

The IGDA Curriculum Framework is a very comprehensive overview of what is needed for game development and includes a lot more details than mentioned here. From these 9 core topics, it can be seen that a video games higher education program should cover quite a lot of fields. There definitely

should be a program-dependent focus, but it would be difficult to say which of these topics can be given less attention than others without researching the needs of the industry and what the video game studios expect from their employee candidates.

2.4 Previous Research

Our goal, when starting to design our own proposed curriculum, was to first understand the video games industry requirements for the qualifications of their new employees and what video game programs in higher education institutions satisfy that need. We found two surveys that answered these questions. One by McGill from 2009 [37] and another by Ip from 2012 [23].

2.4.1 Defining the Expectations Gap by McGill (2009)

The difference between what the video game industry needs and what the formal higher education programs provide has been researched before by McGill [37] in a 2009 study. They surveyed both video game studios as well as higher education institutions. Both rated a large number of different knowledge areas, languages, abilities, and contextual fluencies on a Likert scale from 1 to 5. The video game studios rated the competencies according to what the studio is looking for in freshly graduated new employee candidates. The academic institutions rated the competencies according to what they are currently teaching in their video game programs, concentration tracks, or specializations.

While McGill found several mismatches between what the industry expects and what the academia provides, for our purposes, we were interested more in the industry side. The competencies that McGill asked about were gathered in their previous survey from public job advertisements [36]. Looking at them, it became clear to us that some logical choices (e.g., Video Game Design and Video Game Analysis) were missing. There probably could have been even more competencies that would correspond to the IGDA Curriculum Framework topics. Several new languages like JSON, Rust, Go, and visual scripting languages were also something that we were interested in, largely due to input from our own discussions with industry representatives, but were absent in McGill's study.

Furthermore, around 2009, when the study was published, third-party game engines had not become as widespread and used as they are in current times. The emergence of these game engines (e.g., Unity, Unreal Engine, and Godot) has influenced the video game industry landscape a great deal and was certainly information we would need to know when designing a video game design and development curriculum. The emergence of new tools is not only limited to game engines, but we could also talk about the importance of knowing version control tools, 3D content creation tools, project management tools, raster or vector graphics editors, etc.

Thus, McGill's 2009 survey was a very good starting point, but we needed to do another contemporary survey to quantify the video game industry's current needs. This became Publication I, and in addition to a more detailed question about the different tools, we were also able to compare some of our results with McGill, illustrating temporal changes in the industry needs from more than a decade ago.

2.4.2 Fitting the Needs of an Industry by Ip (2012)

When trying to figure out what sort of video game curricula exist and what they consist of, a very thorough 2012 survey by Ip [23] proved to be helpful. In that survey, they used 29 topic areas defined by Skillset (UK creative media industry training organization) and profiled 272 higher educational video game programs in the UK.

Very importantly, they grouped the video game programs into three groups based on their main theme: Games Design, Games Programming, and Games Art. These are logical groups, and the Skillset topic areas themselves are also grouped the same (nine design topic areas, twelve programming topic areas, and eight art topic areas).

From Ip's study, it can be found that the programs from all three groups are broader than their specific group topic areas. The Game Design group curricula feature, on average, 42% of design topic area courses, 21% of programming, and 3% of art topic area courses. The Game Programming group curricula feature 16% of design, 56% of programming, and 2% of art. Surprisingly, the Game Art curricula feature 23% of design, 22% of programming, and only 14% of art topic area courses.

Ip has also studied the profiles of curricula based on the degree they award. The Bachelor of Art curricula include, on average, 51% of design, 8% of programming, 23% of art, and 18% of optional and other courses. The Bachelor of Science curricula include 29% of design, 53% of programming, 5% of art, and 12% of optional and other courses. This is important information, as in communication with existing program managers inside the University of Tartu, we could not clearly understand should our proposed program award a Bachelor of Arts or Bachelor of Science degree. Ip's findings certainly help in that regard.

However, as Ip's study was, like McGill's, more than a decade old, we decided to do a similar contemporary survey and analysis ourselves. That became Publication III, where we performed some further analysis on the collected and profiled curricula to understand better how the curricula differ in terms of content and degree.

2.5 Research Questions and Objectives

As both McGill's and Ip's studies were done more than a decade ago, we felt they might not answer our research questions 1. (*What are the industry needs?*) and 3. (*What is the content of similar curricula?*). Thus, these motivated us to conduct contemporary similar research (Publications I and III).

Naturally, our research questions 3 (*What are the company-specific expectations?*) and 4. (*What is the stakeholders' feedback on the suggested curricula?*) needed us to research further the local industry and the other stakeholders of such a curriculum.

Regarding our research objective of determining the local industry needs, McGill's previous survey was conducted on game development companies in the US and Canada in 2009. While the technologies and needs have likely changed since then, it was also unclear if their results would, even if they would be contemporary, apply on the local Estonian industry. Thus, that survey did not meet this objective in two regards.

Our second objective was to determine the company-specific expectations for their employee candidates of the local companies. This was something that was, as far as we know, previously unstudied in the detail that we required.

Determining the general contents and awarded degrees of similar programs could perhaps be derived from Ip's 2012 study [23] done on the programs in UK. However, we felt not only that the contemporary situation could be different but also that more thorough analysis methods of hierarchical clustering and principal component analysis could yield us additional results and better understanding.

Naturally, our objective of designing a curriculum proposal through iterative design with stakeholders was something that could not have been done based solely on previous studies. Even if we were to design it based on existing curricula and studies, it could not be trusted to be what the stakeholders want (or are capable of as would be the case with educators) without communicating and iterating on it with them. Furthermore, the IGDA Curriculum Framework describes a modular approach that needs to be adapted to a specific situation. It is designed to assist program designers and does not provide a ready-made solution. Thus iterative design and stakeholder engagement was essential.

3. METHODOLOGY

This chapter gives an overview of the methodologies we used in the curriculum design process. The methodologies are presented topically in the order of the research questions. This order does not correspond to the temporal order of the iterative design but should help the reader orient this and subsequent chapters better. In actuality, our curriculum proposal was designed concurrently with the supporting research papers and separate discussions with different stakeholders, including previous and existing program managers. The curriculum was designed iteratively, following the iterative design methodology and applying the mixed-methods survey research methodology. After input from some stakeholders was gathered, and issues in the existing draft were identified, the draft was improved for the subsequent meetings with other stakeholders. This iterative process resulted in 13 versions of the draft. Simultaneously we created and conducted preliminary studies which helped make informed choices about the curriculum design.

3.1 The Local Industry Needs (Publication I)

The chosen methodology for collecting data of the local industry needs was a quantitative study similar to what McGill had done, but with updated questions. To modernize the questionnaire, we first had meetings with four representatives involved in the local entertainment software industry. We showed them the questionnaire that we had been working on. They were asked if there were any terms they found confusing and what other important qualifications were missing. The questionnaire can be found in Appendix A.

As in the survey by McGill, we asked the industry about items in five different categories. These were *Knowledge Areas*, *Languages*, *Software Tools / Environments*, *Abilities*, and *Contextual Fluency*. Each of these featured a number of items that McGill had previously identified [37]. These items were rated in our survey on the Lickert scale from 1 (Strongly disagree) to 5 (Strongly agree) in terms of importance. However, in conversations with industry representatives and from our own experience, it was clear that new items needed to be added and some rephrased or corrected. We also wanted to ask more about specific tools, most of which were absent from McGill's study due to either not yet having been developed or not in widespread use.

Regarding the tools, we wanted to know a bit more than just how important the skills in one a company assesses. Because tools change over time, it is important to be able to adapt from one tool to another. Thus, we restructured the *Software Tools / Environments* category to ask about general categories of tools (e.g., game engines, version control software, project management tools, etc.). Furthermore, for each tool category, we added a question that listed some popular tools identified through prior communication with industry representatives. The survey participants were then asked which of these tools they use and if the potential employee needs to know that specific tools or could skills in

alternatives also be suitable. Our aim with that question was to identify if there are some tools we specifically need to teach and for which tools we could teach alternatives. For example, very popular 3D content creation tools are Blender, Maya, 3ds Max, and Houdini. Different companies may have developed workflows with each of them. From the educational standpoint, we need to know if we need to teach them all or if teaching one or two is sufficient to ensure employability.

The questionnaire also asked how many annual new hires the company would expect to hire from among university graduates that meet their needs. This was an important question for quantifying the local industry's need for a talented workforce and showing if the graduates from a curriculum that would meet the local industry's needs could find local companies to work at.

3.2 The Company-Specific Expectations

During the research and design process we worked closely with several local companies. These included preliminary stakeholder meetings that helped us design the quantitative survey (Publication I) as well as iterate on the design of the curriculum.

3.2.1 Initial Unstructured Interviews

With some industry representatives, we discussed the curriculum drafts using the unstructured interview methodology before the individual semi-structured company interviews described in Publication II. Those were the representatives who were already very involved (e.g., also educators in our current courses) or who represented larger associations (e.g., GameDev Estonia, Tartu Science Park) and, thus, not just from separate entertainment software companies that were interviewed later.

3.2.2 Semi-Structured Interviews (Publication II)

The semi-structured interviews consisted of three parts. In the first part, we asked them seven questions (see Publication II for details) in a semi-structured way about what they are looking for in new hires, what is their company's internal structure and roles, and what the university should focus on in its teaching. One important question was about the breadth of skills a new hire should have. We asked to what level a game programmer is expected to know game design and vice versa. This was important for us to validate our curriculum design, where game Developer and Designer specializations have a big overlap of courses in the core and narrow field modules. For example, the game designers from our proposed curriculum would be able to implement their designs and prototypes with game engines. Respectively, the game developers would know about the games industry, visual design, teamwork, entrepreneurship, and video game analysis – topics that a regular computer science graduate would not be taught. Both specializations also have narrow field

courses that combine game design and game development for different platforms. It was previously unclear if this broader (compared to some more focused programs, see Figure 7) approach to a curriculum is something the industry expects.

The second part of the interview consisted of showing the company three personas. These personas were created through a short questionnaire sent to our lab's graduates. We included the most and least favorite courses, skills with different tools, a demo project, burn charts [11] of their thesis works, motivations, and descriptions of an ideal game development company they would want to work at. The purpose of these personas was to provide a tangible situation where they would need to decide on a new hire. We assumed that through this exercise, we would gain more insight into what the company looks for in new talent. Additional insight that the company perhaps had not talked about in the first part of the interview.

In the third part of the interview, we discussed the proposed curriculum design with the companies in an unstructured way.

3.3 The Content of Similar Curricula

The discovery and classification of the content of similar curricula consisted of a preliminary and a main study (Publication III). The first, preliminary study was conducted on a smaller sample of nearby curricula and was an internal trial for the main study. The main study included a larger population and resulted in Publication III.

3.3.1 Initial Discovery and Classification

We started the design process by looking at existing curricula from other countries near Estonia. When looking for curricula, we included those that had information available in English and listed the courses with their descriptions and the final degree.

We then chose from among the found curricula the Bachelor's programs that had credits listed for each course and proceeded to profile them to see if there were notable distinctions. On further investigation, a program from KAMK featured a specialization choice from among three modules, which we regarded as separate programs. Our overall assumption was that programs in universities of applied sciences would be different than those in general universities. We listed the courses from the and labeled them with the following classifiers: Industry, Design, Development, Math, Game Studies, Soft Skills, Practice, Elect, Other, and Thesis. Each course from the curricula could get one or more of those classifiers with equal weights, and the results were weighted based on the number of credits each course was worth. Summarizing the profiles of courses across the entire curriculum resulted in curriculum *profiles*.

3.3.2 The Main Discovery and Classification (Publication III)

In our main study, we searched the video game Bachelor's curricula in Europe from three sites: Studyportals (studyportals.com, bachelorsportal.com, and mastersportal.com), Educations.com (educations.com), and Keystone Bachelorstudies (bachelorstudies.com). As our aim was to design a 3-year Bachelor's program, we filtered the results only to include such programs. We then chose for further study only the programs that dealt specifically with video games.

Those programs would then be investigated further and only the ones that include easily accessibly descriptions of their courses, credits, and degrees would be included in the subsequent profiling.

The profiling was to be done similarly to the preliminary study, but we used 11 classifiers instead of the initial 10. We decided to focus only on the mandatory courses, discarding the initial Electives category. The initial Practice category was split into Portfolio and Internship. The initial Design category was split into Design and Art to distinguish between courses related to big-picture problem-solving and technical craftsmanship [41]. For clarity, we also renamed the Thesis classifier to Thesis / Project to include capstone projects.

These profiles show the percentages of how much of each classifier / topic the programs include. The classification of the curricula was done using these classifiers, similar to the preliminary study, with hierarchical clustering using Ward's method and the Euclidean distance function.

We purposefully chose three clusters for the hierarchical clustering as Ip had also categorized the video game curricula into three types – design, programming, and art. The design category we called Game Design and Development, as often the programs in that cluster had such or a similar title.

As we were interested in what kinds of programs result in what degrees, we calculated the percentage of curricula in each created cluster that gave either a Bachelor of Science, Bachelor of Art, or some other degree.

We also profiled the curriculum titles based on industry-, design-, art-, and programming-related keywords in them.

3.4 Stakeholder Feedback

Our data collection method during the design process was unstructured interviews that we used on the stakeholders of the curriculum. The stakeholder groups for the proposed curriculum that we gathered input from were:

- Representatives from the local entertainment software industry
- The current educators who were involved in our study lab's courses.
- The potential educators of both the proposed new courses and existing courses that could accommodate the new program.
- The program managers of existing programs in both our and other universities.
- The administration of our institute.

3.4.1 Industry Representatives

Most of the industry representatives were found through the sources mentioned in Publication I. During that study they were asked if they would want to do an interview and be involved in designing, and potentially developing the proposed curriculum. Additional industry representatives were found from among some of the current educators who were also actively involved in game development in the industry, as well as from a company that was established after the survey of Publication I.

The iterative design process consisted of the industry representatives being sent a copy of the curriculum draft and an unstructured interview where discussions took place and suggestions were made. The changes in the design based on the suggestions depended on also the input from other stakeholders and particular capabilities or effectiveness of implementing a suggestion.

3.4.2 The Educators

Among the stakeholder group of educators there were both educators of current courses in our university as well as potential new educators who were open to contributing to the design and implementation of the curriculum in the future. Such potential new educators were often discovered through other stakeholders or from among existing contacts with whom our study lab had successfully collaborated before.

Similarly with other stakeholders, the educators also got a copy of a version of the curriculum proposal and offered suggestions to it.

3.4.3 Program Managers

There are two groups of program managers we communicated with during the curriculum design process. The first are the program managers of other programs in our institute. With them we mostly discussed the feasibility of the proposal and how to design it to both meet the university rules as well as have connections (*e.g.*, for continuing education) to the existing programs.

The other group were the program managers of similar programs outside our university. The methods for discovering them were specific and personal, thus, for protecting their identities, these methods will not be elaborated on.

3.4.4 Administration

Different iterations of the proposed curriculum draft were also shared with the administration of our institute. The questions discussed with the administration involved mostly the administrative requirements for and financial aspects of the proposed program.

3.4.5 Potential Students

Our lab also runs a course dedicated to primary and high school students. The course is called Let's Make Computer Games. It is an online introductory video game programming course where students learn programming in Python and make small computer games with the Python game development library Pygame. It is a heavily modified, updated, and improved [24,39] version of a course created in 2012 [28]. The school students in that course are already interested in developing video games and, thus, are a good sample to see if our proposed curriculum would be more appealing to them than, for example, our current Computer Science program.

We conducted a short survey among the participants of that course. In the survey, we showed a listing of the courses in the Computer Science Bachelor's curriculum and a listing of the courses in our proposed curriculum. All the courses were categorized into their corresponding modules and also included the time estimations that corresponded to the number of credits. We did not provide the curriculum titles to avoid biased choices based largely on the title. Instead, the curricula were just called the *left* and the *right* ones, as they were presented side-by-side. The participants were asked to rate their inclination to study in those curricula on a scale from 0 to 10, where 0 was "not at all" inclined and 10 was "certainly" inclined.

