TU WIEN Informatics

SmartShot: Ein Serious Game zum Erlernen von Computergrafik

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Kevin Chelappurath
Matrikelnummer 12119880

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Annalena Ulschmid, MSc BSc

Wien, 31. Oktober 2026

Kevin Chelappurath	Michael Wimmer

TU WIEN Informatics

SmartShot: A Serious Learning Game on Computer Graphics

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Kevin Chelappurath
Registration Number 12119880

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Assistance: Annalena Ulschmid, MSc BSc

Vienna, October 31, 2026

Kevin Chelappurath	Michael Wimmer

Erklärung zur Verfassung der Arbeit

Kevin Chelappurath

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 31. Oktober 2026

Kevin Chelappurath

Declaration of Authorship

Kevin Chelappurath

I hereby declare that I have written this thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work – including tables, maps and figures – which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.
I further declare that I have used generative AI tools only as an aid, and that my own intellectual and creative efforts predominate in this work. In the appendix "Overview of Generative AI Tools Used" I have listed all generative AI tools that were used in the creation of this work, and indicated where in the work they were used. If whole passages of text were used without substantial changes, I have indicated the input (prompts) I formulated and the IT application used with its product name and version number/date.

Vienna, October 31, 2026

Kevin Chelappurath

Danksagung

Ich möchte mich bei meiner Betreuerin Annalena Ulschmid für die kontinuierliche Unterstützung, das wertvolle Feedback und die vielen hilfreichen Anregungen während der Entwicklung von *SmartShot* bedanken. Ebenso danke ich der Forschungsgruppe Computer Graphics der TU Wien für die Bereitstellung der Infrastruktur und der Backend-Anbindung. Mein Dank gilt außerdem allen, die das Spiel getestet und Feedback gegeben haben: allen Studierenden, die die Prüfung abgelegt und *SmartShot* zum Lernen genutzt haben, und allen, die keine Prüfung abgelegt und sich trotzdem die Zeit genommen haben, das Spiel auszuprobieren.

Acknowledgements

I would like to thank my advisor, Annalena Ulschmid, for her continuous support, her feedback and many helpful suggestions throughout the development of *SmartShot*. I also thank the Research Unit of Computer Graphics at TU Wien for providing the infrastructure and the backend connectivity. My thanks also go to everyone who tested the game and gave feedback: all students who took the exam and used *SmartShot* to study and everyone who did not take the exam but still took the time to try out the game.

Kurzfassung

Diese Bachelorarbeit befasst sich mit der Entwicklung und Evaluierung von *SmartShot*, einem quizbasierten Lernspiel für Computergrafik. Die Idee hinter dem Projekt ist es, Studierenden eine ansprechendere Möglichkeit zu bieten, ihr Wissen zu testen, indem sie Fragen in einer schießstandähnlichen Umgebung beantworten. Bei jeder Frage trifft die spielende Person die Ziele mit den richtigen Antworten. Die Fragen stammen aus echten Prüfungen von zwei Computergrafik-Lehrveranstaltungen der TU Wien. Die Arbeit betrachtet, wie das Spiel gestaltet und umgesetzt ist, wie Studierende es bei der Prüfungsvorbereitung nutzen und wie sie im Spiel und in der Prüfung abschneiden. In einer Nutzerstudie verwendeten 9 Studierende einer dieser Lehrveranstaltungen *SmartShot* vor ihrer Wiederholungsprüfung. Die meisten spielten nur ein oder zwei Runden an einem einzigen Tag und beantworteten rund 60 % der Fragen vollständig richtig. Wer mehrere Runden spielte, verbesserte sich oder blieb gleich. Die 3 Teilnehmenden mit Prüfungspunkten schnitten besser ab als der Durchschnitt aller Prüfungsteilnehmenden, die Stichprobe ist aber zu klein für Schlüsse über die Wirkung des Spiels.

Abstract

This bachelor's thesis focuses on developing and evaluating *SmartShot*, a quiz-based learning game for computer graphics. The idea behind the project is to create a more engaging way for students to test their knowledge by answering questions in a shooting-range-style environment. For each question, the player shoots the targets with the correct answers. The questions are taken from real exams of two computer graphics courses at TU Wien. The thesis looks at how the game is designed and built, how students use it during their exam preparation and how they perform in the game and in the exam. In a user study, 9 students of one of these courses used *SmartShot* before their retake exam. Most of them played only one or two rounds on a single day and answered around 60 % of the questions completely correctly. Participants who played several rounds improved or stayed the same. The 3 participants with exam points scored above the mean of all exam participants, but the sample is too small to draw conclusions about the effect of the game.

Contents

Kurzfassung **vii**

Abstract **viii**

Contents **ix**

1 Introduction **1**

2 Related Work **4**
2.1 Serious Games and Game-Based Learning 4
2.2 Testing Effect and Feedback 5
2.3 Gamification 5
2.4 Learning Games for Computer Science and Computer Graphics 6

3 Methodology **8**
3.1 Game Design 8
3.2 Implementation 13
3.3 Considered Features 26
3.4 Study Design and Procedure 28

4 Results **32**
4.1 Evaluation 32
4.2 Discussion 36

5 Conclusion **38**

A Example of the Question Data **40**

B Study Information Sheet **41**
B.1 Study Information 41
B.2 Purpose and Procedure of the Study 41
B.3 Voluntary Participation 42
B.4 Potential Risks and Benefits 42
B.5 Data Protection and Confidentiality 42

B.6 Your Rights Under the GDPR 43
B.7 Reliability of the learning game's content 44
B.8 Required Confirmations 44

Overview of Generative AI Tools Used **45**

List of Figures **46**

List of Tables **47**

Acronyms **48**

Bibliography **49**

CHAPTER 1

Introduction

This thesis presents *SmartShot*, a serious game developed with Unity [1], which runs in the browser via the Web Graphics Library (WebGL) and helps students of the computer graphics courses at TU Wien study for an upcoming exam. A *serious game* is a game whose main purpose is not entertainment but, for example, training or learning [2]. In *SmartShot*, the gameplay is only the means to practice exam questions. The students have to learn a lot of abstract, mathematical content (transformations, color models, rasterization, lighting, ...). Additionally, the exams consist of multiple-choice questions with partial credit, which means that wrongly selected answers cost points. Typical exam preparation is rather passive: for example, reading slides and lecture notes can get monotonous and going through old exams does not always give real feedback. However, actively testing oneself works better than repeatedly reading the lecture notes [3, 4].

The main problem is that self-testing is rather effortful and not very rewarding, so many students stick to reading. Games or game elements can ease this problem, because they offer something different from plain reading. Serious games can improve learning [5] and game elements can increase motivation [6]. The idea of this thesis is therefore to wrap the self-testing into a game. More details on the learning theory follow in Chapter 2.

Based on this, this thesis addresses the following research questions (RQs):

- RQ1: How can a game-based quiz for exam preparation in computer graphics courses be designed?
- RQ2: How do students use *SmartShot* during their exam preparation and how does their performance in the game develop?
- RQ3: How did the participants perform in the retake exam compared to all exam participants?

We answer RQ1 in detail in Chapter 3 and RQ2 and RQ3 with the results of the study in Chapter 4. In this study, students of the course *Einführung in Visual Computing* (EVC, *Introduction to Visual Computing*) filled in a questionnaire and then used *SmartShot* to prepare for the retake exam. We linked the questionnaire, the game data and, if they agreed, the exam points via a pseudonymous code (see Section 3.4).

The key idea is answering questions shown on a screen by shooting targets with the corresponding answers in a first-person shooting range. After submitting, there is immediate feedback. *SmartShot* runs in the browser, so there is no download or account needed. We only have to provide one link to play the game.

There is no violence included to keep the focus on the questions and the process of answering them. There are only targets and no real enemies or characters. Shooting is just the way to select answers (see Figure 1.1).



Figure 1.1: A multiple-choice question in *SmartShot*. The player selects answers by shooting the targets A to D and submits them by shooting SUBMIT.

In summary, this thesis contributes the following:

- The design and implementation of *SmartShot* with real exam questions from the courses EVC and *Grundlagen der Computergraphik* (GCG, *Fundamentals of Computer Graphics*), 640 questions in total, which can be expanded (Chapter 3 and Section 3.2.3)
- A scoring system based on the EVC exam scheme, so that the game trains the same trade-off as the real exam: wrong answers cost points, so guessing is risky (Section 3.2.6)

- A user study with students of the EVC course, primarily those taking the retake exam, linked to the game data and the exam points via a pseudonymous code (Section 3.4 and Chapter 4)
- An overview of the technical difficulties of a Unity game in the browser and possible solutions (Section 3.2.12)

The rest of the thesis consists of four chapters. Chapter 2 reviews related work on serious games, the testing effect, how gamification can influence learning and other learning games for computer science or computer graphics. Chapter 3 explains the methodology: the game design, the implementation of the game, features that were considered but not implemented and the study design. Chapter 4 reports the results of the study and discusses them together with the limitations. Chapter 5 sums up the results and outlines plans for future work.

CHAPTER 2

Related Work

This chapter summarizes the research that *SmartShot* builds on. Section 2.1 introduces serious games and how they affect learning. Section 2.2 explains the testing effect and the role of feedback. Section 2.3 covers gamification and Section 2.4 presents other learning games for computer science and computer graphics. The design decisions themselves follow in Chapter 3.

2.1 Serious Games and Game-Based Learning

A serious game is a game whose main purpose is not entertainment, but for example training or learning [2]. We already introduced this term in Chapter 1. This section shows what research says about the effect of such games on learning.

Wouters et al. [5] compared serious games with conventional instruction in a meta-analysis. A meta-analysis combines the results of many earlier studies. Serious games were more effective for learning and retention. However, they were not more motivating than conventional instruction. Players learned more when the game was combined with other teaching methods, when they played it in several sessions and when they worked in groups. For *SmartShot*, this shows that the game should be an addition to the lecture and the exam preparation, not a replacement. It should also work better if students play it several times, which most participants of our study did not do (see Section 4.1.2).

Clark [7] argues that it is the instructional method that influences learning and not the medium. So a game does not automatically lead to better learning. What matters is what the players actually do in it. In *SmartShot*, the method is answering real exam questions. The shooting range is only the medium.

Habgood and Ainsworth [8] recommend intrinsic integration. This means that the learning content is part of the core mechanic of the game and not an add-on, for example as a quiz that appears between two levels of an otherwise unrelated game. They tested this with

their game *Zombie Division*, which teaches mathematics to children aged 7 to 11. In the intrinsic version, the children defeated enemies by dividing the numbers shown on them. In the extrinsic version, the fighting had nothing to do with mathematics and the same division problems appeared as a quiz at the end of each level. The children learned more from the intrinsic version than from the extrinsic version in the same time. When they could choose freely, they also played the intrinsic version much longer. Their study was done with children and mathematics, so we only use it as a design principle. *SmartShot* does not fully meet this principle. The shooting is not connected to the learning content itself, so the questions could be replaced by any other topic (see Section 3.1.3). However, the questions are not an add-on between levels either, since answering them by shooting is the main action of the game.

2.2 Testing Effect and Feedback

Tests are usually seen as a way to assess what students know. However, taking a test also improves later retention. This is called the testing effect. Roediger and Karpicke [3] let students study short educational texts. Then the students had to either take recall tests without feedback or restudy the texts the same number of times. The final test took place after three different time periods: 5 minutes, 2 days or 1 week. After 5 minutes, restudying was better. After 2 days and after 1 week, the recall tests led to better results.

Restudying also increased the students' confidence, although they remembered less. This can explain why many students prefer reading over testing themselves (see Chapter 1). Adesope et al. [4] confirm that students learn more with practice tests than with restudying. How large this advantage is depends on how the tests are designed, who takes part and how the study is set up.

However, there is one problem with multiple-choice tests. Students can learn the wrong answer options (lures) as correct. Butler and Roediger [9] showed that feedback after a multiple-choice test increased the correct answers in a later test. Delayed feedback increased them even more than immediate feedback. Both also reduced the wrong answers taken over from the lures. Therefore, they recommend that teachers give feedback after multiple-choice tests. The EVC exams are multiple-choice, so *SmartShot* shows the correct solution right after every question (see Sections 3.1.6 and 3.2.5).