3.4.6 Public Presentations

To present the curriculum design draft at public events, we created slides that conveyed the ideas or work-in-progress drafts of the curriculum. We chose local events that would involve different stakeholders (gaming fans, game developers, students) and that were relatively easy to go to.

3.5 The Proposed Curriculum

The methodology for designing the proposed curriculum was iterative design. The curriculum draft went through several versions. The changes were based on the feedback from the unstructured interviews with the stakeholders, the research results, including the previously unpublished part of the interviews of Publication II, and feedback from the public presentations.

The suggested changes and feedback from the stakeholders and discoveries from the research were assessed and decided upon based on the feedback from other stakeholders and the feasibility of implementation. If a suggestion seemed justified (*e.g.*, corresponded with research results, felt necessary by the educators or other companies) but was unfeasible to be implemented, an alternative solution was found and discussed with the initial proposer.

Throughout the changes, it was important for us to keep the chain or backbone of Game Jam courses that would start in the second semester and continue until the end of the studies. This was inspired by the "Game a Week" teaching methodology discussed by game design professor Bennett Foddy from New

York University and game design senior lecturer Douglas Wilson from RMIT University in their 2018 GDC talk [57]. The idea they have incorporated into their game design courses is for students to make a new game every week throughout the semester. The methodology is supposed to help the students exercise their creativity, not get too attached to a single idea, and more accurately scope their projects.

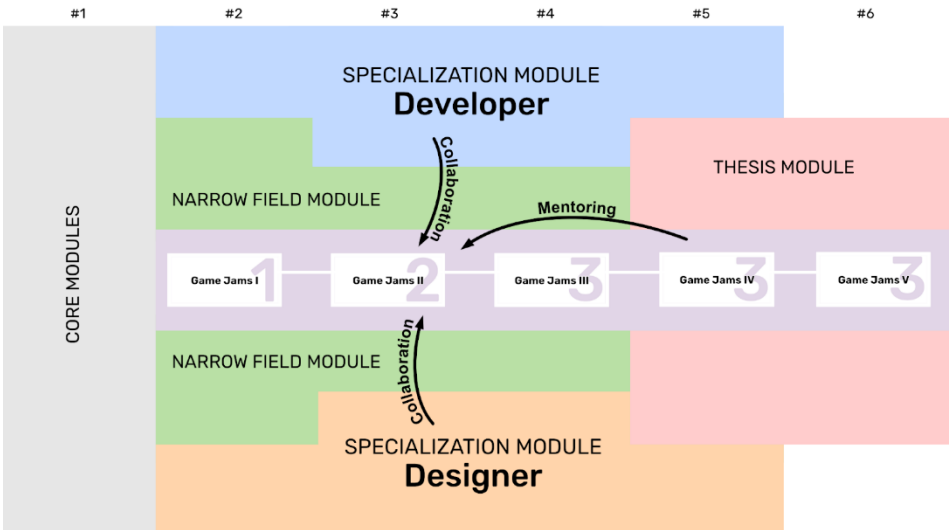


Figure 3. An abstract view of the proposed curriculum modules that shows the interaction between students from the Developer and Designer specializations through game jams.

Another reason for incorporating game jams throughout the studies is that we want the students from both the Designer and Developer specialization to work together often, usually also with artists from Tartu Art School and elsewhere. Thus, game jams serve as nice small-scope projects for developing interdisciplinary teamwork. Furthermore, the plan would be to have students from later years serve as instructors for students in earlier years during the game jams (see Figure 3). This also gives our students skills to assist their less-experienced colleagues – something we believe would benefit them also later in the industry.

Thus, while the individual courses were moved around, changed, and even restructured into core, narrow field, and specialization modules during the iterative design process, we always wanted to keep the core structure of designers and developers from the two specialization modules collaborating through the spine of game jams.

4 RESULTS

There were many results gathered from the different work done in designing the proposed curriculum, and several things were progressing in parallel. Thus, we provide here a timeline of important events and activities related to the work discussed in this dissertation (see Figure 4).

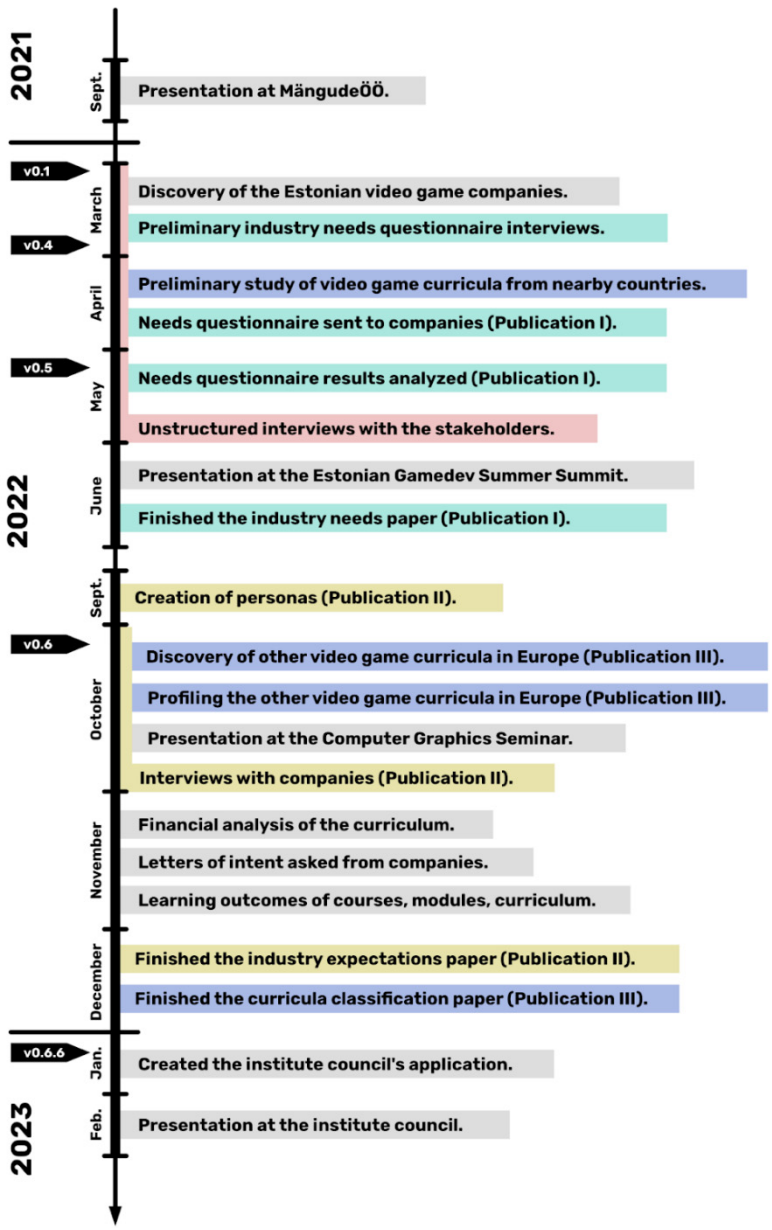


Figure 4. Timeline of the curriculum design process.

4.1 The Local Industry Needs (Publication I)

The local entertainment software companies to participate in the survey (the sample) were found through six sources. First, we looked at the website shipped.ee [3]. This is a privately managed website that lists released commercial Estonian games. The second source was the member list of the Estonian Virtual and Augmented Reality Association [14]. Thirdly, we looked at a list composed by Mark Kendall, which featured Estonian video game startup companies [26]. Then, we looked at another list featuring Estonian game development companies on a website goodfirms.co [18]. As a fifth source, we referred to our own study lab's list of companies we had previously collaborated with [9]. Lastly, we referred to the member list of GameDev Estonia – an association that brings together Estonian video game studios [17]. From all these sources, we chose the companies that had been active in the past 10 years, usually by releasing a product in that timeframe. This process resulted in 70 Estonian entertainment software companies to whom we then sent our questionnaire. Of those, 28 companies answered the survey. The exact lists of asked and surveyed companies are anonymous due to privacy reasons.

From the results of the *Knowledge Area* category, we can make informed decisions on what subjects and by what amount should be covered in the curriculum. For example, the items Code Development (4.71), Video Game Design (4.36), and Game Development Processes (4.36) were rated among the highest in terms of importance. While items that are in general computer science curricula like Relational Databases (2.96), Web Development (2.96), Artificial Intelligence (3.25), and Operating Systems (3.29) were rated comparatively low.

In the *Languages* category, the top-rated items were C# (4.36), visual languages (4.07), and C++ (3.93). The popularity of the C# language could be assumed to be due to the popularity of the Unity game engine. However, due to recent controversies regarding monetization of Unity games [47], that popularity might be subject to change. An alternative popular object-oriented language used in other software development is Java. However, the importance of Java, according to both our and McGill's results, is rated mediocre (2.64) in the entertainment software industry.

The tool category item of game engines was rated extremely important (average 4.6 compared to 4.0 for the next item – version control technologies) in the *Software Tools / Environments* category. More important than the rating was the usage of specific tools. We multiplied the answers to specific tool requirements by the number of new annual hires the company expects. This way, a company that needs, for example, Unreal Engine talent and is ready to hire 20 fresh graduates is more represented than a company that uses some other game engine and maybe is only planning to hire a single graduate every year. In total, there were 53 annual new positions for graduates from all the companies who answered the survey.

For game engines, we discovered that almost everyone uses Unity, and 43 potential jobs out of the 53 need skills specifically in it. Only 5 potential jobs in companies that use Unity allowed for prior skills in alternative game engines. In companies that use Unreal Engine, the situation was similar. From the 32 potential positions in companies that use it, 29 required skills specifically in Unreal Engine, and only in 1 such position prior skills in alternative engines were acceptable (see Figure 5). It is notable that many companies use both of those engines and need their hires to have skills in both.

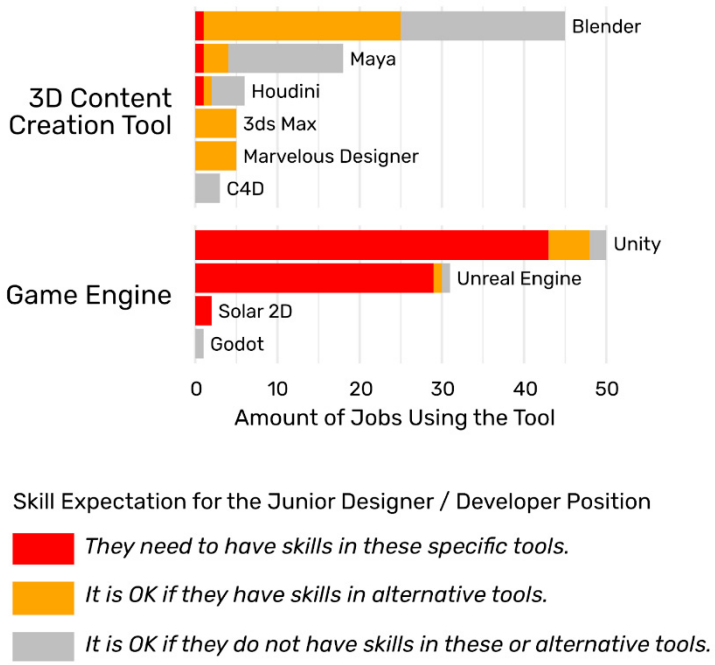


Figure 5. Skill expectations for specific 3D content creation tools and game engines.

In contrast, companies that use 3D Content Creation Tools largely expect their new hires to either not have previous skills in them at all or allow for skills in alternative tools to what they use. For example, 24 potential junior game designer or developer jobs in companies that use Blender do not expect new hires to have any skills in any 3D content creation tools. While 20 of such potential jobs in such companies allow for skills in alternative 3D content creation tools.

While skills with 3D content creation tools are not that much required for these potential job positions, companies did rate the importance of 3D Modeling (3.86), Digital Art (3.79), and Digital Animation (3.79) relatively high in the Contextual Fluency category.

4.2 The Company-Specific Expectations

From the initial unstructured interviews with the company representatives and the three-part semi-structured interviews done in Publication II, we gained a lot of insight into the current local policies of the interviewed companies.

4.2.1 Initial Unstructured Interviews

Based on the initial unstructured interviews, we made many improvements to the questionnaire used in Publication I. For example, the industry representatives said that we should ask about the Rust, Go, GLSH/HLSL, Typescript languages, the previous term "3D modeling" should be renamed to "3D content creation", we should add tools like digital audio workstations (eg, Abelton Live, Audition, FL Studio, Reaper), visual effects tools (e.g., After Effects, Nuke, DaVinci Resolve), version control tools should additionally include Plastic SCM, Google Drive, and Dropbox. They also said that in the knowledge areas category, there should be items "video game design" and "software testing." The languages category also needed something to categorize the visual programming done in Unreal Engine's Blueprints and Unity's Shader Graph. We decided that "visual languages" was a good term for that. There were other similar suggestions and improvements made to McGill's original survey based on the feedback from these industry representatives.

Some of the ideas gained from the initial unstructured interviews regarding the initial curriculum draft were that object-oriented programming (OOP) needs to be complemented with data-oriented programming (DOP). This resulted in us taking a direction towards proposing a new course called Multi-Paradigm Programming, which would include both OOP, DOP, and other paradigms that would be deemed important should the curriculum implementation proceed. Regarding game jams, the industry representatives proposed an idea that some of the internal game jams could be longer than just a weekend, for example, a jam that would last an entire week or even two weeks.

Some of the industry proposals for completely new courses we decided to rather merge into existing courses. For example, there were ideas about courses called Networking in Games, and Databases, Data Structures and Algorithms in Games. One representative proposed that MMO (massively multiplayer online games) programming should be covered.

4.2.2 Semi-Structured Interviews (Publication II)

The data collection method was semi-structured three-part interviews that were conducted on the sample of 11 companies. These were chosen from among the participants of the previous study who had indicated in the quantitative questionnaire that they would be willing to do an interview with us. The exact list of interviewed companies is anonymous due to privacy reasons. The company representatives were usually CEO-s and usually just them. In a few

cases the company had several representatives in the interview, which then included CTO-s, creative directors, and HR managers.

The results of the interviews showed that among the 11 companies, there are a lot of different expectations, hiring processes, and internal structures. Some companies emphasized in the interviews that in their company, the employees need to have a strong drive and take ownership of their work. Other companies mentioned that everyone who wants to could come and contribute to developing their games and products.

The answers about the breadth of skills of designers and developers varied. There were programmers employed in a few of the game studios who knew nothing about the end user and game design. The companies described that these employees just work on their part of the code and do not think about anything else. However, in most of the companies, knowledge about the different other roles and adjacent skills was expected. Usually, programmers were expected to know about game design and game art. One interviewee even said that:

> A programmer who does not know about UX is a bad programmer.

Expectations for the designer regarding prototyping and programming were lower. Often, the interviewed companies said that the designer needs to see the big picture and make paper prototypes but does not necessarily need to know how to code them.

Several companies also attribute managerial responsibilities to the game designer position. For example, in one company, the game designer position was renamed to product manager.

During the persona-driven part of the interviews, several companies stated that they could not make informed choices based on the presented written data alone. The reasoning was that everyone writes appealing text in their applications. What these companies were more interested in was being able to play the applicant's created games. The presented picture and a short description of an example project were not enough.

4.3 The Content of Similar Curricula

4.3.1 Initial Discovery and Classification

Our initial discovery resulted in 24 programs from 16 higher education institutions from the countries of Estonia, Lithuania, Finland, and Sweden (see Figure 6).

This is not a comprehensive list, as there exist other video game programs. For example, the Computer Game Design and Graphics Bachelor's program at the EKA University of Applied Sciences in the neighboring country of Latvia [67]. However, due to limited easily available information on these programs, we decided not to include them in this preliminary study.