2.3 Gamification

Deterding et al. [10] define gamification as "the use of game design elements in non-game contexts". The difference from a serious game is that a serious game is a whole game. Gamification only adds single game elements, for example points or leaderboards, to something that is not a game. *SmartShot* is a serious game, but it also uses gamification elements: points, a personal best and a leaderboard (see Section 3.1.6).

Sailer and Homner [6] analyzed the effect of gamification on learning. They found significant small effects on cognitive ($g = 0.49$), motivational ($g = 0.36$) and behavioral ($g = 0.25$) learning outcomes. Significant means that an effect is probably not caused by chance. The value g is an effect size called Hedges' g, which describes how large the difference between groups with and without gamification is. The cognitive effect also held up in very carefully designed studies. This was less the case for the motivational and behavioral effects. For behavioral outcomes, a fictional game world or story and social interaction made a difference.

However, game elements can also have negative effects. Hanus and Fox [11] compared a course with badges, a leaderboard and coins with the same course without them over a 16-week semester. Students in the gamified course had lower intrinsic motivation and lower final exam scores. Intrinsic motivation means that someone does an activity because it is interesting in itself and not because of a reward. According to the authors, the students probably felt controlled by the rewards, which lowered their intrinsic motivation. For this reason, the leaderboard in *SmartShot* is optional (see Sections 3.1.6 and 3.2.9).

2.4 Learning Games for Computer Science and Computer Graphics

Quiz games are a common way to bring game elements into learning. Wang and Tahir [12] reviewed 93 studies on *Kahoot!*, a quiz game used in the classroom. They conclude that *Kahoot!* can improve learning performance compared with traditional teaching. However, some of the studies found little or no effect. *Kahoot!* is played together in class and led by a teacher and everyone answers the same question at the same time. *SmartShot* is played alone at home, at one's own pace. Six of the 9 participants of our study had already played quiz games like *Kahoot!* (see Section 4.1.1).

Battistella and Gresse von Wangenheim [13] reviewed 107 games for teaching computing in higher education. Most of them are for software engineering, programming fundamentals, networks, security as well as algorithms and complexity. For computer graphics, they found very few or no games. Quiz games are one of the genres used. The games seem to cover the knowledge areas well, but they are often not incorporated into the learning context of a course. *SmartShot* addresses this gap with a game for computer graphics that is tied to the real exams of a real course.

One of the few examples for computer graphics is *Immersivio* by Moradi et al. [14]. It is a problem-based serious game for a 3D computer graphics course. The authors point out that 3D computer graphics has many abstract and algorithmic concepts that students find hard to understand. They tested the game with undergraduate students in Malaysia in a study with a control group. The experimental group (24 students) learned with the game and the control group (26 students) had conventional teaching. The experimental group had significantly higher scores in the post-test, which is a test after the learning phase. *Immersivio* teaches new content with problems, while *SmartShot* is for reviewing

with exam questions. Additionally, the study shows how such an effect can be measured with a control group and a post-test. Our study could not do this (see Section 4.2.3).

In summary, *SmartShot* is a review game for exam preparation and not for teaching new content. It uses real exam questions of EVC and GCG and scores them like the EVC exam (see Section 3.1.5). It runs in the browser without installation (see Section 3.2.11). So *SmartShot* combines the testing effect (see Section 2.2) with a game in which answering the questions is the main action (see Section 2.1).

CHAPTER 3

Methodology

This chapter describes how we designed, built and evaluated *SmartShot*. Section 3.1 explains the game design and Section 3.2 explains how we implemented it. Section 3.3 lists the features that we considered but did not implement. Finally, Section 3.4 describes the design of the user study.

3.1 Game Design

3.1.1 Design Goals and Target Group

The target group of *SmartShot* and the accompanying study consists of students of the courses EVC and GCG at TU Wien. Especially important for the user study are those who use the game to prepare for their exams. *SmartShot* is not a replacement for the lecture or the lecture notes, but is intended as a tool for reviewing what has already been learned and for self-testing.

We derived five design goals based on RQ1 (see Chapter 1):

1. **Active retrieval:** The game should encourage active retrieval instead of passive reading by requiring a decision for every question [3, 4].
2. **Immediate feedback:** Every answer should be followed immediately by feedback. This feedback should include the correct solution [9].
3. **Motivation:** Game elements should increase motivation without distracting from the learning content [6].
4. **Closeness to the exam:** The conditions should be as close to the exam as possible, meaning real exam questions and the same scoring system as in EVC (see Section 3.2.6).

5. **Low barrier to entry:** The barrier should be low, which is why the game runs in the browser without installation and without an account.

Table 3.1 shows how we implemented these goals in the game. Before the implementation, we wrote a Game Design Document (GDD), which lists the planned features by priority (see Section 3.3).

Table 3.1: Design goals and their implementation in *SmartShot*.

Design goal	**Implementation in the game**	**Section**
Active retrieval	Every question has to be answered by shooting one or more answers	3.1.3, 3.2.5
Immediate feedback	Colored solution and points achieved after every question	3.2.5
Motivation	Points, personal best, leaderboard and difficulty levels	3.1.6, 3.2.9
Closeness to the exam	Questions from past exams, scoring according to the EVC scheme	3.1.4, 3.2.6
Low barrier to entry	Playable in the browser, without installation and without an account	3.1.2, 3.2.11

3.1.2 Platform and Visual Style

We developed *SmartShot* as a browser game, so students can open it with a shared link. We wanted to keep the barrier to participation as low as possible to reach more participants. As the game engine, we used Unity [1] (version 6000.3). A game engine is software that provides the basic functions of a game, for example the rendering, the physics and the input. We chose Unity since it is beginner-friendly and has a lot of documentation. Additionally, because of its large community, we could find solutions to many problems quickly. Unity also allows fast prototyping and has a stable export to WebGL. This export supports our fifth design goal, a low barrier to entry (see Section 3.1.1). Unreal Engine, for example, no longer offers an official web export [15].

A render pipeline defines how Unity draws the 3D scene on the screen. We chose the *Built-in Render Pipeline* from Unity instead of the Universal Render Pipeline (URP) or the High Definition Render Pipeline (HDRP). It is the simplest and most stable option for a small 3D game and runs reliably in WebGL. Since it has no unnecessary graphical overhead, the game also runs on less powerful computers.

We chose a small shooting range in a barn-like building as the environment (see Figure 3.1). We modeled all 3D objects (the building, the pistol and the targets) ourselves. We only used artificial intelligence (AI) to generate the pixel-art textures of the sun and the moon

(see the overview of generative AI tools used on page 45). We kept the graphics simple on purpose, since the focus should be on the learning and not on the visuals. Simple graphics also shorten the loading time in the browser.



Figure 3.1: Overview of the shooting hall with the menu panel, the shooting range, the answer targets and the screen for the questions.

The game was designed for a desktop computer with mouse and keyboard in fullscreen mode and not for mobile devices. This was also pointed out in the information sheet of the study (see Section 3.4.1).

3.1.3 Core Mechanics

SmartShot is a first-person quiz shooter. This means that the player sees the game world through the eyes of the character. The core loop consists of reading a question, shooting the targets to select the answers, submitting the answers by shooting a SUBMIT button, viewing the feedback and moving on to the next question by shooting a NEXT button (see Figure 3.2). For questions with several correct answers, the player simply hits several targets. The player can deselect a target by shooting it again, so that a wrong shot is not included in the final submission (see Section 3.2.5). This idea comes from the real exam, where students can fill in a box to mark an answer as not selected. A round consists of up to 20 questions. If the selected chapters contain fewer questions, the round is shorter. At the end, an end screen shows the statistics of the round (see Section 3.2.10).

There are several reasons for choosing shooting as the core mechanic. Aiming and shooting is an active task that requires attention. Also, the player has to commit to an answer by actually hitting it. On the higher difficulty levels, there is also slight time

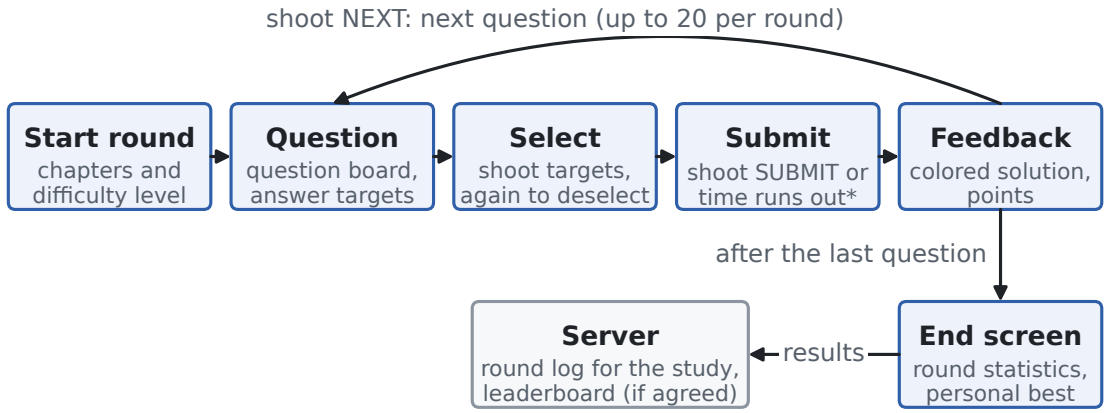


Figure 3.2: Flow of one round. Each question is answered by selecting targets and shooting SUBMIT, followed by feedback.

pressure (a base time of 45 seconds per question on *Medium* and 25 seconds on *Hard*, see Section 3.2.7). So reviewing should be less passive than reading slides or clicking on answers in a conventional online quiz. Additionally, hitting a target and gaining points for correct answers is more rewarding than clicking on an answer. This adds extrinsic motivation, which comes from a reward and not from the activity itself.

However, the shooting mechanic is not directly connected to the learning content. The content could be replaced by questions on any other topic without changing anything about the game. So according to Habgood and Ainsworth [8], this is not an intrinsic integration, where the learning content itself is part of the game mechanic. Instead, the shooting is a motivating frame for answering the questions. Clark [7] also argues that the instructional method and not the medium decides how well students learn. In *SmartShot*, this method is repeated retrieval with immediate feedback (see Section 3.1.1).

3.1.4 Learning Content and Question Selection

The content of *SmartShot* comes from two real computer graphics courses, EVC and GCG. The advantage is that we examine the game with real exam material and not with general knowledge. Also, the participants are exactly the students who take these exams.

The questions are taken from past exams of the two courses and were provided by the advisor of this thesis. These past exams are also available to the students on TUWEL, the e-learning platform of TU Wien. For GCG, the solutions are available there as well. For EVC, the exams are provided without solutions to avoid confusion, since the solutions sometimes still contain errors. So *SmartShot* is especially useful for EVC, since the game shows the solution directly after every question. We did not change the content of the questions. Since the game is only offered in English, we translated the German-language questions into English with the help of AI (see the overview of generative AI tools used

on page 45). The exams are also offered in English, so official English versions of these questions exist as well. These were checked manually and would be more reliable than an AI translation. Replacing the translated questions with them is a possible improvement for the future (see Chapter 5).

With the *Chapter Select* (13 chapters for EVC and 7 chapters for GCG, see Section 3.2.10), players can practice single topics, so they do not have to go through all questions every time. The EVC course has several tests that each cover certain chapters. For EVC, there is therefore an additional quick selection that selects exactly the chapters of one test (see Section 3.2.10). So players can study for one specific test. Players who do not have a calculator at hand or only want to answer conceptual questions can also exclude the questions that require calculations (see Section 3.2.10).

The questions of a round are selected randomly and the answers are shuffled again for every question. So players learn the content and not the position of the correct answer. Sections 3.2.3 and 3.2.4 explain the data format and the presentation of the questions in detail.

3.1.5 Scoring and Difficulty

For the scoring, we used the scheme of the EVC exams on purpose. The GDD originally planned a simpler scheme with plus points for correct and minus points for wrong answers. With the exam scheme, however, the feedback in the game corresponds to the grading in the real exam. Selecting a wrong answer costs points, so guessing is risky. So players train the same decision as in the exam: whether to tick an answer they are not sure about. Section 3.2.6 explains the exact calculation with an example.

The difficulty levels do not change the questions, but only the pressure under which they have to be answered. There are three levels, *Easy*, *Medium* and *Hard* (see Section 3.2.7). The higher levels add a time limit, limited ammunition with reloading and on *Hard* also moving targets. The higher levels mainly test how quickly and confidently the knowledge can be recalled. On *Easy*, there is no time limit. So players who have exam anxiety or want to study at their own pace can play without pressure. This is also pointed out in the information sheet of the study (see Section 3.4.1).

The time limit also depends on the length of the question and calculation questions get extra time. So even long questions can be solved in time (see Section 3.2.7).

3.1.6 Motivation and Game Elements

To increase motivation, we used several game elements: points, accuracy, the personal best, a leaderboard with several rankings and different difficulty levels [6] (see Sections 3.2.6, 3.2.7 and 3.2.9). The leaderboard allows players to compete with others. However, taking part is voluntary and only the pseudonymous code is displayed (see Section 3.2.8). We kept the leaderboard optional, because rankings can also demotivate players [11]. Also,

players who do not share their own results cannot see the rankings of the others either (see Section 3.2.9).

The game also gives immediate feedback after every question, with color-coded answers and the points achieved. If the answer is not completely correct, the feedback additionally shows how many and which correct and wrong answers were selected (see Section 3.2.5). If the evaluation only took place at the end of a round, players would not know during the round how well they are doing, which could lower their motivation.

Players can also adjust the game to their own preferences. For the atmosphere, there is a light and a dark mode. Depending on the mode, the sky, the lighting and the fog change and either the sun or the moon is shining. A particle effect, which is a cloud of many small particles, also makes hits visible. In the settings, the color of the crosshair (the aiming mark in the center of the screen, six colors to choose from), the brightness, the field of view (how much of the scene is visible at once) and the mouse sensitivity can be adjusted. The vertical mouse movement (Y-axis) can also be inverted (see Section 3.2.10).

During the design, we decided not to overload the game, since the learning content and the questions should be the focus. Therefore, there are neither enemies nor a story. The rounds should also be short (up to 20 questions, see Section 3.1.3), so that they do not drag on.

3.2 Implementation

3.2.1 Technical Foundation

For the movement of the player character, we used the First Person Controller from Unity's official *Starter Assets* [16], which is based on Unity's Input System, the package for keyboard and mouse input. It provides simple movement with the W, A, S and D keys, sprinting with the Left Shift key, the ability to look around with the mouse and a jump function. So we did not have to write our own controller and could focus on the quiz and the learning mechanics.

For the user interface (UI) of the game, we used Unity's UI system uGUI. In uGUI, all UI elements such as texts and buttons are placed on a canvas. We implemented the text elements with TextMeshPro, a Unity package for displaying text. For the question board, we placed a canvas directly in the 3D world (see Section 3.2.4). For the heads-up display (HUD), which contains, for example, the timer and the ammunition display (see Section 3.2.7), we used a separate canvas that is overlaid on the entire screen. This canvas is visible during an active round and is hidden in menus such as the pause menu.

3.2.2 Architecture

The whole game takes place in a single scene. In Unity, a scene contains all objects of one level or area. So there is no need to switch between scenes and all areas can be reached quickly. We divided the logic into several components and implemented the

central components as singletons. This means that exactly one instance of each of these components exists, which can be accessed directly by other parts of the game. The most important component of the game is the QuizManager, which loads, displays and evaluates all questions. It also controls the timer and the order of the questions in a round. Table 3.2 summarizes the most important components and their tasks.

Table 3.2: Main components of *SmartShot* and their tasks.

Component	**Task**
QuizManager	Loading, displaying and evaluating questions; timer
ScoreManager	Score of the current round
GameStartManager	Starting a round, measuring the round duration
GameEndManager	End screen
PauseMenuManager	Pause menu, settings, difficulty level
ChapterSelectManager	*Chapter Select*
PlayerIdentity, TutorialManager, LeaderboardConsentManager	First-start sequence
AnalyticsManager, HighscoreClient	Communication with the server
GunShoot, Target, TargetMover	Shooting, targets and their movement

Section 3.1.3 explains the flow of a round. Once a round is completed, an end screen appears. It shows an overview of the round with several statistics as well as, if the player has agreed to the leaderboard, the personal best across all previous rounds (see Section 3.2.9). The game also sends the results to a server (see Figure 3.3). The server logs them for the evaluation of the study and, if the player has given consent, updates the leaderboard (see Sections 3.2.8 and 3.2.9).

Another statistic we recorded is the round duration (see Section 3.2.8). The game calculates it using Unity's internal time counter (`Time.time`). This counter measures the seconds since the game was opened and is not reset when a new round starts. Therefore, the game stores the current value of this counter at the start of a round. At the end of the round, it subtracts the stored value from the current value of the counter, which results in the duration of the round.

The counter also only runs while the game is not paused. When the player opens the pause menu, the counter stops and only continues once the player resumes the game. So the round duration does not include the time in the pause menu and we do not have to measure and subtract it separately. The real clock time would keep running during the pause.

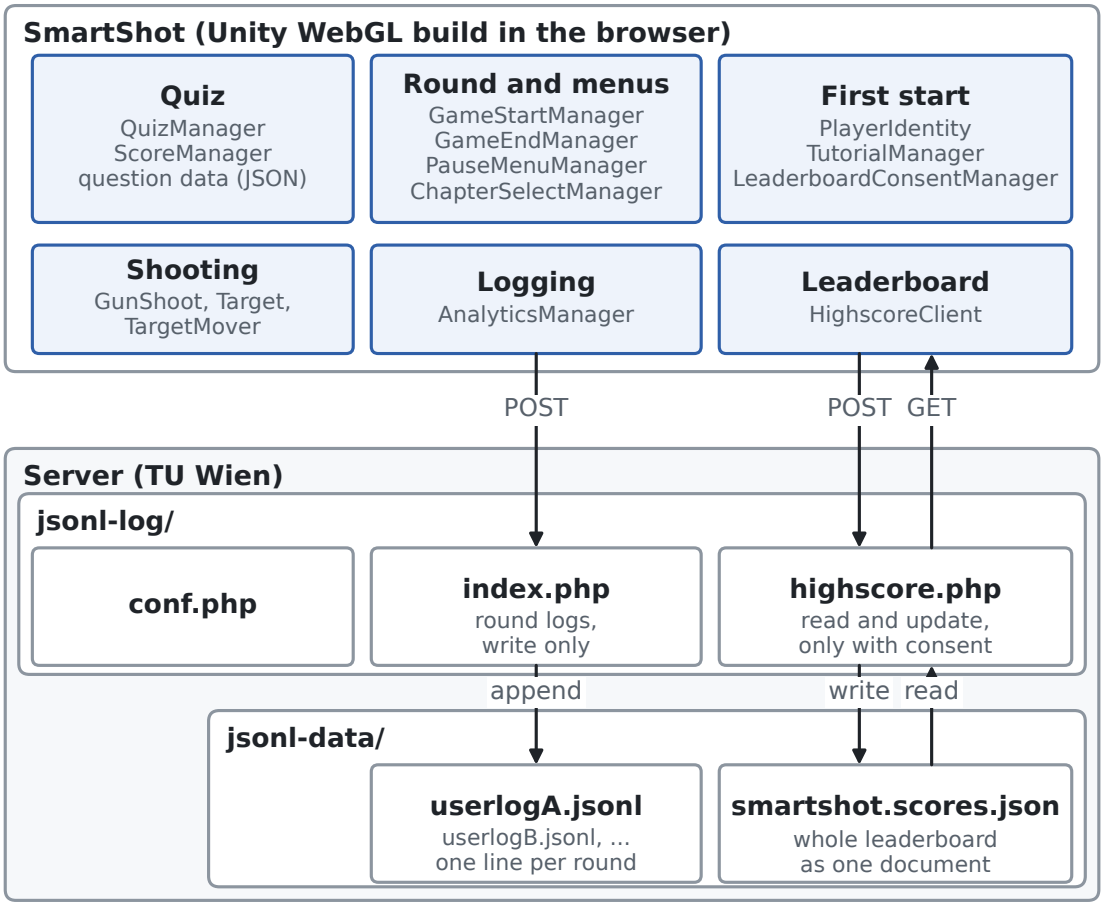


Figure 3.3: Main components of *SmartShot* and their connection to the server for logging and the leaderboard.

3.2.3 Question Data

The question catalog consists of several chapters, separated by the two courses, EVC and GCG. We store the questions as files in the JavaScript Object Notation (JSON) format, with each file containing one chapter. Table 3.3 shows the fields a question consists of. Appendix A shows a complete example.

Since there are only four targets in the game, the game displays at most four answer options per question. It always includes all correct options, except for two GCG questions with five correct options (see Section 3.2.4).

The advantage of this approach is that the questions are independent of the rest of the system. So we can extend and improve them at any time without changing the program code. Also, anyone can edit the files with a text editor without knowing the code. However, to see the changes in the game, we have to build the game again, which means creating a new version of the WebGL application. The question files are part of

Table 3.3: Fields of a question in the JSON files.

Field	**Description**
Question text	Text of the question
Answer options	2 to 8 options, in the order of the question collection
Correct options	Indices of the correct options (one or more)
Points	Point value of the question, 3 by default
`isCalculation`	Value (true or false) indicating whether it is a calculation question
Numeric value	For numeric questions, the correct number in decimal notation (the game allows a deviation of ± 0.005)

this application. If a file is faulty or missing, the game skips it and keeps running.

At the start of a round, the game only uses questions from the chapters selected in the *Chapter Select*. It places all questions from these chapters into a common pool and draws up to 20 questions from it at random. Optionally, players can exclude calculation questions beforehand (see Section 3.2.10). The two courses stay separate and players cannot play both at the same time. Each question also stores the course it belongs to and the border of the question board shows it in color: light blue for EVC questions and orange for GCG questions.

We did not include the first chapter of both courses, since it is an introduction and not relevant for the exams. Table 3.4 summarizes the size of the question bank.

Table 3.4: Size of the question bank.

	EVC	**GCG**	**Total**
Chapters	13 (Ch. 2 to 14)	7 (Ch. 2 to 8)	20
Questions	462	178	640
Calculation questions	130	23	153
Numeric questions (free input)	40	7	47
Several correct answers	62	41	103
Exactly 2 options	228	0	228
More than 4 options	128	55	183

3.2.4 Presentation of Questions and Targets

To select answers, the game contains four targets labeled with the letters A to D as well as a separate SUBMIT target, with which the player submits the answers (see Figure 1.1). After submitting, the SUBMIT target turns into a NEXT target, with which the player moves on to the next question (see Section 3.2.5).

One problem was that many questions from the question collection have more than four answer options. We therefore had to reduce these to four for display. The game always includes all correct answers and fills the remaining slots with randomly selected wrong answers. Only two GCG questions have five correct answers. For these, the game shows four of them. Afterwards, it shuffles the order of the four displayed answers. This happens for every question, including questions with four or fewer answer options. So players cannot memorize the position of a correct answer, since the order is usually different every time a question is repeated.

There are also questions with exactly two answer options, for example true/false questions. For these, the game hides the two outer targets and only shows the two middle targets, labeled A and B (see Figure 3.4a).

On the *Medium* and *Hard* difficulty levels, players no longer answer questions whose answer is a number with the normal answer targets, but with a numeric input. For this, the four targets are labeled -1, -0.01, $+0.01$ and $+1$ from left to right (see Figure 3.4b). This allows players to set a number precisely with up to two decimal places. So that entering large numbers does not take too long, players can hold down the mouse button. The weapon then keeps firing automatically, about ten times per second and each shot changes the number by the same amount. For example, holding the mouse button on the $+1$ target for one second increases the number by 10. The question board shows the currently set number. On the *Easy* difficulty level, the same questions instead appear as regular multiple-choice questions (see Section 3.2.7).



(a) A true/false question. Only the two middle targets are shown.



(b) A numeric question. The player sets the value with the targets for -1, -0.01, $+0.01$ and $+1$.

Figure 3.4: Question types besides multiple choice.

Since the questions and answers vary in length, the game adjusts the layout of the question board for every question, so that it stays readable. The question receives as much height as it needs and the answers share the remaining space. If the space becomes too tight, the game first reduces the gap between the question and the answers and only then the font size. All answers also use the same font size, so that the list looks uniform.