Due to a small sample size, the results of the initial discovery and classification are not generalizable, and the clustering turned out to be somewhat unstable. However, at this junction, our goal was solely to get an initial insight into the different programs. The preliminary data from the found 15 Bachelor's curricula showed that there are a few programs that are very specialized in either video game design or video game programming (the Development classifier). Those were Game Design and Graphics from Uppsala University and Computer Game Programming from Luleå University of Technology. We also saw that many of the programs from the universities of applied sciences feature a large percentage of practice. The programs there included many project courses, portfolios, workshops, and internships.

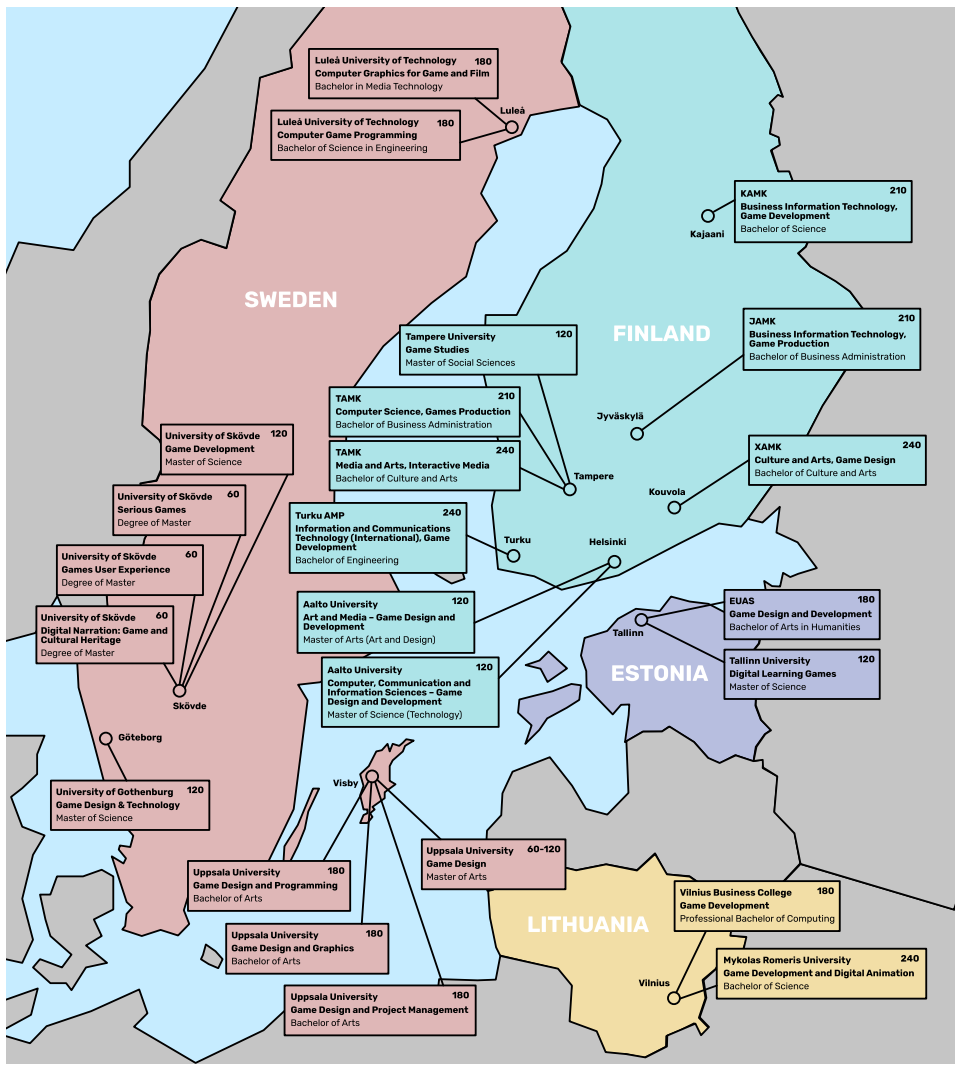


Figure 6. The found 24 programs from 16 higher education institutions from nearby countries.

We also included our own proposed curricula (v0.4) in the profiling and clustering. It was added separately for each specialization module and with the working title of Video Game Designer-Developer. Our Developer specialization module corresponded well with the Game Design and Programming program from Uppsala University – a program that we had taken as a main inspiration in designing our program. The Designer specialization module corresponded well with other programs focused on video game design (see Figure 7).

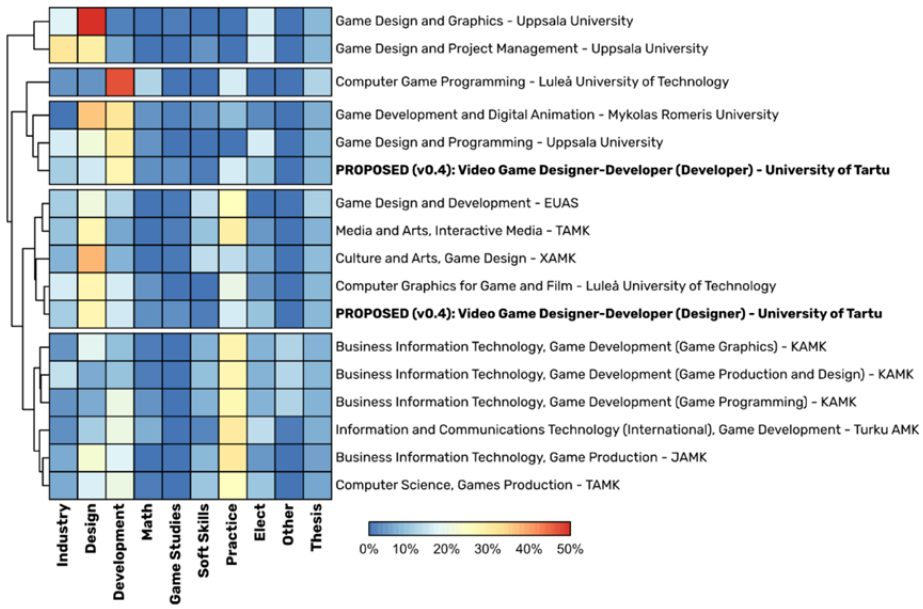


Figure 7. The preliminary clustering of programs from nearby institutions.

Our aim was to create a program that favored dedicated subject courses to practice, and the profiling results showed that this was indeed the case. Compared to programs from Finnish universities of applied sciences like TAMK, KAMK, JAMK, Turku AMK, and the Estonian EAUS, our proposed program has fewer practice courses. However, our proposed curriculum featured more practice than programs from Uppsala University and Mykolas Romeris University.

4.3.2 The Main Discovery and Classification (Publication III)

Our first search yielded 1067 results, out of which 705 were Bachelor's programs. Filtering those to include only the ones that dealt with video games resulted in 230 3-year video game Bachelor's programs. After searching for the information on those 230 programs, we found that not all of them included the necessary information on their respective university websites. While searching for the information, some new suitable programs from the university websites were added. After this last filtering step, we had 113 video game Bachelor's programs with enough information available for profiling.

The clustering results (see Figure 8) showed that the curricula in both the Game Programming and Game Art clusters are very specialized. Large percentages of courses from these clusters are only either about programming (Development classifier) or art. They also tended to mostly give a BSc degree for the programming curricula and a BA degree for the art curricula.

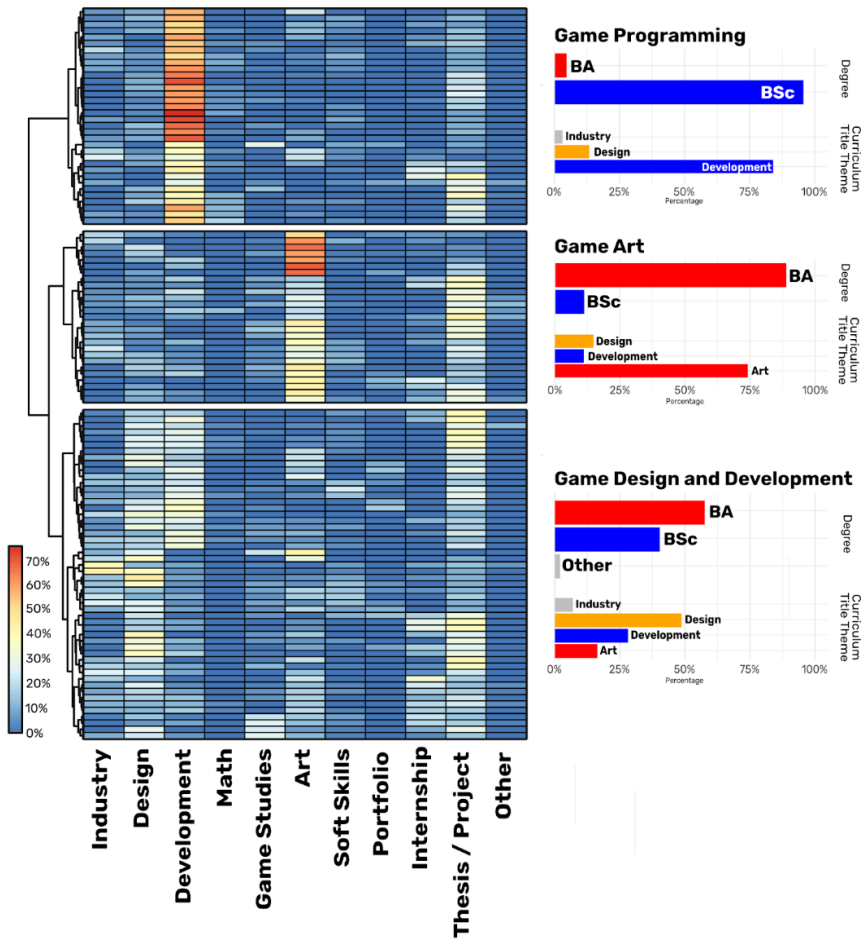


Figure 8. The profiles and the three clusters of surveyed curricula.

What we saw from the analysis was that the curricula in the Game Design and Development cluster tend to give both BA (57%) and BSc (40%) degrees.

The titles in that cluster were quite varied. Some extraordinary examples include Multimedia and Creative Technologies, Computing for Games, Independent Game Design (Game Development), Digital Arts and Entertainment: Game Development, and Computer Science with Game Design.

4.4 Stakeholder Feedback

4.4.1 Industry Representatives

The initial stakeholder meetings with the industry representatives resulted in an updated version of the questionnaire used in Publication I. This is described in more detail in the corresponding chapter.

Most of the discussions with the industry representatives were held during the third part of the interviews conducted in Publication II. During that we showed the companies our proposed curriculum draft v0.6.X. Several companies were very enthusiastic about the capstone project idea and recommended that this needs to be a potential commercial game – ideally even something that the graduates could build their own studio around.

The overall feedback during the third parts of those interviews was very positive. There were a few specific suggestions that were then considered for subsequent iterations of the draft.

In addition to the feedback on the curriculum, we also asked if the companies want to be more involved with either teaching in specific courses or with its management if the proposal is accepted by the institute. Nine of the 11 interviewed companies were interested in contributing by teaching for a couple of weeks in courses that match their expertise and by potentially being part of the capstone project mentoring and grading process. Five companies were interested in being part of a program council that would monitor the effectiveness of the program and propose improvements to meet the changing industry needs better.

4.4.2 The Educators

Throughout the curriculum design, we communicated with the educators who would be involved in the courses of the proposed curriculum. These included thirteen existing and seven potential new educators.

In the data collection method of unstructured interviews, we discussed the latest curriculum draft and how the educator would be willing to contribute to the existing or new courses. For example, in communication with the current educators of the Algorithms and Data Structures course, which is currently taught in Java, it was agreed that there could be an additional C++ group for the students of our curriculum. In fact, such a group was already being planned for additional computer engineering students, so accommodating the extra video game students would not be an issue. Many educators were very welcoming to teach video game students in their courses.

A prominent question in these discussions was how not to create either a so-called *programming* or a *math wall* in the first year. This term was used to refer to an issue where the students find their programming or math courses too complicated for their level and disconnected from their studies. It would cause

them to drop out of the program or not get the sufficient preparation they need for subsequent courses.

Another topic that the potential new programming educators were passionate about was the inclusion of *creative programming* [68]. This programming code that results in sound, animation, color, shapes, etc. The idea is that students should be able to express their creativity through programming, and such an approach could alleviate their fear of code and experimentation.

While discussing the necessary math courses with the current educators, it became obvious that the math needed for video game development and computer graphics differs largely from the math currently taught in our computer science program.

From the design and entrepreneurship side, we initially planned to have a human-computer interaction course in the first semester. The idea was that from the beginning, students should understand that they are building interactive software that needs to be usable (playable) by people. However, in discussion with the corresponding educator, it turned out that they have two Human-Computer Interaction courses, and the one where they could incorporate our students is in the spring semester. So, we moved that course to the second semester and instead proposed that the course Basic Course for Creative Entrepreneur would be in the first semester. In that way, the students would start by learning that they are building creative software products.

One more notable point that arose out of discussions with the educators was about introducing the students to the video games industry.

There were other similar discussions with all the educators of the existing and proposed courses. Many issues were discovered and solved during these interviews. The educators felt confident that if the proposal succeeds, they would happily be ready to teach the new students.

4.4.3 Program Managers

We reached out to and discussed our proposal draft with current program managers of our Bachelor's and Master's Computer Science programs, as well as four external program managers of existing video game programs.

The data collection method of unstructured interviews we used with the previous and current program managers of our Computer Science Bachelor's provided us with data mostly about the high-level organization of the proposed curriculum. For example, in a discussion with the current program manager, we estimated how many new educators would need to be hired for it.

One larger discussion topic that arose with the current Computer Science program managers was the issue of minor specialization. At the University of Tartu, Bachelor students can exchange one of their narrow field and specialization modules with modules from another curriculum. This means that, when designing a program, the curriculum must still be strong even if a student decides to change these two modules in their studies. Such a student still needs to have sufficient skills to be productive in the workforce or continue their

studies. This was a very productive discussion with both program managers, after which the proposed curriculum draft was redesigned to support that. In the revised proposal, the Video Game Development narrow field module and the Developer specialization module were designed to be related to each other, and the Video Game Industry and Game Studies narrow field module and the Designer specialization module became related (please see chapter 4.5 for the proposed curriculum). No matter which narrow field module they change, the students must do the other narrow field and specialization modules. So, in the context of our proposed curriculum, they would miss one of the narrow field modules – the one unrelated to their specialization module – and the practice module. Thus, both the narrow field and their related specialization modules were designed to give sufficient skills to work in the chosen field.

The program managers also said that the university regulations regarding Bachelor's programs are likely to change in the coming years. Thus, it is possible that if the proposed curriculum is to be implemented, it needs to be changed according to the new regulations. Another program manager of a video game program from an outside university similarly said that a curriculum program is never done and always needs updating.

We also discussed the proposed program with the program managers of some of the Master's programs in which the graduates of the proposed program could continue their studies. Those were the Computer Science Master's program at our university and the Digital Learning Games Master's at Tallinn University. Based on the feedback from these program managers, the proposed curriculum draft was checked and updated to meet the requirements for continuing studies in those programs.

The discussions with program managers of external video game Bachelor programs included the troubles that they themselves had faced when building and managing their programs. For example, one program manager said that one of the initial problems in their curriculum was that the students were not actually programming what they had learned about video games. Based on this, we needed to make sure that our programming courses, especially introductory ones, would be about video games and that the programming tasks would be about something they had learned about before. Also, the program manager mentioned that students must learn early on that they are always making things for an audience. They also mentioned that it is very important to explicitly teach conflict management. That program manager discussed a lot more details for specific courses and teaching methods. For the general curriculum context, they said that even if we map the industry needs now when the students graduate in three years, these needs are probably outdated.

Another program manager of an existing video game program discussed that the graduates might not all end up in the entertainment software sector. For example, some of their graduates end up as web developers after graduating. The reasons for that are different. Some may feel that the highly competitive video game industry is too insecure for them. Others just find a different calling. But in a curriculum, it is important to give a bit broader skill than just making

video games so that the graduates who want to work in related fields would still be able to.