The questions store formulas as plain text using `^` and `_`, for example `R^-1` or `P_Q`. In the game, these characters become real superscripts and subscripts, so that the formulas

appear as R^{-1} and P_Q. The game also displays variables such as `P0` and `v1` with an index as P_0 and v_1. The questions in the JSON files remain unchanged.

3.2.5 Shooting and Selection

We implemented shooting with a raycast. This is an invisible ray that is cast from the center of the screen in the viewing direction. The first object hit by this ray counts as hit. To make the shot visible, a short glowing line additionally travels from the muzzle of the weapon to the hit point and disappears shortly afterwards. The game also creates a particle effect at the hit point. Together with the muzzle flash, this gives the player direct feedback for every shot.

When the player hits one of the answer targets, the game selects the corresponding answer and the letter above the target turns yellow (see Figure 3.5a). Hitting the same target again deselects the answer and the letter turns white again. In this way, the player can select several answers. Shooting the SUBMIT target submits the answers (see Section 3.2.4). After that, the player can no longer change the selection.

After submitting, the game colors all correct answers on the question board green and all wrong answers red, regardless of whether the player selected them (see Figure 3.5b). The letters above the targets remain yellow, so that it is still visible which answers the player selected. Below the answers, a text shows the points achieved (see Section 3.2.6). If the answer is not completely correct, it additionally shows how many correct and wrong answers the player selected and how many correct answers they missed. Together with the colors and the yellow letters, this shows exactly which answers were right or wrong. For a wrongly answered numeric question, the text instead shows the correct number and the player's own input.



(a) Selected answers are highlighted in yellow before the player shoots SUBMIT.



(b) Feedback after submitting. Correct and wrong answers are marked in color and the points are shown.

Figure 3.5: Selecting and submitting answers.

3.2.6 Scoring

The scoring follows the scheme of the EVC exams, for the reasons given in Section 3.1.5. If the player answers a question completely correctly, meaning that they select all correct and no wrong answers, they receive the full number of points. Otherwise, they receive partial credit, which the game calculates with the following two formulas:

$$\textit{netCorrect} = \max(0,\ \text{selected correct} - \text{selected wrong})$$
$$\text{points} = \text{round}\left(\text{question points} \cdot \frac{\textit{netCorrect}}{\text{number of correct answers}}\right)$$

First, the game subtracts the number of wrongly selected answers from the number of correctly selected answers, but the result cannot be smaller than 0. It then divides this value by the total number of correct answers of the question, multiplies it by the point value of the question and rounds it to a whole number. This means that every wrongly selected answer cancels out a correctly selected one. A correct answer that is not selected is only missing from *netCorrect* and does not cost extra points. For example, if a question has three correct answers and the player selects two correct and one wrong answer, *netCorrect* is 1, which results in $3 \cdot 1/3 = 1$ point. If the player selects nothing, only wrong answers or at least as many wrong as correct answers, they receive 0 points.

For numeric questions, there is no partial credit. If the entered number deviates from the correct solution by at most 0.005, the player receives the full number of points, otherwise 0 points. In the statistics at the end of a round, only completely correct answers count as correct. The statistics count questions with partial credit as wrong, but their points are still added to the score of the round.

3.2.7 Difficulty Levels

As explained in Section 3.1.5, the difficulty level only changes the pressure and not the questions. There are three levels, *Easy*, *Medium* and *Hard*, which the player can select in the pause menu (see Section 3.2.10). If the player changes the difficulty level during a running round, the game restarts the round. Table 3.5 summarizes the differences.

Table 3.5: Differences between the difficulty levels.

	Easy	**Medium**	**Hard**
Time limit per question	none	45 s	25 s
Magazine	unlimited	6 shots	4 shots
Reload time (R key)	none	1.5 s	2.5 s
Moving targets	no	no	on 60 % of questions
Numeric questions	as multiple choice	numeric input	numeric input

From *Medium* onwards, there is a time limit per question and a limited magazine. This means that the weapon only has a few shots before the player has to reload. The player

can reload at any time with the R key to get a full magazine again. If the magazine is empty, the next click automatically starts reloading instead of firing a shot. Since reloading takes time, the player has to take it into account when answering the questions. Shots at SUBMIT and NEXT as well as at the targets of the numeric input do not use any ammunition, since otherwise entering a number, for example, would hardly be possible.

Since the questions vary in length, the game automatically adjusts the time limit to each question. With more than 600 questions, we could not set a time for each question individually. Therefore, the game adds 0.12 s to the base time for every character beyond 150 characters, with a maximum of 40 s of additional time. Calculation questions receive an additional 15 s. When the time runs out, the game automatically submits the current selection.

On *Hard*, in 60 % of the questions, the targets additionally hang from a rail and move from side to side. This makes it harder to hit the answers. For numeric questions, the targets never move, so that the input remains precise. On *Easy*, the game displays numeric questions as regular multiple-choice questions and scores them in the same way as all other questions (see Sections 3.2.4 and 3.2.6).

3.2.8 Player Code and Logging

When the player starts the game for the first time, they have to enter a pseudonymous code (see Figure 3.6). It consists of the first two letters of the mother's first name, the player's own day of birth as a two-digit number and the last two letters of the place of birth. For example, a mother named Maria, a birthday on the 2nd of a month and Vienna as the place of birth result in the code `ma02na`. When the player enters the code, the game checks its format and only accepts valid codes. The player only has to enter the code once. The game then stores it in the browser and fills it into the input field every time the game starts again, so that the player only has to confirm it.

Through this code, we can link the questionnaires to the game data (see Section 3.4.4). The advantage is that every player has their own code without having to provide a name or an e-mail address. So the data does not directly identify a person.

After every round, the AnalyticsManager sends a record in JSON format via a Hypertext Transfer Protocol (HTTP) POST request to an endpoint on the web space of the research group. A POST request sends data to a server and an endpoint is the address on the server that receives it. There, the server appends each record to a file as a new line. This format is called JSON Lines. A record contains the code, the time of the round and the time of the previous game. It also contains the number of correct and wrong answers, the points, the accuracy and the duration of the round (see Section 3.2.2), as well as the number of rounds, the correct and wrong answers, the accuracy and the playing time over all previous rounds. Finally, the record states whether the player completed or aborted the round.

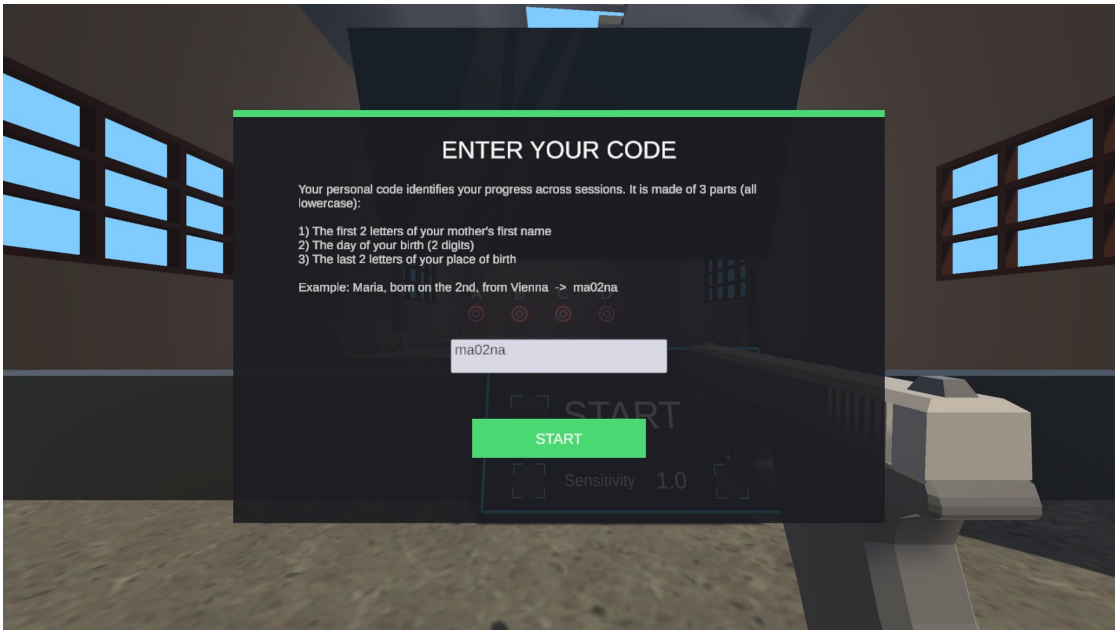


Figure 3.6: The code entry at the start of the game. The code links the game data to the questionnaires.

Since the server only appends the logged data and the game does not read it back, the game keeps the summed values locally in the browser. Aborted rounds do not count towards the summed results, since they have no final result. Only their playing time counts towards the total playing time.

3.2.9 Leaderboard and Consent

For the leaderboard, we use a separate endpoint, which is independent of the logging. Unlike the logging endpoint, it can be written and read. It is also located on the server of the research group and stores JSON data under an identifier (ID), in this case `smartshot`. The server stores the entire leaderboard under this ID as a single JSON document. The advantage is that one request is enough to load all entries and we do not need a separate database.

At the end of a round, the game loads the document, updates the player's own entry and sends the entire document back. For each person, the document stores the best score, the best accuracy and the most correct answers in a single round, as well as the number of rounds, the total points, the total number of correct answers and the number of perfect rounds. Each of these values has its own ranking in the leaderboard. The best values can only increase. The game also only writes the document if it loaded it successfully. This prevents an incomplete document from overwriting the entries of the other players.

A disadvantage of this approach is that two rounds ending at almost the same time can overwrite each other. In this case, the update of the earlier player is lost. To mitigate this, the game additionally stores the summed values locally in the browser and sends them again in the next round, which automatically corrects the entry. For the expected number of participants, this is acceptable, but it is not a complete solution. A better approach would be a database in which every player has their own record, so that the entries cannot overwrite each other.

When the player starts the game for the first time, the game asks them once before the first round whether they want to appear on the leaderboard with their pseudonymous code (see Section 3.2.10). If they decline, the game sends no data to the leaderboard, but in return they cannot see the entries of the other players either. They can change this decision later in the leaderboard. Since the game still keeps the summed values locally even after declining, nothing is lost if they agree later. The logging for the study (see Section 3.2.8) is independent of this decision, since it is part of the study participation and is not displayed publicly.

3.2.10 First Start and Menus

When the player starts the game for the first time, several windows appear in a fixed order. First, the player has to enter the pseudonymous code (see Section 3.2.8) and the game is paused during this step. Then a tutorial explains the controls and the flow of a round, for example that the player can deselect an answer by shooting it again. Afterwards, the game asks for the consent for the leaderboard (see Section 3.2.9) and opens the *Chapter Select*, so that the player knows how to continue after the tutorial. Only then does the player enter the play area, where they start a round via a menu on the wall with the options Start, Exit and Restart. On later visits, the code is already filled in and the player only has to confirm it. The game skips the tutorial, the consent and the *Chapter Select*, so that the player then starts directly in the play area.

The *Chapter Select* (see Figure 3.7) displays the chapters as tiles, separated by the two courses, EVC and GCG. When it opens, all chapters of the respective course are selected at first. For EVC, there is additionally a quick selection based on the previous partial exams: Test 1 covers chapters 2 to 7, Test 2 chapters 8 to 14 and Test 3 chapters 2 to 14. The player can also exclude calculation questions with a filter and deselect all chapters at once with a button. The game does not accept an empty selection. If the player changes the selection during a running round, the game restarts the round, but only when the player closes the *Chapter Select* and not with every click.

Players open the pause menu with the Tab key. Many other games use the Escape key for this. In the browser, however, Escape ends the mouse lock and, if active, the fullscreen mode, which is why we do not use this key (see Section 3.2.12). The pause menu also contains the settings. There, players can adjust the brightness, the field of view, the mouse sensitivity and the color of the crosshair. They can also invert the Y-axis, change the difficulty level (see Section 3.2.7), switch between light and dark mode and turn off

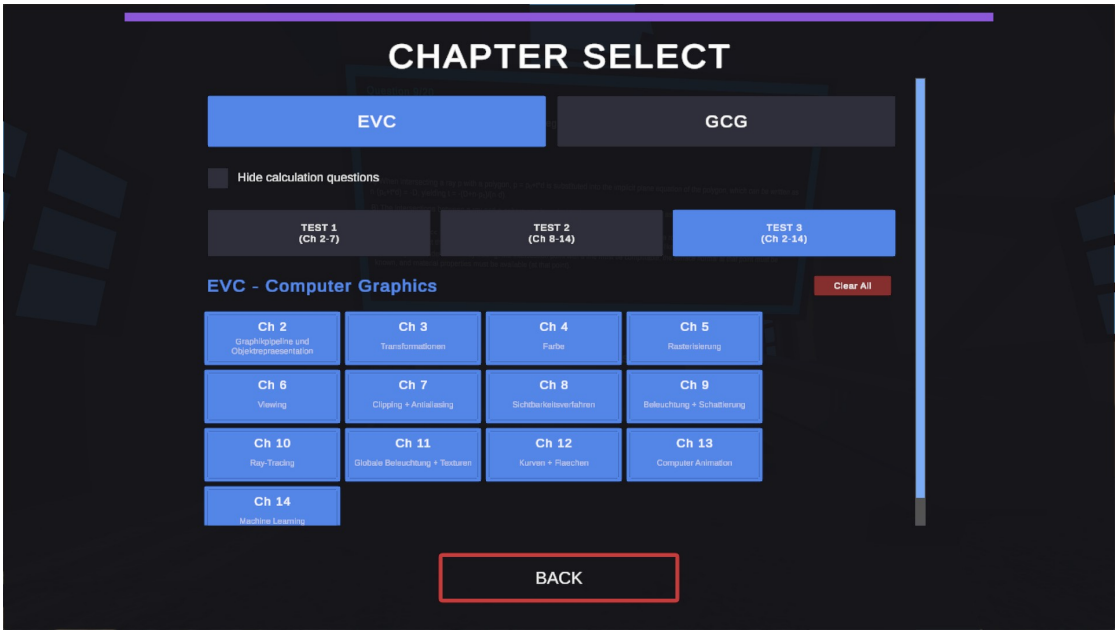


Figure 3.7: The *Chapter Select*. The chapters are shown as tiles, separated by course.

the sunlight or moonlight. If players get stuck in the play area, they can reset their position.

At the end of a round, the end screen (see Figure 3.8) shows the points, the number of correct and wrong answers and the accuracy. If the player has agreed to the leaderboard, it also shows the personal best (see Section 3.2.9). Pressing Tab opens the leaderboard (see Figure 3.9) and pressing Enter returns the player to the play area.

3.2.11 Browser Deployment

Since Unity's default template responds poorly to changes in the window size, we created a custom HyperText Markup Language (HTML) template for the WebGL version. In this template, the display area automatically adapts to the window size and the template limits the pixel density to twice the standard density. So on screens with a very high resolution, the game does not have to render too many pixels and runs smoothly. Players can play the game in fullscreen mode, either via a button below the display area or via the F key. During text input, for example when entering the pseudonymous code (see Section 3.2.8), the game disables the fullscreen function of the F key. Otherwise, players could not enter codes that contain an F, for example for the place of birth Düsseldorf or the mother's first name Fiona. Also, when players close or reload the tab, the browser asks whether they really want to leave the page, so that a running round is not lost by accident.



Figure 3.8: The end screen after a round with the points, the number of correct and wrong answers and the accuracy. The values are example values.

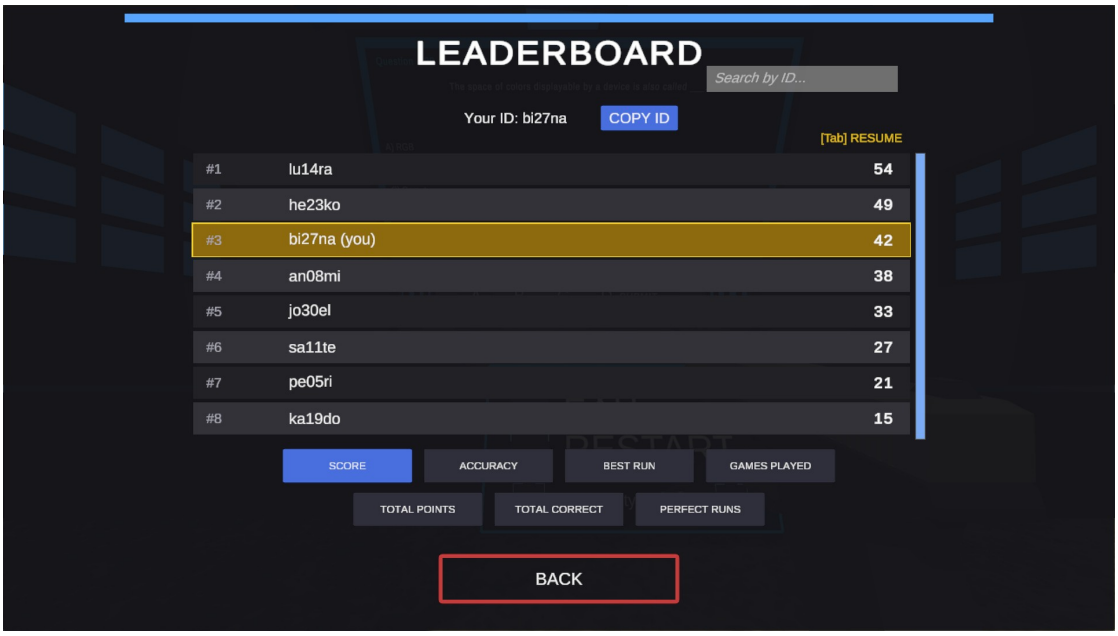


Figure 3.9: The leaderboard with the player's own code highlighted. The other codes shown are made up.

For functions that Unity does not provide in the browser itself, we created custom JavaScript plugins. JavaScript is the programming language of web pages and a plugin is a small piece of code that adds a function to the game. These control the mouse lock (see Section 3.2.12) and the mouse cursor, copy text to the clipboard, tell the web page whether a text input field is currently active and open links in a new tab.

The UI scales with the screen size, with a reference resolution of 2560×1440 pixels. The menus are anchored to the edges of the screen and are scrollable, so that they are not cut off at other aspect ratios.

We host the WebGL application on the web space of the research group. The endpoints for the logging and the leaderboard (see Sections 3.2.8 and 3.2.9) only accept requests from this page, which is a simple protection against misuse.

3.2.12 Technical Challenges

A game in the browser brings several technical challenges. The biggest one concerns the mouse lock (pointer lock). The mouse lock hides the cursor and keeps it inside the game. It is necessary so that the player can control the camera with the mouse. However, browsers only allow the mouse lock after an action by the user (for example a click) and end it every time the player presses Escape or switches to another tab. The game has to detect this every time, pause itself and ask the player to restore the lock with a click. For this purpose, the game queries the actual state through a custom JavaScript plugin (see Section 3.2.11), since the state reported by Unity is not reliable in the browser. Also, the click that restores the lock must not count as a shot. Otherwise, the player could, for example, select an answer or use ammunition unintentionally.

Another problem is the order in which Unity starts the components when the game loads, since it does not guarantee a fixed order. Therefore, each component explicitly checks important states such as “round running” or “code entry open” before it reacts to them. This prevents, for example, the timer of a question from already running while the player is still entering the code.

The game also has to work with any window size and aspect ratio. Menus and the HUD therefore have to remain readable in all formats (see Section 3.2.11). Many of these problems only occurred in the browser and not in the Unity editor, where we developed the game. Therefore, we had to test every fix in a new WebGL build, which took about 1.5 hours each time.

3.3 Considered Features

3.3.1 Scope of the GDD

The GDD sorts the planned features into three priority levels:

- **Must Have:** features that the game must include.
- **Should Have:** features that are important but not essential and should be implemented if possible.
- **Extra:** features that are only implemented if there is time left.

We implemented all Must Haves:

- movement (see Section 3.2.1)
- shooting (see Section 3.2.5)
- displaying questions (see Section 3.2.4)
- targets (see Section 3.2.4)
- hit detection (see Section 3.2.5)
- feedback (see Section 3.2.5)
- points (see Section 3.2.6)
- round flow (see Section 3.1.3)
- restart and exit (see Section 3.2.10)

We implemented almost all Should Haves. Only three are missing: an adjustable number of targets, multiple levels and sound design. Of the six Extras, we implemented two, the leaderboard (see Section 3.2.9) and the switch between GCG and EVC as the *Chapter Select* (see Section 3.2.10). We only partially implemented a third one, the improved UI with progress indicators, since there is only a question counter ("Question 3/20"). Table 3.6 lists all Should Haves and Extras and whether we implemented them. Section 3.3.2 explains why we did not implement the remaining features.

Table 3.6: Should Have and Extra features of the GDD and their implementation.

Feature (GDD)	**Priority**	**Implemented**	**Section**
Mouse sensitivity	Should Have	yes	3.2.10
Weapon handling, reloading	Should Have	yes	3.2.5, 3.2.7
Settings menu	Should Have	yes (in the pause menu)	3.2.10
Difficulty levels	Should Have	yes	3.2.7
Moving targets	Should Have	yes (*Hard*)	3.2.7
Adjustable number of targets	Should Have	no	3.3.2
Timer	Should Have	yes (from *Medium* on)	3.2.7
Dark mode	Should Have	yes	3.1.6
Multiple levels or environments	Should Have	no	3.3.2
Scoreboard at the end of the round	Should Have	yes (end screen)	3.2.10
Log files	Should Have	yes	3.2.8
Tutorial	Should Have	yes	3.2.10
Hit effects	Should Have	yes	3.2.5
Sound design	Should Have	no	3.3.2
Weapon customization	Extra	no	3.3.2
Leaderboard	Extra	yes	3.2.9
Unlockable content	Extra	no	3.3.2
Detailed sound design	Extra	no	3.3.2
Switch between GCG and EVC	Extra	yes (*Chapter Select*)	3.2.10
Improved UI with progress indicators	Extra	partially	–

3.3.2 Features Not Implemented

This section explains why we did not implement the remaining features of the GDD:

- **Scope** (right mouse button): A scope is not an important feature for the gameplay. The GDD only listed it in the key bindings, but not in the priority lists. We also thought that it could even distract from the focus of the game, which is studying for the exam.
- **Weapon customization and unlockable content** (both Extra): Both are only nice-to-have features. Additionally, they contradict the goal of not overloading the game (see Section 3.1.6). The effort to implement them would be too high compared to their benefit for learning.
- **Multiple levels or environments** (Should Have): During the implementation, we decided that a single shooting range is sufficient for the round flow. Having everything in one scene is more intuitive and keeps the loading time in the browser short (see Sections 3.1.2 and 3.2.2).
- **Adjustable number of targets** (Should Have): In the end, this feature was not needed, since we decided to always use four targets. For questions with two answer

options, such as true/false questions, the game only shows the two middle targets (see Section 3.2.4).

- **Sound design** (Should Have and Extra): Sound would be nice, but unfortunately, we could not finish it in time. The volume setting in the pause menu already exists, but since there is no sound, it currently has no function. We kept it for possible future work (see Chapter 5).

3.3.3 Discarded Alternatives

This section explains which alternatives we considered during the implementation and why we dropped them:

- **Pause key (Tab instead of Escape or P):** The GDD planned the Escape key for the pause menu. In the browser, however, Escape ends the mouse lock (see Section 3.2.10). We then chose P, since Pause starts with P. During testing, we noticed that P is too far away from the other keys (W, A, S, D, R and F), so that players have to lift their hand to reach it. Therefore, we chose Tab, which is close to the other keys, is not used for anything else during a round and can be reached without lifting the hand.
- **Numeric input only on *Medium* and *Hard*:** On *Easy*, numeric questions appear as normal multiple-choice questions (see Section 3.2.4). We could also have used the numeric input on all difficulty levels. However, the numeric input is harder, because players have to know the exact result and cannot simply exclude wrong answers. It also takes longer to set the number. That is why we only use it on *Medium* and *Hard*, so that *Easy* can be played without pressure (see Section 3.1.5).
- **Resume button in the pause menu:** There was a Resume button in the pause menu to close the menu and continue the game. We removed it, since the browser accepts the mouse lock more reliably after a key press than after a click on a button (see Section 3.2.12). Also, it is more intuitive to open and close the menu with the same key. Therefore, the pause menu shows the hint "[Tab] RESUME" instead of a button.