4.4.4 Administration

The discussions with the institute's administration mostly revolved around the topic of financing the curriculum. Together with administrators, we assessed the required qualifications and subsequent wages for the potential new educators. In the end, discussions with the administration helped us assemble a sufficient financial analysis for the proposal.

4.4.5 Potential Students

In total, 35 participants of the Let's Make Computer Games course answered the questionnaire. The results (see Figure 9) show that, on average, the students rated their inclination for our proposed curriculum as 8.2 and the current Computer Science curriculum as 5.71. The t-test shows a statistically significant difference ($p<0.01$) among the rated inclinations.

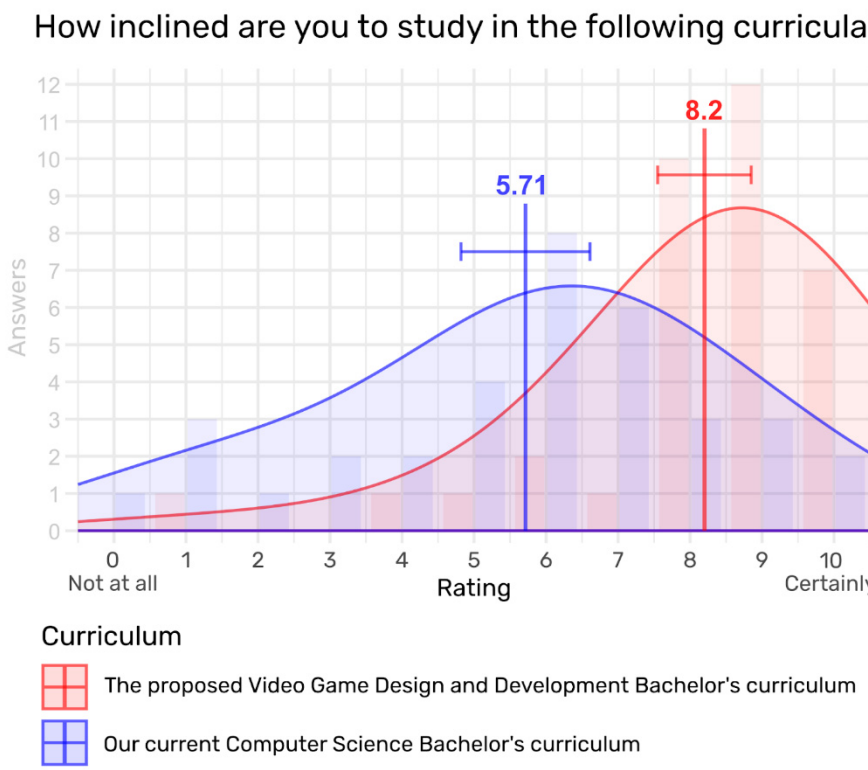


Figure 9. The results of the Let's Make Computer Games course participants survey.

These results show not only a high interest in our proposed curriculum but also that the direct competition between our proposed curriculum and the existing Computer Science curriculum is not very high. Students who are inclined to enroll in our curriculum, on average, are not as inclined in enrolling to the Computer Science curriculum.

4.4.6 Public Presentations

The proposed curriculum drafts were presented to audiences in three instances, two of which were public events. The events, where we presented a version of the curriculum design draft, were MängudeÖÖ, the Estonian Gamedev Summer Summit, and the Computer Graphics Seminar course.

MängudeÖÖ is a much loved and popular gaming and pop-culture event in Estonia with a long history. In 2023, the event took place for the 25th time [59]. A very early vision of the curriculum plan was presented in MängudeÖÖ 2021 as a vision for the future after describing the current studies in our institute [58]. After the talk, several people approached and were very interested in when they could enroll in such a program.

The next presentation was at the Estonian Gamedev Summer Summit event on the 17th of June, 2022 [49]. This event was organized by GameDev Estonia, an association that brings together video game developers in Estonia. The event was targeted at Estonian video game studios, and the purpose was to share experiences and knowledge. The talk was focused on the proposed curriculum and featured the draft version 0.5 of it. The feedback was positive, and several developers wanted to discuss the draft after the presentation.

The last presentation was in a seminar course that our lab gives – the Computer Graphics Seminar. The presentation there was in October of 2022. The goal was to get more feedback from students. Draft version 0.6.3 was shown, and the slides had a link to a feedback form in case someone wanted to give feedback later and anonymously. Through the feedback form, students expressed that they were very happy to see such a program. One concern was that the C++ course was 3 credits, but the student thought it should be more.

4.5 The Proposed Curriculum

Based on the findings from the research conducted in publications I–III, as well as input from industry professionals and educators within our institute, we have designed a 180 ECTS credits Video Game Design and Development Bachelor's curriculum.

During the iterative design process, we wanted the curriculum to match the industry needs as well as possible and incorporate existing courses from the university. Creating too many specialized new courses would not be an effective use of university resources. This did cause mismatches between the industry recommendations and what we could propose. For example, the course Basics Course for Creative Entrepreneur was, during the interviews conducted in

Publication II, recommended to be moved closer to the Graduation Project. The reasoning was that then the students could directly apply the entrepreneurship skills in their final project. However, other educators noted that the third semester is already very intense with courses such as Computer Game Development and Design, and, depending on the specialization, Computer Graphics or Game Design I. These are also courses that seemed most suitable for the third semester by the educators. Furthermore, due to the Human-Computer Interaction course being read in the spring semester, we had to move the Basics Course for Creative Entrepreneur to the first semester – even further away from the Graduation Project. There were several such conflicts, and our goal was to balance the proposed curriculum as well as we could.

The curriculum went through several versions, and not all of them included changes that would be meaningful to cover here. Most of them involved renaming the courses and modules or moving the courses around due to requirements and recommendations from the educators. One larger change was in version v0.6, where we restructured the proposal modules to accommodate the minor specialization required in our university.

At the University of Tartu, Bachelor level curricula are structured around core modules, narrow field modules, and specialization modules, each comprising 24 ECTS credits. Additionally, a practice module is included. The students are required to complete a graduation thesis or project, ranging from 6 to 12 ECTS credits, which may also be incorporated in its own separate and larger module. Out of the total 180 ECTS credits, 6-18 must be for free courses, and 12 must be field-related elective courses.

The proposed curriculum consists of two core modules: *Programming* and *Design and Entrepreneurship*. The guiding principle behind them is to give a solid foundation for programming and interactive software development.

The two narrow field modules would be: *Video Game Industry and Game Studies* and *Video Game Development*. The first would focus on video games in wider contexts (business, cultural, ethical, historical), and the second is about learning to design and develop video games for different platforms.

The specialization modules, of which the students must pick one, are *Developer* and *Designer*. The first specializes in more advanced programming, and the second goes in-depth with game design.

The practice module consists of many game jams, an internship, participation in expos and contests, and individual projects and research. The last module would be the graduation project module, which would focus on creating a portfolio and a larger capstone project.

These modules would leave 6 credits for free courses and 12 for field-related electives, which could include courses such as Computer Handling, Databases, and Operating Systems for more technically minded students.

Figure 10 provides a visual breakdown of the described modules and their courses with corresponding credit loads.

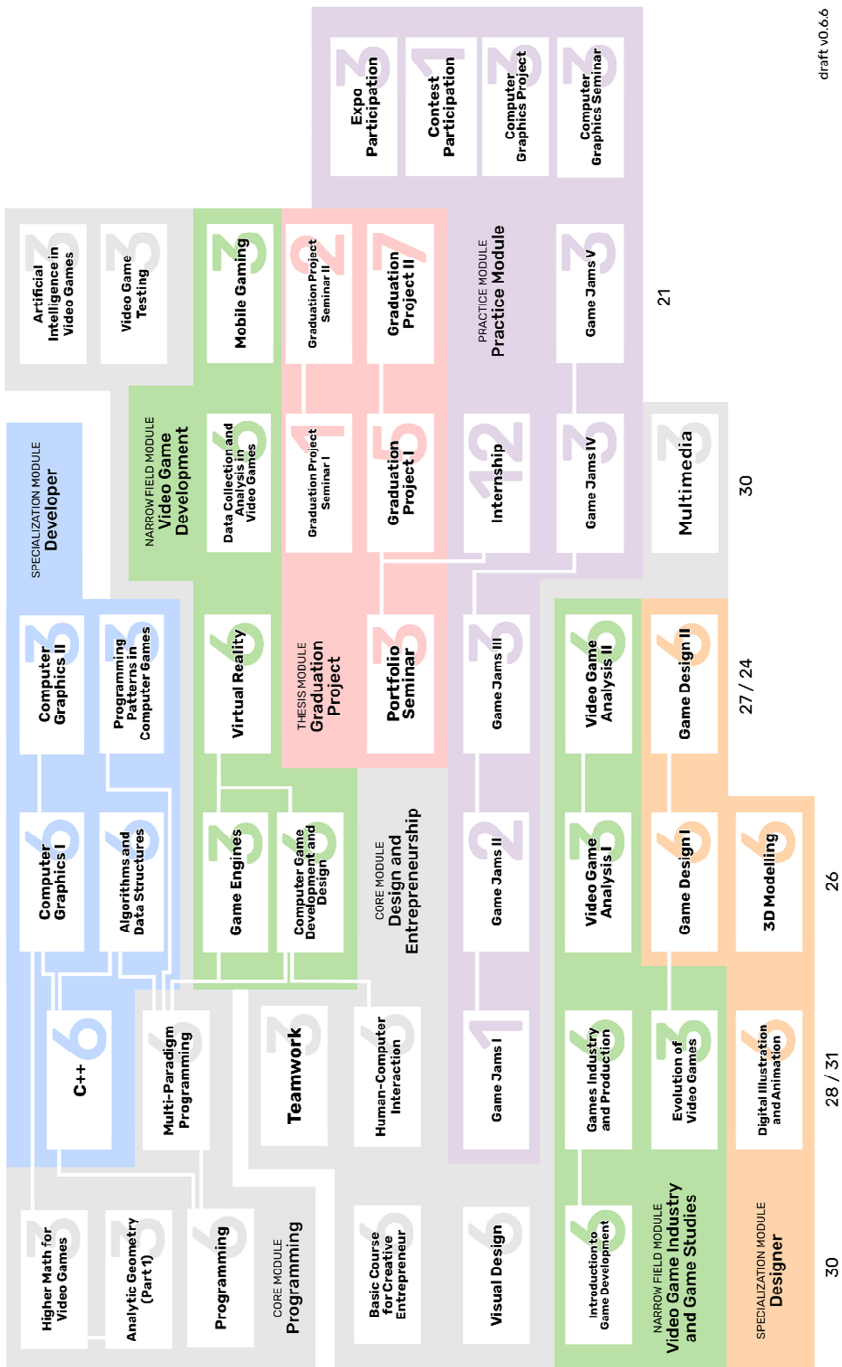


Figure 10. The designed Video Game Design and Development Bachelor's curricula. The numbers on the top show semesters, numbers inside each course are course credits and the numbers on the bottom show the sum of the course credits per semester.

4.5.1 Semester #1

The first semester is designed to be as interdisciplinary as possible. It includes courses on necessary math, programming, entrepreneurship, and visual design. The principle here is to make it clear from the get-go that creating video games touches on many different areas. Being able to express one's ideas with visual design and think of making games as a business must not come only as an afterthought but rather be integrated into the studies right away. Furthermore, in their first programming course, the students would also learn programming by making games. This integrated course already exists in our institute and has been very effective in teaching programming up to the object-oriented level.

4.5.2 Semester #2

In the second semester, the students split into their specialization *Designer* and *Developer* modules. Because of that, it is important to make sure the differently specialized students still stay in touch and collaborate with each other. Thus, in the second semester, the game jams start, and there is a course about Teamwork. An important course here is Games Industry and Production, which brings representatives from the industry to meet the new students and discuss their company cultures and expectations. This sets the students up to think about what sort of companies exist and where they want to work after graduating.

4.5.3 Semester #3

The third semester is the most intense in terms of challenging courses. This is the semester where all the students learn to use different game engines. Our principle here is not to specialize the entire curriculum to a single game engine but rather give the students skills and understanding to use and easily learn different tools for different jobs. Both the specialization modules would also focus on learning advanced subjects like computer graphics, algorithms and data structures, in-depth game design, and basic 3D modeling skills.

4.5.4 Semester #4

In semester #4, there is the portfolio seminar. The portfolio is the first element of the graduation project. During the interviews with the companies in publication II, it became clear that companies value a lot the possibility checking out and playing the candidates' prior games. The portfolio will be essential for the graduates in finding a job after graduating. At that point, there would have been six mandatory game jams and about three to four course projects that the students could use to put together into their portfolios.

4.5.5 Semester #5

The graduation capstone project starts in the fifth semester. This is designed to be a team project lasting two semesters, which would result in a potentially commercial video game. It is where both the *Designer* and *Developer* students work together to make a game that they can be proud of. To supplement that, there are courses on data collection and analysis that encourage data-driven user flow development, and multimedia, to learn video editing for making their game trailers. While there is also an Internship course marked on that semester, the internship can be taken on or split between this and the sixth semester.

4.5.6 Semester #6

In the last semester, there are a couple of small but important supplementary courses about video game testing, specifics of designing games for mobile, and the role of artificial intelligence in video games. When the students complete their capstone projects, the team projects would be assessed together with individual portfolios by a committee during a defense procedure.

5. DISCUSSION

During the iterative design process of the curriculum proposal, it became obvious that there are different opinions on what such a program should contain and how it should be taught. There is a lot to consider and decide. This is also one of the principles behind the IGDA Curriculum Framework that educators should consider what areas of video game development they want to, need to, or are capable of including in their programs.

5.1 The Local Industry Needs (Publication I)

During the industry needs study (Publication I) we asked specifically about the tools that are used in the companies. We noted that the industry needs can change quickly and considerably as a decade ago the third-party game engines only started to gain popularity. We also mentioned that while in our survey the Unity game engines was the most needed, this could change in the future due to questionable management decisions. However, we assume that it will be very unlikely for Unity to be discarded by the industry completely. So, while the importance of C# might decrease in the following years, it will likely remain notably important for the industry.

This is an important consideration for the aspect of designing a curriculum as focusing a program on teaching only one technology could make the program vulnerable to disruptions in that particular technology.

During the curriculum proposal design, we did not focus too much on the specific tools used in the courses. However, in the design of our existing courses, which would be part of the curriculum, we do aim to provide a sufficient variation of tools. For example, we have courses that use Unity and courses that use Unreal Engine, and in our Game Engines course, we teach both with the addition of the Godot game engine.

We also saw in the results of Publication I that the needs for contextual fluencies in 3D Modeling, Digital Art, and Digital Animation were relatively highly rated. This means that there is at least some expectation that a new junior video game designer or developer should be able to communicate professionally with video game artists. This is the reason why, in the curriculum proposal, we included several introductory art courses, mainly in the Designer specialization.

5.2 The Company-Specific Expectations

Both the initial unstructured interviews as well as the semi-structured interviews conducted in Publication II provided useful insight. We found that the local companies differ a lot in their internal policies and expectations for employee candidates.

5.2.1 Initial Unstructured Interviews

The initial unstructured interviews helped us in creating the initial curriculum drafts. However, the initial suggestions included a lot of new custom courses (e.g., Networking in Games, and Databases, Data Structures and Algorithms in Games). We decided to incorporate more already existing courses (e.g., the general Algorithms and Data Structures course from our computer science program) and include the needed game-specific things in other proposed courses. For example, our current course, Game Engines, already features video game networking, and databases could be sufficiently covered inside the proposed Data Collection and Analysis in Video Games course.

One suggestion given during the initial interviews was to include MMO programming in the curriculum. When it comes to this suggestion, we feel that it can be too difficult for a Bachelor's curriculum. Programming multiplayer games is already quite difficult, and programming a game that could support hundreds of thousands of concurrent players seemed to us not to be that suitable for a Bachelor's level program.

However, most of the suggestions were very useful and got, in different ways, incorporated into the curriculum draft and were iterated upon at later stages of design. These initial interviews also helped us create a contemporary version of the questionnaire used in Publication I. Thus, they were essential at the beginning of our iterative design process.

5.2.2 Semi-Structured Interviews (Publication II)

Due to the notable differences in how the companies operate and what they expect, it became clear to us that our proposed Games Industry and Production course needs to facilitate these differences to students. For us, it validated and strengthened our aim of bringing in representatives from different companies in that course to explain their particular expectations and workflows.

The many answers that talked about the different roles having large overlaps in the field of game development, confirmed to us that the students taking the Developer specialization should usually have broader skills than just code programming. They, too, need to be able to understand parts of game design, for example, through the shared narrow field courses and collaborative game jams. They also need to know a bit about game art to make simple changes. Thus, our proposed core Visual Design course is necessary.

Although the companies specified that a game designer does not have to be able to code, we feel with other educators that our current aim of teaching the Designer specialization students to implement games with game engines is beneficial for giving the graduates more opportunities in the job market. The path for everyone is to take the Programming and Multi-Paradigm Programming courses and then design and implement several games for different platforms in subsequent courses and game jams. However, we do agree that game designers should also be able to see the so-called big picture. They should, for example, be able to identify their target audience, create strong

designs for different games, including serious games, understand the messages that their games convey, and tackle larger ethical issues like the Gamer's Dilemma [32]. These are topics that would be in the proposed Game Design I and II courses.

The fact that game designers often have managerial roles, is an indication that besides the Basics Course for Creative Entrepreneur course and the Teamwork course, there should be additional project management topics covered in the Game Design I and II courses. Further research would be beneficial for determining if the game designer position serves more as a product manager or, rather as a project manager role [35].

As the interviewed companies usually desired to try out games made by their potential employee candidates, the graduates of our program must have something to show them. In the proposed curriculum, there are a lot of games made during course projects and game jams. Thus, by the end of the program, the student should have numerous examples to show. However, knowing which ones to choose and creating a solid portfolio is not a trivial task. Thus, in our proposed curriculum, we have a Portfolio Seminar to ensure that every graduate has a nice portfolio of playable games that hiring companies can check out. Furthermore, the capstone graduation project is also proposed as a larger teamwork game project.

5.3 The Content of Similar Curricula

5.3.1 Initial Discovery and Classification

The initial discovery, profiling, and classification of nearby curricula helped us position our design in relation to them. We had an idea what kind of a curriculum we want to design and a preliminary comparison with existing similar (and different) curricula helped find more focus.

The piloted methodology of profiling the curricula and clustering them also proved successful for us, and we proceeded to use it on a much larger dataset in our Publication III to investigate other aspects of existing video game Bachelor's programs.

5.3.2 The Main Discovery and Classification (Publication III)

The hierarchical clustering done in Publication III gave us a much better understanding of what kinds of video game curricula are out there and how they differ in the aspects that we studied. We gained a lot of understanding into the existing curricula and that helped us in the design of our proposal.

As we were designing a design and development curriculum, our interest was in the Game Design and Development cluster of Publication III. It was useful to see that the curricula in that cluster tend to be broader, and some include more courses classified as Industry, Art, and a few even include Game

Studies. While this cluster can certainly be analyzed in more depth and perhaps clustered further, the results were sufficient for us to improve our proposal.

Initially, our working title for the program was Video Game Designer-Developer. During the stakeholder meetings with educators, several of them expressed concern that it could be better. Based on other titles in the Game Design and Development cluster of Publication III, we decided to go with a more generic title of Video Game Design and Development.

From analyzing the clusters of Publication III and the different degrees they reward, it seemed that the Game Art cluster included mostly BA programs and the Game Programming had mostly BSc programs. But for the Game Development and Design cluster that the choice between BA and BSc seems to come down to other considerations that just being art- or programming-focused. In our proposed curriculum, we decided to base the choice on the nature of the student's final work. As our program would end with a portfolio and a large capstone project instead of a written scientific thesis, we concluded that the Bachelor of Arts degree would be more suitable for it.

5.4 Stakeholder Feedback

Most of the work on the curriculum design was based on the stakeholders' feedback. All the different stakeholder groups had their own specific insight into the design and collaborating with them provided a comprehensive overview of what they would expect from, how they could contribute with, and what should we consider in such a curriculum. Stakeholders often raised important issues that should be solved in the design.

5.4.1 Industry Representatives

It was encouraging to see many of the industry representatives enthusiastic about the proposed curriculum. Even more so that many companies wanted to be actively involved in implementing, developing, and monitoring of such a program. It seems very important to involve the industry in such activities as the program is supposed to, ultimately, benefit them.

5.4.2 Educators

The discussions with the educators often involved the co-called *math* and *programming walls*. To alleviate these issues, we wanted to design the initial programming and math courses with effectiveness and focus in mind. We have already given for many years an online introductory video game programming course called Let's Make Computer Games to primary and high school students. That course usually includes pupils already from the 6th and 7th grades, and by the end of 10 study weeks with an approximate 3 credits load, they have achieved the first basics of object-oriented programming. Thus, in discussion with the educators involved in the programming courses of the proposed curriculum, it seemed that extending that course to 6 credits and then having the

Multi-Paradigm Programming course afterward would give the students enough skill to continue developing video games with game engines and possibly learn algorithms, data structures, and computer graphics in subsequent courses.

The discussions also discovered that the current math that is taught to Computer Science students is quite different to what video game students would need. The effective solution that we reached was first to have an introductory analytical geometry course for half a semester, followed by a new course – Higher Math for Video Games.

Educators also emphasized that the students should learn of and be in contact with the local entertainment software industry. For that, we proposed a course in the second semester called Games Industry and Production. From our research in Publication II, it became clear that companies, at least in Estonia, are all quite different in their internal policies and expectations. Thus, our proposal included a course in which there would mostly be representatives from different companies talking about the industry and the specifics of their company. The educator that would be in charge of that course has a lot of contacts in the industry and was very positive about that kind of course structure.

5.4.3 Program Managers

One of the program managers of an existing similar program had mentioned that students should understand early on that they are building things for an audience. This is one of the reasons why we think that the Human-Computer Interaction course needs to be as early as possible.

They also added that the program should explicitly teach conflict management. In our curriculum proposal, we have a dedicated Teamwork course that would include it.

As the program managers mentioned that a curriculum always needs to be updated and is never done, it became clear that when running a curriculum, we need to include industry partners to monitor the curriculum with us, create metrics to determine if the curriculum is providing what the industry expects, and do occasional surveys to update the industry's needs.

An important statement by one program manager was that graduates should be able to find employment in other similar fields as well. In our proposed program, the courses should be organized so that, by the end, the students would be very good real-time interactive software developers. Based on the discussion with that program manager, the possible application of the learned skills in related fields is something we certainly want to provide and monitor. Also, in our proposal, we have purposefully left out the elective courses. These courses could best help the students who want to pivot from video games. For example, a web development course should be listed among the electives.

5.4.4 Administration

While the administration was very supportive of the program and helped with administrative requirements and the financial analysis, they did express the

concern that not everyone in our institute will feel that such a program, one that is about video games, should exist in a university. Experiences with such people were also described by a few other program managers of similar programs. So, while there exist thousands of programs in universities already, the global industry is large and growing, the field is untrivial, and there are scientific journals from ACM and IEEE on the subject, there still exists pushback from the academia.

5.4.5 Potential Students

There were three reasons why we did not do separate interviews or studies directly with university students. First, the current computer science students might not fully be the correct target audience. So, a separate process should have been developed to find out which of the current computer science students would instead enroll in a video game program. Furthermore, one source of motivation for creating the proposed curriculum was due to several of our lab's students already expressing their desires for it. Secondly, we decided it would be more useful to survey a sample of school students who are already interested in developing video games. More about this in Chapter 3.3. Thirdly, we wanted to focus on what the industry expects the students to know after graduation rather than what the students would want to learn in such a program. To clarify, we certainly want to provide a curriculum that the students enrolled in it would like and find useful. However, asking individual students from outside the program could result in a bias towards specific technologies or areas, which might differ from what the industry expects.

The results of the survey conducted on the participants of the Let's Make Computer Games course that showed that that, in direct comparison, more students of that course would rather be interested in our proposed curriculum than in our Computer Science program, is an interesting discussion point. Our guess is that they, given the absence of our program and that they can afford the tuition fee, they would also rather enroll in a similar EUAS video games program in Estonia rather than our Computer Science program. However, this could certainly be studied further, and the actual reasons behind this difference in inclinations would also be useful to know.

5.4.6 Public Presentations

In the Estonian Gamedev Summer Summit presentation, one of the questions was that if this is a program for designers and developers, then where do video game artists come from? The answer to that is that from art schools. In Tartu, we have the Tartu Art School, which has dedicated programs for both 3D designers and graphic designers. Usually, students from Tartu Art School also join our current game jams and work on video game art alongside our programmers. Furthermore, the Pallas University of Applied Sciences features a program about graphic and media design, and the Estonian Academy of Arts also has very strong programs in graphic design and animation.

From the Computer Graphics Seminar presentation that was oriented towards students, the feedback given via the form expressed concerns about the C++ course only being 3 ECTS credits instead of 6. In a later version, we changed it to have 6 credits. Another concern was that students did not understand what the Video Game Analysis course was about, even though it was explained during the presentation. We feel that this is a common issue among computer science students that video games are regarded only as software programs and not as cultural, societal, or literary objects. Even the idea that video games are analyzed as such is foreign to some students who are actually very much into video games. This is something that the proposed curriculum is aimed at improving.

5.5. The Proposed Curriculum

Regarding the "Game a Week" concept [57] that we used as inspiration for our spine of game jams. It must be said that for us, one game every week seems perhaps too intense, but the concept was intriguing. We want the students to rather participate in a limited number of game jams throughout their whole studies. As game jams have a limited time scope, and there is enough time between them, we hope that this would avoid the potential burnout that a very intense schedule of making a new game every week could cause in students.

5.5.1 Design Guidelines

Throughout the research that led to the three publications, the communication with our university educators, students, and industry partners, and the iterative design process we used to create the curriculum, we discovered and established five design guidelines that seemed useful to us and that we followed in designing the proposed curriculum:

1. **All mandatory courses must have an explicit focus on video games.** The curriculum should not have the so-called math or programming wall for students to get stuck at. Rather, the subjects taught here should have clear and explicit benefits for satisfying the students' goals for enrolling in the first place – i.e., satisfy the goal of learning to make better video games. This is something we developed strong feelings for through communication with our current computer science students, who express that many of their mandatory subjects are not very valuable to them. It seems likely that learning subjects with ambiguous values decreases motivation and potentially increases the student dropout rate. However, further studies on that would be necessary.
2. **The contextual fluencies needed in the industry must be covered.** In their future careers, the developers and designers who graduate from this curriculum would need to be able to communicate with artists, musicians, story writers, and other people from different disciplines, as evidenced by the survey result from publication I. The graduates need to have both the

vocabulary as well as enough knowledge of the different disciplines to be able to do so effectively.

3. **The students in the different specialization modules must have collaborative touchpoints and experiences throughout their studies.** This is something accomplished by our *spine* of game jams that starts already from the second semester. Participating in many collaborative projects teaches that people can accomplish more together. Game jams are more suitable for this than regular course projects due to them being in a specific and explicitly bound timeframe. Good teamwork skills are valued in the industry, as seen from both publications I and II.
4. **The projects students build during their studies need to have clear and explicit benefits for the students also after their graduation.** This is why a portfolio must be part of the graduation project, and the capstone video game itself would need to have the potential to be commercialized. The need for it was emphasized by several companies in publication II.
5. **As video games are interdisciplinary, so must be the curriculum**. The graduates of a video game design and development curriculum must fulfill managerial roles between highly specialized programmers and artists. The game makers also have both cultural and economic responsibilities, which cannot be overlooked or ignored during their studies. The education must be comprehensive. This was evidenced in publication III and indicated by the industry in publication II.

While these were the principles we used in our curriculum design process, these principles can, of course, be used for developing and improving existing video game curricula in other universities as well.

5.6 Possible Open Questions and Further Studies

In a previous study Bayliss and Bierre [6] have researched how Computer Science and Video Game Development and Design students differ. That study was done in 2008 and both fields have undergone significant changes in terms of tools and required skills. Therefore, conducting a similar study with modern programs and students would yield valuable insights. Furthermore, as considered in this dissertation, the students of a video game curriculum could switch fields and find employment in other areas. Investigating this could help quantify the benefit that a video game program would have on a local industry.

We created a contemporary survey of the industry needs and in Publication I compared the results with a similar study by McGill's from 2009 [37]. We mentioned that the results of our study might be affected by the recent troubling situation regarding the Unity game engine [27]. It would be interesting to see if that situation has a measurable effect on the industry needs in a year or two. We could assume that as the trust among the game studios in Unity fades, the current dominating position in the industry needs can also change.

In our Publication III we surveyed 113 Bachelor's curricula. We chose ones that had easily accessible details about their courses available in English. Such a survey could be done in a larger scale by also including programs that have the information available in another language or where the information needs to be required directly from the institute. This could enable a possibility of a comparison between programs in English and in other languages.

Another improvement that could be made with a larger dataset based on Publication III would be further clustering. Currently it seems that the Video Game Design and Development cluster could potentially be clustered further. Perhaps into separate Video Game Design and Video Game Development clusters. However, for stable results, a larger set of data would be necessary.

The small survey we did on the participants of the Let's Make Computer Games course could be repeated on regular high school students. That would give a more generalizable result.

Regarding our proposed curriculum design and guiding principles, these are certainly something that should be researched in practice. If such a program were implemented, it would allow for research into its effectiveness, further iterations, and enhancement of the guiding principles.

5.6.1 The IGDA Curriculum Framework

We used the IGDA Curriculum Framework [22] as a basis of what different areas a video game program could include. It does provide a very good and thorough overview of the inter- and multidisciplinary nature of the field. We used it at some of the stakeholder interviews to illustrate that point.

It is true that the framework is from 2008 and could suffer the same issue of being potentially outdated as the previous studies by McGill (2009) [37], Ip (2012) [23], Bayliss and Bierre (2008) [6]. During the work in this dissertation we were in short contact with an IGDA representative, who mentioned that they are working on a 2020 version of the framework [16]. However, information about that is scarce and public announcements about it seem to have stopped around 2019-2020.

Based on our current work, the 2008 version of the framework still seems useful enough for curriculum design. There could be some small improvements done. For example, it could include the MDA (mechanics-dynamics-aesthetics) framework [21] of game design that has become popular in video game design studies.

We suspect that a deeper understanding and more constructive suggestions for the IGDA Curriculum Framework would require more experience of not just designing a curriculum but also implementing it, analyzing, and iterating the design of running curriculum. That would provide a better understanding of what works in terms of curriculum design, what needs to be improved, and how best to convey that information to other curriculum designers.

6. CONCLUSION

As we set out to improve our current video game education at the University of Tartu's Institute of Computer Science, it became clear that we needed to answer four research questions: 1) What are the industry needs? 2) What are the company-specific expectations? 3) What is the content of the similar curricula? 4) What is the stakeholders feedback to the suggested curriculum?

To answer the first question, we updated and replicated the survey conducted by McGill about a decade ago [37]. In our survey, 28 Estonian video game or closely related companies provided quantitative data on their needs in the categories of *Knowledge Areas*, *Languages*, *Software Tools / Environments*, *Abilities*, and *Contextual Fluencies*. Compared to McGill, we observed increases in the need for JavaScript and C#, while there were decreases in the need for C++, C, and XML. From among our added items, skills with visual languages and JSON were rated to be highly needed. Among the other categories, a notable difference was found in the Abilities category. There our results show that Organization / Time Management is needed more than before.

From our survey results, we found the answers regarding the need for skills with specific tools to be the most actionable. Differently from McGill, we asked to rate the tool categories and then had the companies specify which tools they use. For each tool, we asked do they expect the video game designer or developer job candidates to know exactly these tools or would skills in alternative tools also be acceptable. Our results show that the game engines Unity and Unreal Engine are ones that companies expect their job candidates to be specifically skilled in, and alternatives for those would not be suitable. Furthermore, several companies marked them both as necessary. This is an important result that leads us to propose that a video game design and development program should not be teaching just a single game engine.