3.4 Study Design and Procedure

3.4.1 Procedure

To answer RQ2 and RQ3, we conducted a user study with students of the EVC course. It was planned as a pre-post study over several weeks, with a questionnaire before and after using the game. It took place while the students prepared for the exam. There was no control group. The study consisted of three steps (see Figure 3.10). First, the participants filled in a pre-questionnaire. Then they used *SmartShot* for their exam

preparation. After the EVC retake exam on 29 September 2026, they were asked to fill in a post-questionnaire.

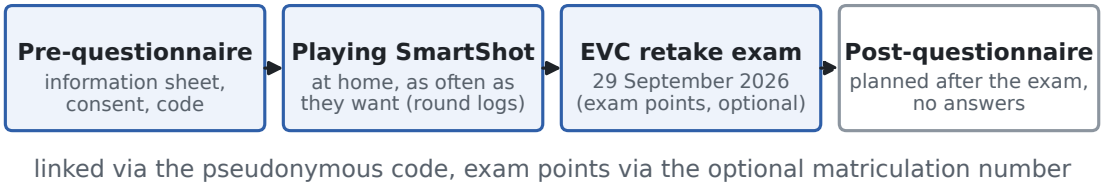


Figure 3.10: Procedure of the study. The post-questionnaire was planned but received no answers.

The pre-questionnaire starts with an information sheet. It explains the procedure, the voluntary participation, the risks and the data protection (see Section 3.4.4). It also mentions that players can switch to a mode without time limit (see Section 3.1.5). Before continuing, the participants had to confirm three things: that they accept the conditions, that they play on a desktop computer in fullscreen mode and that they test the game over several weeks and fill in the post-questionnaire afterwards. Then they created their code (see Section 3.2.8) and answered the questions (see Section 3.4.3). Submitting the questionnaire leads directly to the game.

There were no fixed tasks and no supervision. The participants could play *SmartShot* as often and as long as they wanted. Since the game runs in the browser, they could play it on their own computer, for example at home (see Section 3.2.11). The first answer to the pre-questionnaire came on 15 September and the last one on 25 September.

3.4.2 Participants

The target group of the study consisted of students of the EVC course, primarily those taking the retake exam on 29 September 2026. Therefore, the pre-questionnaire asked whether the participants were taking this exam. Students of GCG could also play, since the *Chapter Select* also contains the GCG questions. However, the focus of the study is EVC (see Section 3.1.1).

The advisor announced the study in the EVC course on TUWEL with a link to the pre-questionnaire. The link was also shared in topic-related group chats. Participation was voluntary and there was no reward. The study started on 15 September, two weeks before the retake exam. The number of participants and their background follow in Section 4.1.1.

3.4.3 Data Collection

We collected data from four sources:

- **Pre-questionnaire:** Demographic data (age, gender, occupation, previous computer graphics courses), study habits (hours per week, study methods, digital quizzes, learning games), gaming habits (first-person shooters, comfort with aiming, color vision deficiency), expectations for *SmartShot* and the familiarity with eight exam topics.
- **Game data:** The game logged every round (see Section 3.2.8). From these round logs, we can see how often and how long each participant played and how the accuracy and the points developed over the rounds.
- **Exam points:** The points of the retake exam (maximum 120) of the participants who entered their matriculation number (student ID number). For comparison, the advisor provided the mean, the median and the standard deviation of all 63 graded retake exams (see Section 3.4.5).
- **Post-questionnaire:** It was planned for after the retake exam. It asks for the exam grade (optional) and the perceived workload with the NASA Task Load Index (NASA-TLX) [17]. The NASA-TLX was developed at the National Aeronautics and Space Administration (NASA) and is a questionnaire that rates the workload of a task on six subscales. We used five of them (mental demand, temporal demand, performance, effort and frustration) and left out physical demand because the game is played sitting at a computer and needs hardly any physical effort. It also asks for the familiarity with the same eight topics, ratings of the game (perceived learning, engagement, clarity, overall experience), a comparison with reading slides, a recommendation rating and two open questions.

Most ratings used scales of four or five points. The recommendation rating in the post-questionnaire and the NASA-TLX subscales went from 0 to 10. Additionally, players could give feedback at any time through a short form that is linked in the pause menu.

For our evaluation, we use the exam points and not the course grade, because the course grade also includes the exercises and the best two of three tests and we only have the points of the retake exam. Additionally, taking the retake exam does not mean that a student failed before. Some students take it to improve their grade.

Because nobody filled in the post-questionnaire, not even after a reminder e-mail from the advisor, the evaluation is based on the other three sources. We use the pre-questionnaire to describe the participants (see Section 4.1.1). The game data answers RQ2 and the exam points answer RQ3. Section 4.2.3 discusses possible reasons why nobody filled in the post-questionnaire.

3.4.4 Linking and Data Protection

At the start of the pre-questionnaire, the participants saw an information sheet (see Appendix B). It describes the purpose and the procedure of the study, the collected data, its handling under the General Data Protection Regulation (GDPR) and the rights of the participants. Participation required agreeing to it. The sheet also points out that the questions in the game may contain errors.

We did not collect names or e-mail addresses. Instead, the pseudonymous code links the pre-questionnaire and the game logs (see Section 3.2.8). The game only sends round data after the player has entered a code and the leaderboard has its own consent in the game (see Section 3.2.9). The matriculation number was optional and was only used to look up the exam points. It was stored separately from the other answers and only the advisor had access to it. We only report the exam points in aggregated form and never per person. The participants could request the deletion of their data until 10 October 2026.

3.4.5 Evaluation

Because of the small number of participants, we only use descriptive statistics and no significance tests. Descriptive statistics only summarize the data. A significance test would check whether a difference is probably not caused by chance, but it needs more participants. Outliers are values that are far away from the others. In such a small sample, single outliers have a strong influence on the mean. Therefore, we report most values as median and range. The median is the middle value when all values are sorted and the range goes from the smallest to the largest value. The standard deviation describes how much the values spread around the mean. In the tables and figures, n is the number of participants. For RQ3, we only compare the exam points of the participants with the aggregated values of all exam participants.

CHAPTER 4

Results

This chapter presents the results of the study (see Section 3.4 for the design). The results are based on three data sources: the pre-questionnaire, the round logs of the game and the exam points of the retake exam on 29 September 2026. The post-questionnaire received no answers, so there is no rating of the game and no comparison of before and after playing. Section 4.1 first describes the participants. Then it presents the usage of the game and the performance in it for RQ2 and the exam results for RQ3. Section 4.2 discusses these results and the limitations of the study.

4.1 Evaluation

4.1.1 Participants

The pre-questionnaire received 10 submissions between 15 and 25 September 2026. One person submitted it twice, so we only kept the first submission. This leaves 9 participants. All of them confirmed the three conditions of the study (see Section 3.4.1).

The participants were between 20 and 41 years old, with a median of 23 years. Five were men, three were women and one preferred not to say. Seven were students and four were employed, so some of them study and work at the same time. Six of the 9 participants planned to take the EVC retake exam on 29 September.

The previous knowledge of computer graphics was mixed. Four participants had already completed EVC, two GCG, two *Programmiertechniken für Visual Computing* (PTVC, *Programming Techniques for Visual Computing*) and two a computer graphics lecture of the master's program. One had taken a basic computer graphics course at another university. Four had not completed any computer graphics course yet. Since multiple answers were possible, some participants are counted more than once.

Before playing, the participants rated how familiar they were with eight exam topics on a scale from 1 (not at all) to 4 (very familiar). Figure 4.1 shows the mean rating per topic. Over all topics, the mean was 2.36, which lies between slightly and moderately familiar. The participants felt most familiar with computer graphics concepts and terminology, color models and color spaces and textures (mean 2.56 each). They felt least familiar with sampling theory and curves (mean 2.11 each).

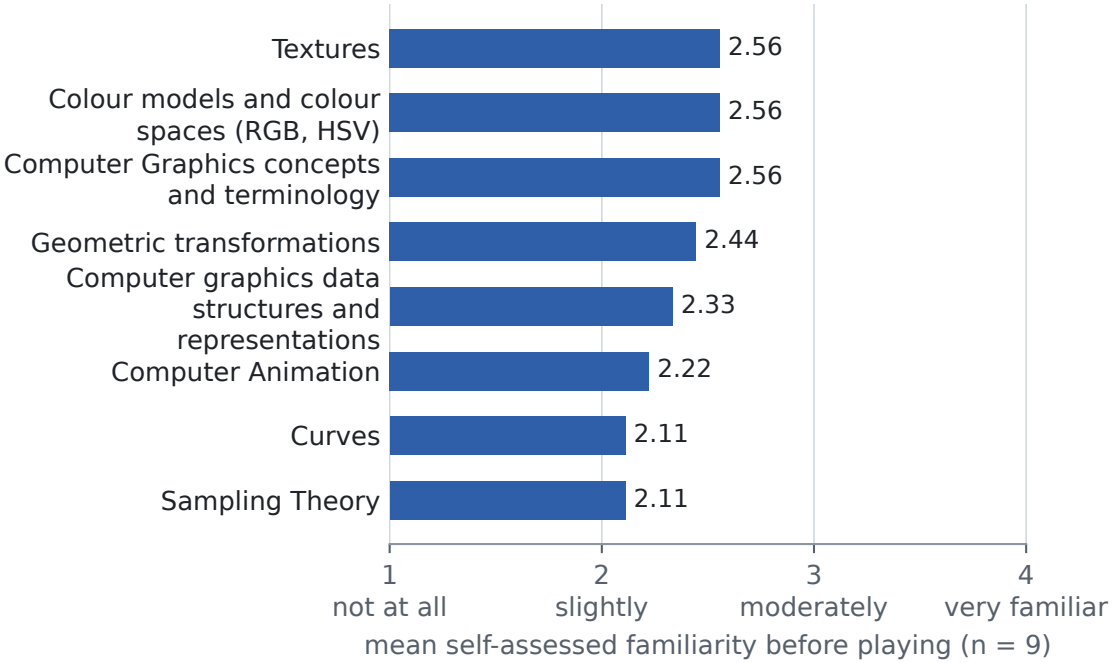


Figure 4.1: Self-reported familiarity with the exam topics before playing (n = 9).

Most participants study by reading textbooks or lecture notes (8 of 9) and by watching instructional videos (7). Six also use practice questions or quizzes, but nobody uses educational games or simulations. Five had already used digital quiz tools such as Anki, VOWI (a wiki by students of TU Wien), Quizlet or StudySmarter. On a scale from 1 to 5, their confidence in learning independently with digital materials was rather high (mean 3.56).

Seven participants had played a learning game at least once, most often quiz games like *Kahoot!* (6). Seven also play first-person shooters at least rarely. Their comfort with aiming in first person was high (mean 3.33 on a scale from 1 to 4), so aiming was probably not a big obstacle for most of them. None of the participants has a color vision deficiency, which matters for the green and red feedback (see Section 3.2.5).

The expectations towards *SmartShot* were rather cautious. On a scale from 1 to 5, the expected engagement (mean 2.67) and the expected usefulness of *SmartShot* (mean 2.56) were slightly below the middle. The general usefulness of learning games for university was rated similarly (mean 2.89). Six participants preferred playing without a time limit,

which fits the *Easy* level (see Section 3.1.5). In the free text, participants expected to actively recall the course material, to memorize it more easily through the playful setup and to pass the exam. Two people mentioned that tiredness because of full-time work or lack of sleep may affect their experience.

4.1.2 Game Usage

All 9 participants played *SmartShot* at least once. All their rounds took place between 15 and 25 September, so before the exam on 29 September. Together, they started 33 rounds. Of these, 20 were completed and 13 were aborted before the last question. Seven participants completed at least one round. The other two only started rounds and aborted them. Most aborted rounds were stopped right after the first question.

Figure 4.2 shows how much the participants used the game. A typical participant started 2 rounds, completed them and only played for a single day. Only 3 participants completed three or more rounds. So most participants did not spread their practice over several weeks as planned. In total, the participants answered 437 questions and played for 4.3 hours. This playing time does not include the time in the pause menu (see Section 3.2.2).