While our survey in publication I gave us a lot of actionable quantitative data, it lacked in terms of providing more in-depth qualitative data and could not answer our second research question. Thus, to find out more, in publication II, we conducted semi-structured interviews with local entertainment technology companies. For those interviews, we created personas of graduated students based on data from our current graduated students who had made video games as their theses. The purpose of these was to get the companies to imagine a tangible situation and give feedback on what sort of data regarding their job candidates they need and value.

With publication II, we gained a lot of valuable insight into how the graduate needs to be prepared prior to applying for a job in the local entertainment technology industry. Many companies look for thorough portfolios where they can actually play the applicant's created video games. Just a picture and a description are usually not enough. This supports our design that the graduate must have a portfolio, and the courses or activities throughout the curriculum should provide input for that portfolio.

Another important result from publication II was that game designers and developers are often expected to embody managerial roles, be able to work well in teams, and have a breadth of skills or at least contextual fluencies. There were a few companies where this was not necessary – i.e., a developer there could just be skilled in programming and know about nothing else – but in most companies, that was not so. Via these interviews, it became clear that the local industry is varied, and the companies have different expectations. Thus, prior education needs to provide a strong and versatile enough foundation.

With the insight from publications I and II, we sought to answer the third research question and decided to explore the types of Bachelor curricula already existing in other countries. Such a survey was done by Ip about a decade ago [23]. We felt that during the last decade, there might have been notable changes in these programs and that the classification of programs for our purposes could be different. Thus, we decided to conduct a similar survey ourselves in publication III. Due to the standard created through the Bologna Process, we chose to focus on European programs. Out of convenience, we picked the ones in English that had public data available.

Performing cluster analysis on the 113 curricula, we were able to show the existence of three clusters. These were the curricula that 1) focus on game programming, 2) focus on game art, 3) deal with game design and development and are broader than the other two. Performing principal component analysis on the data showed that many curricula that follow the *Industry* and *Design* loadings (mostly the *Game Design and Development* programs) lie between the curricula with *Development* (mostly the *Game Programming* programs) and *Art* (mostly the *Game Art* programs) loadings. The interpretation of that is that game design and development education, at least in part, serves as an important connection between the specialized game programming and game art fields.

Based on the results from these three research endeavors and through iterative design with both our educators and the industry, covering our fourth research question, we designed a 180 ECTS credits Video Game Design and Development Bachelor's program. Through the design, we found and relied on five guiding principles. These could be summarized as follows: 1) a video game program should be about video games, 2) the needed interdisciplinary contextual fluencies must be included, 3) there must be many touchpoints between the program specializations, 4) course projects should be explicitly valuable after the course, i.e., in a portfolio, and 5) the program should also give a solid foundation for video games as cultural and financial objects.

We believe the research results of this dissertation will be useful for curriculum designers and education researchers who work with existing or new video game curricula. Of course, it must be understood that the research in this dissertation is limited by both the sample size and the locality. In Publications I and II we focused on the Estonian industry. The industry in some other region might be different. Further studies with larger sets of data, more thorough methodologies, and based on implementation of the designed curriculum can provide additional insight into various aspects described in this dissertation.

BIBLIOGRAPHY

[1] Ernest Adams. 2013. *Fundamentals of Game Design* (3rd edition ed.). New Riders, Berkeley, California.

[2] A. Andrade. 2015. Game engines: a survey. *EAI Endorsed Trans. Game-Based Learn.* vol 2, (November 2015), e8. DOI:https://doi.org/10.4108/eai.5-11-2015.150615

[3] Anonymous. 2022. shipped.ee. Retrieved June 8, 2022 from http://shipped.ee/

[4] L. Argent, B. Depper, R. Fajardo, S. Gjertson, S.T. Leutenegger, M.A. Lopez, and J. Rutenbeck. 2006. Building a game development program. *Computer* 39, 6 (June 2006), 52–60. DOI:https://doi.org/10.1109/MC.2006.189

[5] Atari Inc. 1980. *Adventure Game Program Instructions*.

[6] Jessica D. Bayliss and Kevin Bierre. 2008. Game design and development students: who are they? In *Proceedings of the 3rd international conference on Game development in computer science education* (GDCSE '08), Association for Computing Machinery, New York, NY, USA, 6–10. DOI:https://doi.org/10.1145/1463673.1463675

[7] Jayson Chi and Terry Sun. Development drivers: Third-party engines and mobile gaming.

[8] Ron Coleman, Mary Krembs, Alan Labouseur, and Jim Weir. 2005. Game design & programming concentration within the computer science curriculum. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education* (SIGCSE '05), Association for Computing Machinery, New York, NY, USA, 545–550. DOI:https://doi.org/10.1145/1047344.1047514

[9] Computer Graphics and Virtual Reality Lab [CGVR Lab]. 2022. Companies. *The CGVR Lab*. Retrieved June 8, 2022 from https://cgvr.cs.ut.ee/wp/index.php/companies/

[10] André Czauderna. 2018. Academic Game Design Education: A Comparative Perspective. In *Serious Games* (Lecture Notes in Computer Science), Springer International Publishing, Cham, 9–12. DOI:https://doi.org/10.1007/978-3-030-02762-9_2

[11] George Dinwiddie. 2009. Feel The Burn, Getting the Most out of Burn Charts. *Better Software 11*, 26–31.

[12] Eesti Konjunktuuriinstituut. 2022. *Eesti loomemajanduse olukorra uuring ja kaardistus*. Talinn. Retrieved from https://eas.ee/uuring-kultuur-hoogustab-aina-enam-majandust-ning-panustab-rahvusvahelistumisse-ja-eksporti/

[13] Anthony Estey, Amy Gooch, and Bruce Gooch. 2009. Addressing industry issues in a multi-disciplinary course on game design. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (FDG '09), Association for Computing Machinery, New York, NY, USA, 71–78. DOI:https://doi.org/10.1145/1536513.1536534

[14] Estonian Virtual and Augmented Reality Association [EEVR]. 2022. Estonian Virtual and Augmented Reality Association. *EEVR*. Retrieved June 8, 2022 from https://eevr.ee/

[15] Nuno Fachada and Nélio Códices. 2020. Top-down Design of a CS Curriculum for a Computer Games BA. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (ITiCSE '20), Association for Computing Machinery, New York, NY, USA, 300–306. DOI:https://doi.org/10.1145/3341525.3387378

[16] Suzanne Freyjadis. 2019. (27) A New Kind of Video Game Curriculum Framework | LinkedIn. Retrieved November 2, 2023 from https://www.linkedin.com/pulse/new-kind-video-game-curriculum-framework-suzanne-freyjadis/?articleId=6579470022060634112
[17] GameDev Estonia. 2022. Database. *GameDev Estonia*. Retrieved June 8, 2022 from https://gamedevestonia.ee/location/
[18] GoodFirms. 2022. Top Game Development Companies in Estonia 2022 | GoodFirms. Retrieved June 8, 2022 from https://www.goodfirms.co/game-development-companies/estonia
[19] Jason Gregory. 2018. *Game Engine Architecture, Third Edition* (3rd edition ed.). A K Peters/CRC Press, Boca Raton.
[20] Johan Huizinga. 2016. *Homo Ludens: A Study of the Play-Element in Culture*. Angelico Press, Kettering, OH.
[21] Robin Hunicke, Marc Leblanc, and Robert Zubek. 2004. MDA: A Formal Approach to Game Design and Game Research. *AAAI Workshop - Tech. Rep.* 1, (January 2004).
[22] IGDA. 2008. IGDA Curriculum Framework 2008. Retrieved June 8, 2022 from https://docplayer.net/1868720-Igda-curriculum-framework.html
[23] Barry Ip. 2012. Fitting the Needs of an Industry: An Examination of Games Design, Development, and Art Courses in the UK. *ACM Trans. Comput. Educ.* 12, 2 (April 2012), 6:1-6:35. DOI:https://doi.org/10.1145/2160547.2160549
[24] Jaan Janno. 2017. Sissejuhatav mänguarenduse ja programmeerimise kursuse õppematerjal. Thesis. Retrieved October 13, 2023 from https://dspace.ut.ee/handle/10062/65857
[25] Jussi Kasurinen, Saeed Mirzaeifar, and Uolevi Nikula. 2013. Computer science students making games: a study on skill gaps and requirement. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* (Koli Calling '13), Association for Computing Machinery, New York, NY, USA, 33–41. DOI:https://doi.org/10.1145/2526968.2526972
[26] Mark Kendall. 2021. 33 Top Gaming Startups and Companies in Estonia (2021) – BestStartup.eu. Retrieved June 8, 2022 from https://beststartup.eu/33-top-gaming-startups-and-companies-in-estonia-2021/
[27] K. L. Krithika. 2023. Unity is Not as United as It Sounds. *Analytics India Magazine*. Retrieved October 6, 2023 from https://analyticsindiamag.com/unity-is-not-as-united-as-it-sounds/
[28] Tiina Kull. 2012. Teeme ise arvutimänge - algus. (May 2012). Retrieved October 13, 2023 from https://dspace.ut.ee/handle/10062/25840
[29] David S. Kushner and David Kushner. 2003. *Masters of doom: how two guys created an empire and transformed pop culture* (First edition ed.). Random House, New York.
[30] Scott Leutenegger and Jeffrey Edgington. 2007. A games first approach to teaching introductory programming. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education* (SIGCSE '07), Association for Computing Machinery, New York, NY, USA, 115–118. DOI:https://doi.org/10.1145/1227310.1227352
[31] Joe Linhoff and Amber Settle. 2008. Teaching game programming using XNA. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education* (ITiCSE '08), Association for Computing

Machinery, New York, NY, USA, 250–254. DOI:https://doi.org/10.1145/1384271.1384338
[32] Morgan Luck. 2009. The gamer's dilemma: An analysis of the arguments for the moral distinction between virtual murder and virtual paedophilia. *Ethics Inf. Technol.* 11, (March 2009), 31–36. DOI:https://doi.org/10.1007/s10676-008-9168-4
[33] Margus Luik. 2016. Kursus "Programmeerimismustrid arvutimängudes." Thesis. Retrieved October 13, 2023 from https://dspace.ut.ee/handle/10062/56164
[34] Michael Mateas and Jim Whitehead. 2007. Design Issues for Undergraduate Game-Oriented Degrees. *GDCSE'07* (January 2007). Retrieved October 23, 2022 from https://www.academia.edu/2806268/Design_issues_for_undergraduate_game_oriented_degrees
[35] Bruce McCarthy. 2018. *Product Manager vs. Project Manager*. O'Reilly Media, Inc. Retrieved October 8, 2023 from https://learning.oreilly.com/library/view/product-manager-vs/9781492034452/
[36] Monica M. McGill. 2008. Critical skills for game developers: an analysis of skills sought by industry. 89–96. DOI:https://doi.org/10.1145/1496984.1497000
[37] Monica M. McGill. 2009. Defining the expectation gap: a comparison of industry needs and existing game development curriculum. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (FDG '09), Association for Computing Machinery, New York, NY, USA, 129–136. DOI:https://doi.org/10.1145/1536513.1536542
[38] Koji Mikami, Taichi Watanabe, Katsunori Yamaji, Kenji Ozawa, Akinori Ito, Motonobu Kawashima, Ryota Takeuchi, Kunio Kondo, and Mitsuru Kaneko. 2010. Construction trial of a practical education curriculum for game development by industry–university collaboration in Japan. *Comput. Graph.* 34, 6 (December 2010), 791–799. DOI:https://doi.org/10.1016/j.cag.2010.09.015
[39] Mark Muhhin. 2018. Ülesannete komplekt ainele „Teeme ise arvutimänge - algus". Thesis. Retrieved October 13, 2023 from https://dspace.ut.ee/handle/10062/66011
[40] Mark Muhhin. 2021. Videomängude arenguloo e-kursus. University of Tartu, Tartu. Retrieved from https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=72556&year=2021&language=en
[41] Alan Pipes. 1999. *FOUNDATIONS OF ART AND DESIGN*. LAURENCE KING, London.
[42] Colin B. Price, John Colvin, and Warren Wright. 2006. Introducing Game Development into the Computing Curriculum — A Progressive Methodology. *Innov. Teach. Learn. Inf. Comput. Sci.* 5, 3 (October 2006), 1–10. DOI:https://doi.org/10.11120/ital.2006.05030004
[43] Timothy E. Roden and Rob LeGrand. 2013. Growing a computer science program with a focus on game development. In *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13), Association for Computing Machinery, New York, NY, USA, 555–560. DOI:https://doi.org/10.1145/2445196.2445362
[44] Petri Saarikoski and Jaakko Suominen. 2009. Computer Hobbyists and the Gaming Industry in Finland. *IEEE Ann. Hist. Comput.* 31, 3 (July 2009), 20–33. DOI:https://doi.org/10.1109/MAHC.2009.39

[45] Jesse Schell. 2008. *The Art of Game Design: A book of lenses*. CRC Press.
[46] Suzi Stephenson. 2019. TIGA Survey Reveals that Unity 3D Engine Dominates the UK Third Party Engine Market. *TIGA*. Retrieved October 6, 2023 from https://tiga.org/news/tiga-survey-reveals-that-unity-3d-engine-dominates-the-uk-third-party-engine-market
[47] Keith Stuart. 2023. Game developers furious as Unity Engine announces new fees. *The Guardian*. Retrieved October 6, 2023 from https://www.theguardian.com/games/2023/sep/12/unity-engine-fees-backlash-response
[48] Alexander Styhre and Björn Remneland-Wikhamn. 2021. The video game as agencement and the image of new gaming experiences: the work of indie video game developers. *Cult. Organ.* 27, 6 (November 2021), 476–489. DOI:https://doi.org/10.1080/14759551.2021.1919893
[49] tainask. 2022. Estonian GameDev Summer Summit. *LAB Focus*. Retrieved October 7, 2023 from https://blogit.lab.fi/labfocus/en/estonian-gamedev-summer-summit/
[50] Raimond-Hendrik Tunnel. 2015. Arvutigraafika õppematerjal. Thesis. Retrieved October 13, 2023 from https://dspace.ut.ee/handle/10062/56053
[51] Dana Vrajitoru and Paul Toprac. 2016. *Games Programming in Computer Science Education*.
[52] Omri Wallach. 2020. 50 Years of Gaming History, by Revenue Stream (1970-2020). *Visual Capitalist*. Retrieved October 6, 2023 from https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/
[53] WiMi5. 2015. Pros and cons of using a third party game engine. *Medium*. Retrieved October 6, 2023 from https://medium.com/@WiMi5/pros-and-cons-of-using-a-third-party-game-engine-41c88f69261c
[54] Mark Wolf. 2001. Genre and the Video Game. *Medium Video Game* (January 2001). Retrieved December 24, 2022 from https://www.academia.edu/435740/Genre_and_the_Video_Game
[55] Fan Zhang, David Kaufman, and Simon Fraser. 2014. USING VIDEO GAMES IN COMPUTER SCIENCEEDUCATION. (2014).
[56] 2010. And Then There Was Apple. *Apple II History*. Retrieved October 6, 2023 from https://www.apple2history.org/museum-a-l/articles/ca8610/
[57] 2018. *Game a Week: Teaching Students to Prototype*. Retrieved October 7, 2023 from https://www.youtube.com/watch?v=9O9Q8OVWrFA
[58] 2021. *MängudeÖÖ Optimist/Digital loengusaal – RAIMOND TUNNEL: ARVUTIMÄNGUDE HARIDUS TARTU ÜLIKOOLIS*. Retrieved October 7, 2023 from https://www.youtube.com/watch?v=K8JJ5boaOr8
[59] 2023. OTSEÜLEKANNE ⟩ MängudeÖÖ peasaal toob vaatajateni e-sporti, muusikat ning idamaist kultuuri. *E-sport*. Retrieved October 7, 2023 from https://tehnika.postimees.ee/7866059/otseulekanne-mangudeoo-peasaal-toob-vaatajateni-e-sporti-muusikat-ning-idamaist-kultuuri
[60] 2023. List of video game genres. *Wikipedia*. Retrieved October 6, 2023 from https://en.wikipedia.org/w/index.php?title=List_of_video_game_genres&oldid=1178254147
[61] 2023. Creation Engine. *Fallout Wiki*. Retrieved October 6, 2023 from https://fallout.fandom.com/wiki/Creation_Engine
[62] *History of Digital Games*. Retrieved October 6, 2023 from https://learning.oreilly.com/library/view/history-of-digital/9781317503804/

[63] Doom clones. *Doom Wiki*. Retrieved October 6, 2023 from https://doom.fandom.com/wiki/Doom_clones
[64] Crytek announces its Game Engine CryENGINE. Retrieved October 6, 2023 from https://www.crytek.com/news/crytek-announces-its-game-engine-cryengine
[65] Source - Valve Developer Community. Retrieved October 6, 2023 from https://developer.valvesoftware.com/wiki/Source
[66] Games Programs in Community Colleges & Universities – HEVGA. Retrieved October 8, 2023 from https://hevga.org/placeholder-games-programs/
[67] Computer Game Design and Graphics - Ekonomikas un Kultūras augstskola. Retrieved October 7, 2023 from https://www.augstskola.lv/?parent=340&lng=eng
[68] *Processing: Creative Coding and Generative Art in Processing 2*. Retrieved October 13, 2023 from https://learning.oreilly.com/library/view/processing-creative-coding/9781430244646/

APPENDIX A – THE INDUSTRY NEEDS QUESTIONNAIRE

Video Game Designer-Developer Curriculum Needs Survey

Dear Estonian video game development company,

In the University of Tartu we are currently deliberating on a new Bachelor level Video Game Designer-Developer curriculum. Our aim is to provide a solid skillset and education in the field of video games for new generations of video game makers. We envision a multi-disciplinary curriculum, where studying designers and developers can both specialize and work together in learning and creating video games.

Prominent principle for the curriculum is to ensure our graduates find employment on the VIDEO GAME DESIGNER / DEVELOPER positions in Estonian video game companies. For that we ask you for what are your expectations regarding YOUR JOB CANDIDATES for the said positions – what skills, fluency, knowledge are you expecting of graduates who would love to come work for you. Please take this time to fill this questionnaire to the best of your ability so that the new curriculum would benefit you the most.

We would also be very happy to interview you in the future. If you are also interested in an interview, then we will get back to you with that and other collaboration possibilities after analyzing the questionnaire results.

By answering this questionnaire you agree that your entered information will be used in the design of the video game designer-developer curriculum and academic research.

The analysis and summary of the results will be made freely available to the participants.

Kind regards,
Raimond Tunnel
https://cgvr.cs.ut.ee/

Email

_____

Company Name

_____

Number of Current Employees

_____

Fields of Activity

☐ Arcade / Casino Games
☐ Computer Graphics
☐ Console Games
☐ Mobile Games
☐ PC Games
☐ Virtual Activity

Curriculum Expectations

As students go through their studies, the knowledge and skills they learn need to be beneficial for them finding employment and work in the industry. Thus, in this section we ask you about the proficiencies important for the video game designers-developers in your company.

This will be the actionable information we use to design the curriculum and the courses in it.

What are your expectations for a Bachelor level Video Game Designer-Developer curriculum and its graduates?

_____
_____
_____

The following Knowledge Areas are important in your candidates:
These are areas the candidate for your Video Game Designer or Video Game Developer position should have skills and knowledge in. "Strongly disagree" means that it is not important at all for the candidate.

	1 – Strongly Disagree	2 – Disagree	3 – Neutral	4 – Agree	5 – Strongly Agree
Algorithm Development					
Artificial Intelligence					
Code Development					
Computer Organization					
Data Structures					
Game Development Processes					
Graphics					
Large Scale Development Processes					
Low-Level / Embedded Systems					
Math					
MMO Programming					
Multi-Thread Programming					
Networking					
Object-Oriented Programming					
Operating Systems					
Optimization					
Physics					
Real-Time Systems					
Relational Databases					

Software Development Processes					
Software Testing					
Tool Development					
User Interface					
Version Control Processes					
Video Game Analysis					
Video Game Design					
Web Development					

Is there any other Knowledge Area you consider important?
What? How important?

_____
_____
_____

The following Languages are important in your candidates:
In candidates for your Video Game Designer or Video Game Developer position. “Strongly disagree” means that it is not important at all for the candidate.

	1 – Strongly Disagree	2 – Disagree	3 – Neutral	4 – Agree	5 – Strongly Agree
Assembly					
C					
C#					
C++					
CSS					
GLSL / HLSL					
Go					
HTML					
Java					

JavaScript					
JSON					
Lua					
Perl					
PHP					
Python					
Rust					
SQL					
TypeScript					
XML					
Visual languages (eg, Blueprints in Unreal Engine, Visual Scripting in Unity)					

Is there any other Language you consider important?
What? How important?

The following Software Tools / Environments are important in your candidates:
In candidates for your Video Game Designer or Video Game Developer position. “Strongly disagree” means that it is not important at all for the candidate.

	1 – Strongly Disagree	2 – Disagree	3 – Neutral	4 – Agree	5 – Strongly Agree
3D content creation tool (Blender, Maya, Houdini, 3DS Max, Modo)					
Audio editor (Audacity, ...)					
Database server (SQL, NoSQL)					
Digital audio workstation (Ableton Live, Audition, FL Studio, REAPER)					

Game engine (Unity, Unreal Engine, Godot, GameMaker Studio, Open 3D Engine, CryEngine, Construct 3, Source 2)					
Graphics API (DirectX, OpenGL, Vulkan, WebGL)					
IDE (Visual Studio, Visual Studio Code, Monodevelop, Rider)					
Project management tools (Confluence, Jira, Trello)					
Raster graphics editor (Photoshop, PaintShop Pro, Affinity Photo, Krita, GIMP)					
Software runtime (.NET Framework / Mono, Java, Node.js)					
Texturing tools (Substance 3D, Material Maker, ArmorPaint, Quixel Mixer)					
Vector graphics editor (Affinity Designer, Adobe Illustrator, Inkscape)					
Version control technology (Perforce, Git, SVN, Mercurial, Unity Collab, Plastic SCM, Dropbox, Google Drive)					
Video editor (DaVinci Resolve, Adobe Premiere, OpenShot, Kdenlive)					
Visual effects tools (After Effects, Nuke, DaVinci Resolve)					

Which are the main 3D Content Creation Tools in the company?

☐ **3ds Max**	☐ **Blender**	☐ **Houdini**
☐ **Maya**	☐ **Marvelous Designer**	☐ **Modo**
☐ **We do not use 3D content creation tools.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which is the main Audio Editor in the company?

☐ **3ds Max**		
☐ **We do not use audio editors.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which is the main Digital Audio Workstation (DAW) in the company?

☐ **Abelton Live**	☐ **Audition**	☐ **FL Studio**	☐ **REAPER**
☐ **We do not use 3D content creation tools.**			☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Game Engines in the company?

☐ **Construct 3**	☐ **CryEngine**	☐ **GameMaker Studio**
☐ **Godot**	☐ **Open 3D Engine (ex Lumberyard)**	☐ **Source 2**
☐ **Unity**	☐ **Unreal Engine**	
☐ **We do not use game engines.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Graphics APIs in the company?

☐ **DirectX**	☐ **OpenGL**	☐ **Vulkan**
☐ **WebGL**		
☐ **We do not use graphics APIs.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

Which are the main IDEs in the company?

☐ **Monodevelop**	☐ **Rider**	☐ **Visual Studio**
☐ **Visual Studio Code**		
☐ **We do not use IDEs.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Project Management Tools in the company?

☐ **Confluence**	☐ **Jira**	☐ **Trello**
☐ **We do not use project management tools.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.
☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Raster Graphics Editors in the company?

☐ **Affinity Photo**	☐ **GIMP**	☐ **PaintShop Pro**
☐ **Photoshop**	☐ **Krita**	
☐ **We do not use raster graphics editors.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.
☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Software Runtimes in the company?

 ☐ **.NET / Mono**	 ☐ **Java**	 ☐ **Node.js**
☐ **We do not use software runtimes.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

Which are the main Texturing Tools in the company?

 ☐ **ArmorPaint**	 ☐ **Material Maker**	 ☐ **Quixel Mixer**
 ☐ **Substance 3D**		
☐ **We do not use texturing tools.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

Which are the main Vector Graphics Editors in the company?

 ☐ **Affinity Designer**	 ☐ **Illustrator**	 ☐ **Inkscape**
 ☐ **Substance 3D**		
☐ **We do not use vector graphics editors.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.
☐ It is OK if they have skills in alternative tools.
☐ They need to have skills in these specific tools.

Which are the main Version Control Technologies in the company?

 ☐ **Dropbox**	 ☐ **Git**	 ☐ **Google Drive**
 ☐ **Mercurial**	 ☐ **SVN**	 ☐ **Perforce**
 ☐ **Plastic SCM**	 ☐ **Unity Collab**	
☐ **We do not use vector graphics editors.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

Which are the main Video Editors in the company?

 ☐ **DaVinci Resolve**	 ☐ **Kdenlive**	 ☐ **Premiere**
 ☐ **OpenShot**		
☐ **We do not use video editors.**		☐ **Other: _____**

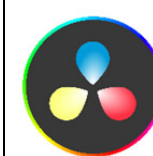
How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

Which are the main Visual Effects Tools in the company?

 ☐ **After Effects**	 ☐ **Nuke**	 ☐ **DaVinci Resolve**
☐ **We do not use visual effects tools.**		☐ **Other: _____**

How important are the skills in these specific tools for a candidate?
For a candidate to your Video Game Designer or Video Game Developer position.

☐ It is OK if they do not have skills in these or alternative tools.

☐ It is OK if they have skills in alternative tools.

☐ They need to have skills in these specific tools.

The following Abilities are important in your candidates:
In candidates for your Video Game Designer or Video Game Developer position. “Strongly disagree“ means that it is not important at all for the candidate.

	1 – Strongly Disagree	2 – Disagree	3 – Neutral	4 – Agree	5 – Strongly Agree
Attitude / Disposition					
Communication Skills					
Interpersonal Skills					
Leadership					
Organization / Time Management					
Problem Solving Abilities					
Work Ethic					

Is there any other Ability you consider important?
What? How important?

_____
_____
_____

The following Contextual Fluencies are important in your candidates:
These are areas the candidate for your Video Game Designer or Video Game Developer position should be able to think in terms of and adequately discuss about. “Strongly disagree” means that it is not important at all for the candidate.

	1 – Strongly Disagree	2 – Disagree	3 – Neutral	4 – Agree	5 – Strongly Agree
2D Modelling					
3D Modelling					
Audio (Sound Effects)					
Color Theory					
Creative Writing					
Digital Animation					
Digital Art					
Digital Photography					
Drawing					
Music					
Painting					
Sculpture					
Storytelling					
Theatre Arts					

Is there any other Contextual Fluency you consider important?
What? How important?

_____
_____
_____

Collaboration

When educating new generations of video game designers-developers, your collaboration will be most needed. It is important the students get real-world practical experience already during their studies through internships or other activities in collaboration with you. Most importantly, that they would have an option to apply for a video game designer-developer job after graduation.

Would you potentially have internship positions for 2nd and 3rd year students?
Can be paid or unpaid.
☐ Yes.
☐ No.

Would you be interested in preparing and conducting some study in the form of academic lectures, seminars, or lab sessions to the students yourself?
☐ Yes.
☐ No.

How many graduates would you expect to employ annually?
Assuming they meet your criteria.
_____

Would you be interested in an interview?

☐ Yes.

☐ No.

Other interesting collaboration ideas you have:

_____

_____

_____

Last thoughts

Anything else you would like to generally add that did not have a question for.

_____

_____

_____

ACKNOWLEDGEMENTS

I am grateful for all the numerous local entertainment technology companies that participated in my research and gave valuable insight into what would they need from a Video Game Design and Development Bachelor's program.

Thank you: Cult Software OÜ, Hammer&Ravens OÜ, Interactive Fate OÜ, Cydonian Gameworks OÜ, Placeholder Gameworks OÜ, Creative Mobile OÜ, chaosmonger studio OÜ, FutuClass OÜ, OÜ BlueRay, Maru VR Productions OÜ, Mobi Lab OÜ, FROST FX OÜ, Wolfprint 3D OÜ, ExteriorBox OÜ, OneEyeAnt OÜ, Arcweave OÜ, Frozen Kingdom OÜ, Friedegggames OÜ, 3D Technologies R&D AS, Artineering OÜ, Tall Troll Games OÜ, Motivum Games OÜ, Lifelong Fund OÜ, Joyixir OÜ, ALPA Kids OÜ, Banished OÜ, Clickwork Games OÜ, StoryBox Games OÜ, and Simple Magic OÜ!

Thanks also to the many educators and colleagues in the University of Tartu, who helped design the curriculum and were happy to incorporate their existing courses into such a program.

Thank you: Henri Hanson, Ulrich Norbisrath, Jaanus Jaggo, Ats Kurvet, Daniel Nael, Mark Muhhin, Mathias Plans, Indrek Zolk, Egon Elbre, Mark Fišel, Risko Ruus, Alex Nolte, Anu Aunapuu, Juko-Mart Kõlar, Sven Aller, Marie-Luise Meier, Taavi Kivisik, Egon Elbre, Varmo Vene, Helle Hein, Pelle Jakovits, Eero Vainikko, Jakob Mass, Jaan Janno, and Piret Orav!

Special thanks to Tartu Art School, for being supportive in offering to teach the basic art skills and contextual fluency courses in the proposed program and for the potential student collaboration during game jams and the capstone project!

Big thanks go out to all the students who have over the years asked me to design a larger video games educational program than just a couple of elective courses. There are too many of you to count, but if we have ever talked about video games or education, then I mean you! Without your desire and passion to learn more about video games, none of this would have happened!

Thanks to Prof. Jessica Bayliss from the Rochester Institute of Technology and Assoc. Prof. Martin Sillaots from the University of Tallinn for discussing the topic with me and providing insight into the challenges and solutions in their own Game Design and Development programs.

Lastly, I thank my supervisor Ulrich Norbisrath. Thank you for helping me navigate through the world of science and giving me a very clear incentive and motivation for completing a PhD thesis!

This research was partially funded by the European Social Fund.

SUMMARY IN ESTONIAN

Videomängude disaini ja arenduse bakalaureuseõppekava Eestile

Videomängude disaini ja arenduse õpetamine kõrgharidusasutustes on mitmekülgne probleem. Videomängud on loomult erialadevahelised ning arendamise ökosüsteem on kiiresti arenev ja muutuv. Seega vastava kõrghariduse õppekava loomisel on vaja teha keerulisi otsuseid. Näiteks tuleb otsustada, kui suur osakaal õppekavas peaks olema disainil ja kui suur arendusel? Kuidas on nii disain kui arendus omavahel seotud? Milliseid konkreetseid oskusi ning teadmisi on vaja disaini ja milliseid arenduse poolelt? Milliseid programmeerimiskeeli ja mängumootoreid tuleks õpetada? Kuidas tagada, et õppekava lõpetajad oleksid piisavalt pädevad pärast lõpetamist erialaselt töötama? Mis roll tuleb võtta lõpetajatel tulevikus videomängude loomisel seotud teiste erialade (nt programmeerimine ja kunst) esindajatega?