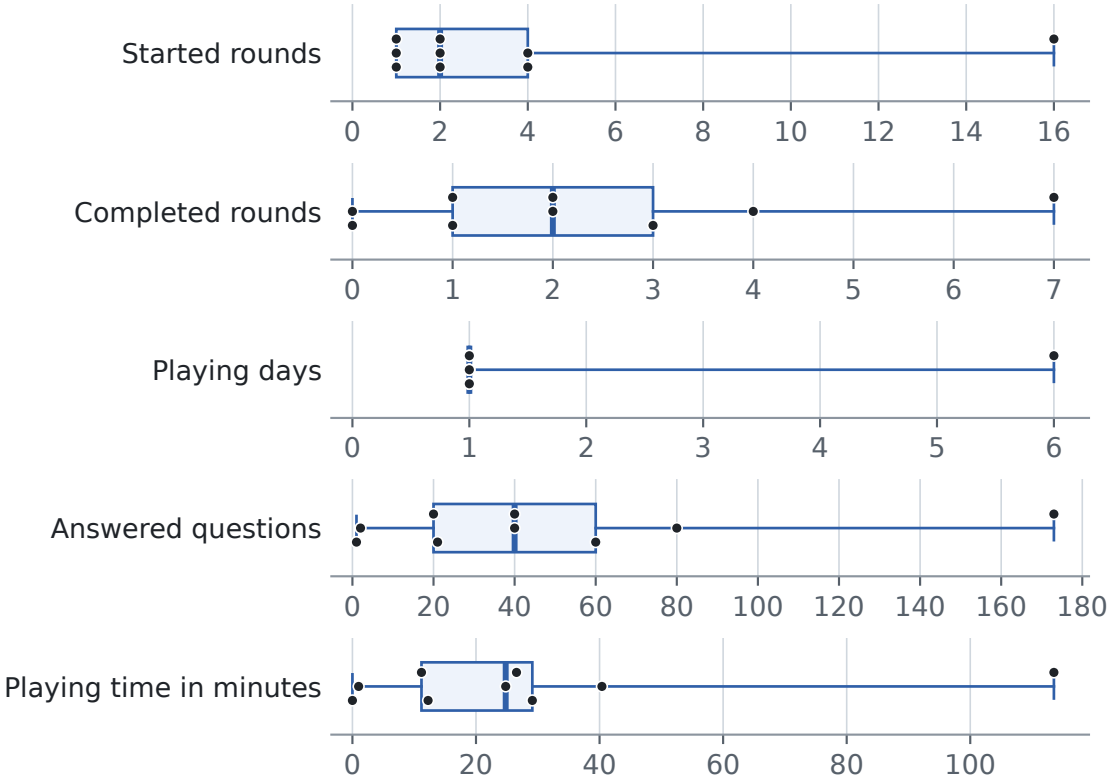


Figure 4.2: Usage of *SmartShot* per participant (n = 9). Each dot is one participant. The box contains the middle half of the participants, the thick line marks the median and the whiskers reach from the lowest to the highest value.

A completed round took a median of 5.9 minutes. The longest round took 113.5 minutes, even though the time in the pause menu is not counted. This single round raises the mean duration of a round to 12.6 minutes, which is why we report medians. Six of the 7 participants with a completed round agreed to appear on the leaderboard (see Section 3.2.9).

Additionally, 8 other codes played *SmartShot* without filling in the pre-questionnaire. They played 21 rounds (12 completed) and answered 291 questions in about 3.1 hours between 11 September and 24 October. Five of these rounds took place after the exam. Since these players did not give consent to the study, they are not part of the analysis. Still, this shows that at least 17 people played *SmartShot*.

4.1.3 Performance in the Game

To describe the performance, we use the accuracy of a round. It is the share of questions that were answered completely correctly, so with all correct and no wrong answers (see Section 3.2.6). We only look at the 20 completed rounds, since most aborted rounds ended after the first question.

In a typical completed round, the participants answered 12 or 13 of the 20 questions completely correctly. This is a median accuracy of 62.5 %. The weakest round had an accuracy of 25 % and the best one 85 %. Nobody played a perfect round, which means a round with all 20 questions completely correct.

To see whether the participants got better, we compare the first and the last completed round of each participant. This is only possible for participants with at least two completed rounds. Of the 7 participants who completed a round, only 5 did so more than once. For these 5 participants, the median accuracy rose from 45 % in the first round to 68.4 % in the last round. Three of them improved and two stayed the same. Nobody got worse.

These numbers have to be interpreted carefully. The questions are drawn randomly and the game does not log the difficulty level or the chapters. So the last round could also have had easier questions or other chapters than the first one. Additionally, 5 participants are very few (see Section 4.2.3).

4.1.4 Exam Results

Six of the 9 participants said in the pre-questionnaire that they would take the retake exam on 29 September 2026. Giving the matriculation number was optional and four participants did so. The advisor looked up their points. Three of them took the exam and one did not. So we have exam points for 3 participants. We do not know whether the participants without a matriculation number took the exam.

Table 4.1 compares these 3 participants with all 63 graded exams of the retake exam. All 3 participants played *SmartShot* before the exam. All of them scored above the mean of all exams and 2 of them also above the median. Their mean of 72.1 points (60 % of the

maximum) is 13 points above the mean of all exams. The 3 participants are also part of the 63 exams.

Table 4.1: Points in the retake exam (maximum 120 points). For the 3 participants, we only report values for the whole group, so there is no median and no standard deviation. Otherwise, these values could reveal the points of a single person.

	All exams	**Participants**
Number of exams	63	3
Mean	59.2	72.1
Median	61.6	–
Standard deviation	18.5	–
Above the mean of all exams	–	3 of 3
Above the median of all exams	–	2 of 3

However, this result does not allow a conclusion about the effect of the game. Even if we picked 3 random exams, 2 or more of them would be above the median in half of the cases. Additionally, students who voluntarily take part in a study and give their matriculation number may be more motivated anyway.

4.2 Discussion

This section discusses the results with respect to the research questions. RQ2 asked how students use *SmartShot* during their exam preparation and how their performance in the game develops. RQ3 asked how the participants performed in the retake exam compared to all exam participants. Finally, we discuss the limitations of the study.

4.2.1 Game Usage and Performance (RQ2)

The participants used *SmartShot* rather briefly. Most of them played one or two rounds on a single day before the exam, for about 25 minutes in total. Only a few played more often. In a typical round, they answered around 60 % of the questions completely correctly. The participants who played several rounds improved or stayed the same.

The improvement is consistent with the testing effect, because answering questions is a form of practice testing (see Section 2.2). However, it can also come from getting used to the controls and the scoring or from easier questions in the later rounds. Wouters et al. [5] found larger learning gains when there were several training sessions (see Section 2.1). Most of our participants only played on one day, so *SmartShot* could not show this advantage. The short usage fits the cautious expectations in the pre-questionnaire (see Section 4.1.1). Additionally, the study only started two weeks before the exam, so there was little time for regular use.

On the technical side, the game ran in the browser on the participants' own computers. Nobody reported technical problems, neither via the feedback form nor by e-mail.

However, this does not prove that there were none. Some of the aborted rounds could also have technical reasons that were not reported.

4.2.2 Exam Results (RQ3)

The 3 participants with exam points scored above the mean of all exam participants. Because of the small number, the self-selection (the participants decided themselves to take part) and the missing control group, this does not show that *SmartShot* caused the good result. It only shows that these participants did well in the exam after using the game. To measure an effect of the game on the exam, a study with a control group would be needed, like the one by Moradi et al. [14] (see Section 2.4).

4.2.3 Limitations

The biggest limitation is the missing post-questionnaire. Nobody filled it in, not even after a reminder e-mail from the advisor. Therefore, we do not know whether the students found the game engaging and useful, how high the workload was and whether their familiarity with the topics changed. One possible reason is that it was only due after the exam. Additionally, there was no reward for filling it in.

The sample is also small: 9 participants, of whom only 5 completed several rounds and only 3 have exam points. The participation was voluntary, so probably rather motivated students took part. There is no control group, so the exam points cannot be attributed to the game.

The difficulty level and the chapters were not logged, although the information sheet lists them (see Appendix B). So we do not know which level was played and the development in Section 4.1.3 may depend on the drawn questions. The information sheet also lists the level of study and the field of study, but the pre-questionnaire did not ask for them. The usage period was also short. It was planned over several weeks, but most participants played on a single day, since the study could only start two weeks before the exam.

Further, 8 codes played without filling in the pre-questionnaire and are not analyzed, so the real usage of the game is higher than the study shows. Of the 33 rounds, 13 were aborted, mostly after the first question. The reason for this is unknown. The questions come from old exams, so participants who had already practiced with these exams may have known some answers. In the pre-questionnaire, the highest option of the question about aiming was labeled “5” on a scale from 1 to 4, so we counted it as 4.

CHAPTER 5

Conclusion

This thesis presented *SmartShot*, a browser game for exam preparation in computer graphics. The game contains 640 real exam questions of the courses EVC and GCG. The player answers them by shooting targets in a shooting range and gets the correct solution right after every question. To answer RQ1, Chapter 3 described the design and the implementation of the game. The most important design decisions were answering by shooting as the core mechanic, the immediate feedback, the scoring based on the EVC exam scheme, three difficulty levels, the *Chapter Select* and an optional leaderboard.

To answer RQ2 and RQ3, we conducted a study with 9 EVC students before the retake exam (see Chapter 4). Most participants used *SmartShot* only briefly, with a median of 2 rounds on a single day. The median accuracy of a completed round was 62.5 %. The 5 participants with several completed rounds improved or stayed the same. The 3 participants with exam points scored above the mean of all exam participants. However, because of the small sample, the self-selection and the missing control group, we cannot draw conclusions about the effect of the game. Still, at least 17 people played *SmartShot* (see Section 4.1.2).

There are several ways to continue this work. First, the study should be repeated with a better setup. It should start earlier in the semester, so that students can use the game over several weeks. The post-questionnaire should be part of the game or of a supervised session before the exam, so that it actually gets answers (see Section 4.2.3). A larger sample and a control group would also make it possible to measure the effect of the game on the exam, similar to the study by Moradi et al. [14] (see Section 2.4).

Second, the logging should be improved. The round logs should include the difficulty level and the chapters (see Section 4.2.3). Then the development of the performance in the game could be interpreted better. It would also help to log every single question and whether it was answered correctly. With this, we could see which topics and questions are especially hard for the students. Additionally, the leaderboard should use a database

with one record per player, so that two rounds ending at the same time cannot overwrite each other (see Section 3.2.9).

Third, the game itself can be improved. The questions that were translated with AI should be replaced with the official English versions from the exams (see Section 3.1.4). Sound is still missing, although the volume setting in the pause menu already exists (see Section 3.3.2). Finally, the questions are stored per chapter in JSON files (see Section 3.2.3). So further computer graphics courses could be added without changing the code of the game.

APPENDIX A

Example of the Question Data

Each chapter is stored in its own JSON file, which contains a list of questions (see Section 3.2.3). The following example shows the structure of such a file with a single question. It is the multiple-choice question with three correct options from Figure 1.1.

```
{
  "questions": [
    {
      "questionText":
        "Which data structure enables efficient spatial search?",
      "answerOptions": ["Regular Grid", "Octree", "Kd-tree", "B-Rep"],
      "correctIndices": [0, 1, 2],
      "points": 3,
      "isCalculation": false
    }
  ]
}
```

A numeric question additionally contains the field `isNumericAnswer` and the correct number in `correctNumericAnswer`. The answer options are only used on *Easy*, where the game shows the question as a regular multiple-choice question (see Section 3.2.4). The following question is the one from Figure 3.4b.

```
{
  "questionText":
    "Assume the gradient was (-1.6, 0.3). Calculate the edge strength.",
  "answerOptions": ["-1.63", "-1.90", "0.65", "1.14",
                    "1.63", "1.90", "-0.65", "-1.14"],
  "correctIndices": [4],
  "points": 3,
  "isCalculation": true,
  "isNumericAnswer": true,
  "correctNumericAnswer": 1.63
}
```

APPENDIX B

Study Information Sheet

The following information sheet was shown on the first page of the pre-questionnaire (see Section 3.4). It is reproduced here as it was shown to the participants.