Käesolev lõputöö annab vastust nendele ja teistele küsimustele Eesti riigi kohaliku meelelahutustarkvara sektori kontekstis. Nimelt Eesti riigi iduettevõtete-sõbralik poliitika ja tehnoloogiline tase soosivad videomängude loomise ettevõtete teket ning arengut. Eesti suurimas kõrgharidusasutuses, Tartu Ülikoolis, on mõned videomängude-teemalised populaarsed ainekursused. Samas Eesti Konjunktuuriinstituudi läbiviidud 2022. aasta uuring näitab, et meelelahutustarkvara tööstuse arengut pärsib piisavalt andeka ja oskusliku tööjõu puudumine. Käesoleva doktoritöö motivatsiooniks oli selgitada välja, millised puudujäägid, vajadused ja ootused Eesti videomängutööstuses on, millised on teistes Euroopa riikides videomängude bakalaureuseõppekavad ning töötada välja Tartu Ülikoolis potentsiaalselt realiseeritav ja tööstuse vajadustele vastav õppekava.

Doktoritöö algab Tartu Ülikoolis oleva praeguse olukorra kirjeldusega. Antakse kiire ülevaade praegustest ainekursustest ja nende populaarsusest. Töö jätkub kolme publikatsiooniga, mis uurisid Eesti meelelahutustarkvara sektori vajadusi ning Euroopa videomängude bakalaureuseõppekavu.

Esimeses publikatsioonis tegime kvantitatiivse uuringu. See põhines 2009. aastal Monica M. McGilli poolt tehtud tööl. Meie uuringule vastasid 28 Eesti meelelahutustarkvara ettevõtet, kes hindasid erinevate oskuste vajadust Likerti skaalal ja viies kategoorias. Küsiti teadmiste, programmeerimiskeelte, tarkvara, oskuste ja kontekstipõhise suhtlusoskuse kohta. Teadmiste osas olid tulemused üldiselt kooskõlas McGilli uuringuga. Meie uuringus leiti, et tarkvara optimeerimine on uuritud ettevõtetes varasemast statistiliselt rohkem nõutud. Samuti on palju nõutud videomängude disain, mida McGilli uuring ei käsitlenud. Programmeerimiskeelte osas leidsime, et kõige nõutumad keeled on C#, visuaalsed keeled, C++ ja JSON. JavaScripti ja C# keelte nõudlus on statistiliselt oluliselt tõusnud ning keelte C ning XML nõudlus statistiliselt oluliselt vähenenud. Oskuste kategoorias on meie uuringus statistiliselt rohkem nõutud

ajaplaneerimise oskus. Kontekstipõhise suhtlusoskuse kategoorias olid sarnaselt McGilli uuringule kõige nõutumad erialased suhtlusoskused 3D modelleerimise, digitaalses kunsti ning digitaalses animatsiooni valdkondades.

Uurisime esimeses publikatsioonis ka konkreetsete tööriistade kasutust ja ootusi Eesti meelelahutustarkvara ettevõtetes. Vastajatel paluti märkida, mis tööriistu 14-s kategoorias nad kasutavad ning kas nende ettevõttesse tööle kandideerijal on vaja osata neidsamu tööriistu või sobiksid ka oskused mõne alternatiivse tööriistaga. Tulemustest leidsime, et enamik Eesti ettevõtteid ootab spetsiifiliselt Unreal Engine ja Unity mängumootorite oskust. Alternatiivid nende kahe tööriista osas üldiselt ei sobinud. Mitmes ettevõttes oodatakse oskust mõlemas mängumootoris. Sellest tulemusest järeldub, et videomängude disaini ja arenduse õppekava ei tohiks keskenduda ainult ühele mängumootorile. Lisaks Unreal Engine ja Unity mootorite õpetamisele oleks mõistlik õpiväljund võimekus omandada tulevikus uusi tehnoloogiaid ja mängumootoreid.

Teises publikatsioonis viisime läbi kvalitatiivse uuringu. Intervjueerisime 11 eelnevale uuringule vastanud ettevõtet, kes olid nõus meie uuringus jätkama. Intervjuud koosnesid kahest uuritavast osast. Esimeses osas vestlesime ettevõtte esindajaga pool-struktureeritud formaadis ning küsisime küsimusi ettevõtete omapära, töörollide, tööülesannete ja ootuste kohta tööle kandideerijatele. Küsisime veel, mida ettevõtte meelest peaks ülikoolis õpetama, et lõpetajad nende juures tööle saaksid. Avastasime, et Eesti ettevõtted on oma ootustelt mitmekülgsed. Samas läbivalt oodati, et videomängude disainer või arendaja oleks piisavalt laia taustaga, et juhtida projekti, suhelda erialaselt erineva taustaga kolleegidega ning mõista mängu kasutajakogemust. Ühes ettevõttes oli hiljuti nimetatud mängudisaineri ametikoht ümber projektijuhiks. Paljudes uuritud ettevõtetes oodati, et ka spetsialiseerunud töötajad mõistaks tervikpilti. Ühest sellisest ettevõttest öeldi isegi: „Programmeerija, kes ei tea midagi kasutajakogemusest, on halb programmeerija.“ Samas leidus ka erandeid. Paar ettevõtet vastasid, et nendel on tööl programmeerija, kes tegeleb ainult programmeerimisega, muud temalt ei eeldata ning nende ettevõttes selline korraldus töötab. Nendest tulemustest selgub, et videomängude disaini ja arenduse õppekava peaks olema võrdlemisi laiapõhjaline ning juba varakult tuleb õppetöös käsitleda mitmekülgseid ettevõtteid ning nende erinevaid ootusi.

Intervjuude teine osa põhines kolmel bakalaureuse lõpetanud üliõpilase persoonal. Need kolm persoonat lõime oma varasemalt lõpetanud üliõpilastel läbiviidud küsitlusel ning lõputööde tegemise jooksul kogutud andmetel. Intervjuus esitati persoonad ettevõttele ja küsiti, mida teeks ettevõte selliste tööle kandideerivate inimeste infoga edasi. Ettevõtete vastused erinesid – mõni kutsuks kõik edasi töövestlusele, mõned valisid erinevate tingimuste järgi teatud kandidaadid välja, mõni ei soovinud ühtegi kutsuda. Levinum kommentaar oli, et ettevõttel oleks otsuse tegemiseks vaja näha rohkem kandidaadi tehtud mänge ja neid ka ise mängida. Persoonalehel olev staatiline pilt mängust ja kirjeldus ei olnud piisavad. Sellest lähtuvalt leidsime, et portfoolio on videomängude disaini ja arenduse õppekaval tulevase töö leidmise jaoks väga oluline.

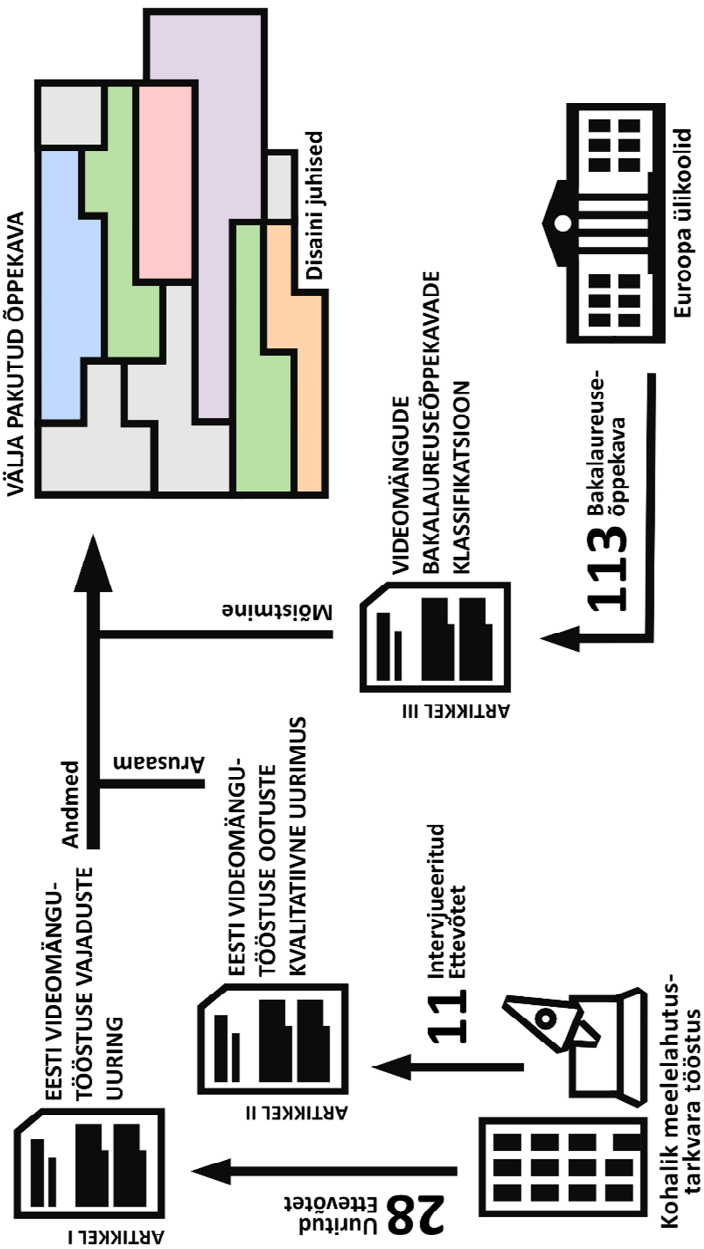
Kolmandas publikatsioonis uurisime Euroopa kõrgkoolides loetavaid videomängude bakalaureuseõppekavu. Valisime Euroopa õppekavad, sest Bologna protsessi ühtlustuse tõttu on nende omavaheline võrdlus selgem. Lihtsuse huvides võtsime uuringusse õppekavad, millel oli avalikult kättesaadav ja piisavalt põhjalik ingliskeelne sisuülevaade. Pärast õppekavade otsingut ja filtreerimist jäi uuringusse 113 sobivat videomängude bakalaureuseõppekava. Meie leitud õppekavades loetletud ained sildistasime 11 klassifikaatoriga, mis põhinesid nii IGDA (International Game Developers Association) õppekava raamistiku teemadel kui ka 2012. aastal Barry Ipi poolt tehtud sarnases uuringus kasutatud kategooriatel. Tööstus, disain, arendus ja matemaatika on neli näidet nendest klassifikaatoritest. Kaaludes igat aine küljes olevat silti vastava aine ainepunktide arvuga, lõime iga õppekava profiili. See profiil on 11-elemendiline vektor, mille iga element vastab ühele klassifikaatorile ja sisaldab protsenti, kui palju selle klassifikaatoriga ja ainepunktidega kaalutud õppeained tervest õppekavast moodustavad.

Pärast õppekavade profiilide leidmist, tegime õppekavadel hierarhilise klasteranalüüsi. Meie eeldus oli, et õppekavad jagunevad kolme klastrisse: mängude programmeerimine, mängude kunst ning mängude disain ja arendus. Eeldus põhines Barry Ipi uuringus, kus olid õppekavad sarnase kolme teema järgi eraldatud. Klasteranalüüsi tulemusena saime leida, kui suured osakaalud nendes kolmes klastris on erinevate klassifikaatoriga ainetel. Näiteks leidsime, et mängude programmeerimise õppekavades moodustavad suure osa arenduse klassifikaatoriga ained, maksimaalselt oli see ühes õppekavas koguni 75%. Uurisime ja leidsime veel, et videomängude programmeerimise õppekavad annavad tüüpiliselt bakalaureusekraadi rahvusvahelise nimetusega „Bachelor of Science“ (BSc) ning videomängude kunsti õppekavad kraadi „Bachelor of Arts“ (BA). Eestikeelsed vasted nendele üldnimetustele puuduvad. Videomängude disaini ja arenduse õppekavad erinevad kraadi poolest – mõnel õppekaval antakse üks kraad ja mõnel teine kraad. Seejärel tegime loodud profiilidel ka põhikomponentanalüüsi. Leidsime, et kõige suurema kaaluga klassifikaatorid on arendus, kunst, disain ja tööstus. Visualiseerides nii esimest kui teist põhikomponenti, leidsime, et disaini ja tööstuse suunaga õppekavad on tüüpiliselt mängude disaini ja arenduse õppekavade klastrist, arendusele suunatud õppekavad on tüüpiliselt mängude programmeerimise õppekavade klastrist ning kunstile suunatud õppekavad mängude kunsti õppekavade klastrist. Lisaks sellele loogilisele tulemusele leidsime klasteranalüüsi põhjal ka, et videomängude disaini ja arenduse klastri õppekavad paiknevad oma profiilide poolest videomängude programmeerimise ja videomängude kunsti õppekavade vahel. Sellest järeldame, et suur osa nendest õppekavadest õpetavad, kuidas juhtida ning ühendada programmeerijate ning kunstnike tööd videomängude loomisel.

Tehtud kolme uuringu tulemuste põhjal lõime Tartu Ülikoolis potentsiaalselt realiseeritava videomängude disaini ja arenduse bakalaureuseõppekava plaani. Õppekava loomisel suhtlesime kohalike õppejõudude, meelelahutustarkvara sektori ettevõtete, erinevate õppetasemete õpilaste ja teiste huvirühmadega. Doktoristöös on esitatud loodud õppekava plaan ning üldised seletused iga

semestri ainete kavandamise põhimõtetest. Sõnastasime viis printsiipi, mis uuringute tulemuste ja õppekava koostamise protsessi jooksul ilmnesid. Esiteks, kõik kohustuslikud ained peavad olema selge kasuteguriga videomängude disainiks või arenduseks. Selle selguse vajalikkust tõid välja ettevõtted meie teises uuringus. Teiseks, kontekstipõhine suhtlusoskus tööstuses nõutud erialadel peab olema kaetud. Selle õppekava lõpetajad peavad juhtima ja koostööd tegema teiste erialadega, seega peavad nad oskama teiste tööd mõista ja sellest nendega rääkima. Kolmandaks, disaini ja arenduse erialamoodulite üliõpilastel peab olema õppekava jooksul läbivalt ühiseid koostööpunkte. Selle lahendamiseks disainisime õppekavasse mängujämmide „selgroo," kus kõik õppekaval õppivad üliõpilased teevad õppetöö raames tihti sihipärast koostööd. Neljandaks, õpingute jooksul loodud projektid peavad olema õpilastele selgelt kasulikud ka pärast lõpetamist. Ettevõtted leidsid, et ilma portfooliota kandidaati on raske tööle võtta või isegi töövestlusele kutsuda. Seega õppekava lõpuprojekti ühe osana on disainitud portfoolio, kuhu õpilased koondavad oma õpingute jooksul tehtud projektid. Viiendaks, kuna videomängud on erialadevahelised, peab ka õppekava seda olema. Õpe peab olema piisavalt lai, et toetada nii printsiipi kaks, kui ka anda piisavad teadmised ja oskused baastasemel tööks nii videomängude kui ka üldiselt meelelahutustarkvara loomisel vajalike valdkondadega.

VISUAL ABSTRACT IN ESTONIAN



PUBLICATIONS

CURRICULUM VITAE

Name:	Raimond-Hendrik Tunnel
Date of birth:	August 29, 1989
Citizenship:	Estonia
Current position:	Junior Research Fellow of Computer Graphics Education, Institute of Computer Science, University of Tartu
Phone:	+372 5555 9463
E-mail:	raimond.tunnel@ut.ee

Education:

2013 – 2015	University of Tartu, Institute of Computer Science, Master of Science (in Engineering), Computer Science, *cum laude*
2008 – 2012	University of Tartu, Institute of Computer Science, Bachelor of Science (in Engineering), Computer Science
1996 – 2008	Tallinn Secondary School No. 32, general secondary education

ELULOOKIRJELDUS

Nimi:	Raimond-Hendrik Tunnel
Sünniaeg:	29. August 1989
Kodakondsus:	Eesti
Praegune töökoht:	arvutigraafika hariduse nooremteadur, arvutiteaduse instituut, Tartu Ülikool
Telefon:	+372 5555 9463
E-post:	raimond.tunnel@ut.ee

Haridus:

2013 – 2015	Tartu Ülikool, arvutiteaduse instituut, tehnikateaduse magister (MSc), informaatika, *cum laude*
2008 – 2012	Tartu Ülikool, arvutiteaduse instituut, tehnikateaduse bakalaureus (BSc), informaatika
1996 – 2008	Tallinna 32. Keskkool, üldkeskharidus

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.