Information Sheet for Participation in a Scientific Study Conducted as Part of a Bachelor's Thesis: SmartShot: A Serious Learning Game on Computer Graphics

B.1 Study Information

Investigator(s): Kevin Chelappurath (e12119880@student.tuwien.ac.at)

Supervision / Principal Investigator: Proj.Ass.in Annalena Ulschmid, MSc BSc (annalena.ulschmid@tuwien.ac.at, +43 1 58801 18646), TU Wien – Faculty of Informatics, Institute of Visual Computing & Human-Centered Technology, Favoritenstraße 9-11 / E193-02, A-1040 Vienna, Austria

Data Protection Officer: Mag. Christina Thirsfeld (datenschutz@tuwien.ac.at, +43 1 58801 41000), TU Wien, Karlsplatz 13/018, A-1040 Vienna, Austria

B.2 Purpose and Procedure of the Study

The aim of this study is to investigate whether a first-person-shooter-based quiz game covering computer graphics and computer vision course content (EVC, GCG) can serve as a meaningful tool for exam preparation – either as a supplement or as an alternative to traditional study methods. As part of the study, you will use SmartShot over the course of several weeks to prepare for your exams. Before your first session, you will complete this short questionnaire; after the testing period, you will be asked to complete a follow-up (post) questionnaire. The purpose of the study is not to evaluate your individual performance, but the contribution of our learning game.

The estimated duration to complete this questionnaire is approximately 10 minutes. The amount of time you spend playing the game is entirely up to you.

Please play the game on a desktop PC, as it is not optimized for mobile devices, enable hardware/graphics acceleration for your browser if you notice a low framerate, and enable fullscreen mode.

B.3 Voluntary Participation

Participation in this study is entirely voluntary. You may withdraw your consent at any time and without giving reasons, with no disadvantages arising for you as a result. You may discontinue the study at any time.

B.4 Potential Risks and Benefits

Participation in this study involves minimal risk — meaning the risks are no greater than those encountered in ordinary daily life. Using a desktop PC for a longer time can lead to eye strain, headache or fatigue and musculoskeletal discomfort. We recommend taking breaks from time to time, stretching and looking at objects in the distance.

To the best of our knowledge the game does not contain any disturbing or photosensitive content (flashing lights, strobing/high-contrast patterns). If you are unsure or have a family history of epilepsy we discourage participation in this user study. You can adjust lighting, brightness and crosshair color in the game.

As the game serves as an exam-preparation tool and offers an optional timed mode, participants with elevated test anxiety may experience transient anxiety symptoms similar to those in a real examination situation. You may switch to the untimed mode, interrupt or terminate participation at any point, without providing a reason and without any disadvantage. If you experience persistent distress after the session, please contact the study team; we can provide information on the counseling services offered by the TU Wien.

The study contributes to scientific knowledge in the field of computer science education. In particular, it will broaden the understanding of employing learning games in computer graphics, which is both an underexplored field in serious gaming and closely related in its learning goals to creating digital games itself.

B.5 Data Protection and Confidentiality

The collection, processing, and storage of your personal data is carried out in accordance with the General Data Protection Regulation (GDPR) and applicable national data protection law. The following data will be collected as part of the study:

- Demographic information: age, gender, level of study, field of study, prior exposure to EVC/GCG course content
- Measurement data from playing the game: session duration, questions answered, correctness of answers, difficulty level selected, chapters/topics practiced
- Questionnaire responses: pre-game and post-game questionnaire answers regarding expectations, motivation, perceived difficulty and task load, usability and learning outcome.

Your data will be processed in pseudonymized or anonymized form. This means that your information is separated from your personal data such as student ID and assigned a code. You may voluntarily state your student ID, which we will use to link your exam grade to your survey data. During evaluation we will remove any link between your assigned code and your student ID, such that no conclusions are linked to your identity. Only the principal investigator can link the data back to your person, and only via a separately and securely stored code list. The results of the study will be published exclusively in anonymized form; no conclusions can be drawn about your identity.

The anonymized data collected and the results derived from them will be used for scientific publications in relevant academic journals. If you wish to have the data collected about you in the course of this study deleted, you may send an e-mail to the principal investigator no later than 10.10.2026, stating your user ID.

B.6 Your Rights Under the GDPR

You have the right at any time to:

- Access the data stored about you (Art. 15 GDPR)
- Rectification of inaccurate data (Art. 16 GDPR)
- Erasure of your data (Art. 17 GDPR)
- Restriction of processing (Art. 18 GDPR)
- Data portability (Art. 20 GDPR)
- Object to processing (Art. 21 GDPR)
- Lodge a complaint with a data protection supervisory authority (Art. 77 GDPR)

You may withdraw your consent at any time with effect for the future. The lawfulness of the processing carried out prior to the withdrawal remains unaffected. Please note that subsequent deletion of your data is only possible as long as your data can still be attributed to you personally (i.e., before full anonymization has taken place).

B.7 Reliability of the learning game's content

The content of the learning game has been reviewed to the best of our ability. The questions were derived from past exams made available via TUWEL. However, these source materials may themselves contain errors. We therefore cannot guarantee the accuracy or completeness of the content, and in particular it is possible that individual answer options are incorrectly labelled as correct or incorrect. The learning game is intended solely as a supplementary study aid and does not constitute an authoritative source. Its content cannot be invoked as grounds for contesting a grade in any future examination. Should you identify an incorrectly labelled answer, we would be grateful if you reported it via the in-game feedback form.

B.8 Required Confirmations

- I have read the above information and accept the conditions.
- I understand that I should play the game on a desktop PC and enable fullscreen mode.
- I understand that I will test SmartShot potentially over the course of several weeks while using it for exam preparation, and that I will be asked to complete a follow-up (post) questionnaire afterwards.

Overview of Generative AI Tools Used

We used Claude (Anthropic) [18] as a tool in this thesis, specifically the models Claude Sonnet 5, Claude Sonnet 5.5 and Claude Opus 5.5. For the text, we used Claude for proofreading and for bullet points, which we then rewrote in our own words. During the development of *SmartShot*, we used Claude Code as a programming assistant. Claude also created the diagrams and the scripts for evaluating the study data. Additionally, Claude generated the pixel-art textures of the sun and the moon and translated the German exam questions into English. The concept, the game design, the 3D models, the study and all decisions about the content are our own. We checked every result of the AI and are responsible for the whole thesis.

List of Figures

1.1 A multiple-choice question in *SmartShot* 2

3.1 Overview of the shooting hall 10
3.2 Flow of one round 11
3.3 Main components and their connection to the server 15
3.4 Question types besides multiple choice 17
3.5 Selecting and submitting answers 18
3.6 Code entry at the start of the game 21
3.7 The *Chapter Select* 23
3.8 End screen after a round 24
3.9 The leaderboard 24
3.10 Procedure of the study 29

4.1 Familiarity with the exam topics before playing 33
4.2 Usage of *SmartShot* per participant (n = 9) 34

List of Tables

3.1 Design goals and their implementation in *SmartShot* 9
3.2 Main components of *SmartShot* and their tasks 14
3.3 Fields of a question in the JSON files 16
3.4 Size of the question bank 16
3.5 Differences between the difficulty levels 19
3.6 Should Have and Extra features of the GDD and their implementation . . 27

4.1 Points in the retake exam (maximum 120 points) 36

Acronyms

AI Artificial Intelligence. 9–12, 39

EVC Einführung in Visual Computing (Introduction to Visual Computing). 2, 3, 5, 7–9, 11, 12, 15, 16, 19, 22, 26–29, 32, 38

GCG Grundlagen der Computergraphik (Fundamentals of Computer Graphics). 2, 7, 8, 11, 12, 15–17, 22, 26, 27, 29, 32, 38

GDD Game Design Document. 9, 12, 26–28

GDPR General Data Protection Regulation. 31

HDRP High Definition Render Pipeline. 9

HTML HyperText Markup Language. 23

HTTP Hypertext Transfer Protocol. 20

HUD Heads-Up Display. 13, 25

ID Identifier. 21, 30

JSON JavaScript Object Notation. 15, 18, 20, 21, 39

NASA National Aeronautics and Space Administration. 30

NASA-TLX NASA Task Load Index. 30

PTVC Programmiertechniken für Visual Computing (Programming Techniques for Visual Computing). 32

RQ Research Question. 1, 2, 8, 28, 30–32, 36, 38

UI User Interface. 13, 25–27

URP Universal Render Pipeline. 9

WebGL Web Graphics Library. 1, 9, 15, 23, 25

Bibliography

[1] Unity Technologies. Unity Real-Time Development Platform. https://unity.com/. Accessed: 2026-10-29.

[2] David R. Michael and Sandra L. Chen. *Serious Games: Games That Educate, Train, and Inform*. Thomson Course Technology PTR, Boston, MA, 2005. ISBN 978-1-59200-622-9, https://books.google.com/books?vid=ISBN9781592006229.

[3] Henry L. Roediger and Jeffrey D. Karpicke. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, 17(3):249–255, 2006. https://doi.org/10.1111/j.1467-9280.2006.01693.x.

[4] Olusola O. Adesope, Dominic A. Trevisan, and Narayankripa Sundararajan. Rethinking the use of tests: A meta-analysis of practice testing. *Review of Educational Research*, 87(3):659–701, 2017. https://doi.org/10.3102/0034654316689306.

[5] Pieter Wouters, Christof van Nimwegen, Herre van Oostendorp, and Erik D. van der Spek. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2):249–265, 2013. https://doi.org/10.1037/a0031311.

[6] Michael Sailer and Lisa Homner. The gamification of learning: A meta-analysis. *Educational Psychology Review*, 32(1):77–112, 2020. https://doi.org/10.1007/s10648-019-09498-w.

[7] Richard E. Clark. Media will never influence learning. *Educational Technology Research and Development*, 42(2):21–29, 1994. https://doi.org/10.1007/BF02299088.

[8] M. P. Jacob Habgood and Shaaron E. Ainsworth. Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *Journal of the Learning Sciences*, 20(2):169–206, 2011. https://doi.org/10.1080/10508406.2010.508029.

[9] Andrew C. Butler and Henry L. Roediger. Feedback enhances the positive effects and reduces the negative effects of multiple-choice testing. *Memory & Cognition*, 36(3):604–616, 2008. https://doi.org/10.3758/MC.36.3.604.

[10] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15, 2011. https://doi.org/10.1145/2181037.2181040.

[11] Michael D. Hanus and Jesse Fox. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80:152–161, 2015. https://doi.org/10.1016/j.compedu.2014.08.019.

[12] Alf Inge Wang and Rabail Tahir. The effect of using Kahoot! for learning – a literature review. *Computers & Education*, 149:103818, 2020. https://doi.org/10.1016/j.compedu.2020.103818.

[13] Paulo Eduardo Battistella and Christiane Gresse von Wangenheim. Games for teaching computing in higher education – a systematic review. *IEEE Technology and Engineering Education*, 9(1):8–30, 2016. https://softwarequalitygroup.paginas.ufsc.br/files/2020/02/ITEE-Games-for-Teaching-Computing-in-Higher-Education_Vdraft.pdf.

[14] Meisam Moradi, Nurul Fazmidar Binti Mohd Noor, and Rukaini Binti Haji Abdullah. The effects of problem-based serious games on learning 3D computer graphics. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 46(4):989–1004, 2022. https://doi.org/10.1007/s40998-022-00526-0.

[15] Epic Games. HTML5 Game Development, Unreal Engine 4.26 Documentation. https://web.archive.org/web/20210613175533/https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/HTML5/. Archived copy of 2021-06-13 in the Internet Archive. Accessed: 2026-10-31.

[16] Unity Technologies. Starter Assets: FirstPerson, Unity Asset Store. https://assetstore.unity.com/packages/essentials/starter-assets-firstperson-updates-in-new-charactercontroller-pa-196525. Accessed: 2026-10-29.

[17] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in*

Psychology, pages 139–183. North-Holland, 1988. https://doi.org/10.1016/S0166-4115(08)62386-9.

[18] Anthropic. Claude and Claude Code. https://www.anthropic.com/claude. Accessed: 2026-10-29.