ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός και Ανάπτυξη Ψηφιακού Παιχνιδιού Μάθησης

Design and Development of Digital Learning Game

Θεοφίλου Στυλιανός
Αριθμός Μητρώου: 1072791

Επιβλέπων
Σιντόρης Χρήστος, Ε.ΔΙ.Π.

Μέλη Επιτροπής Αξιολόγησης
Σγάρμπας Κυριάκος, Καθηγητής

Πάτρα
Δεκέμβριος 2024

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

Σχεδιασμός και Ανάπτυξη Ψηφιακού Παιχνιδιού Μάθησης

του φοιτητή του τμήματος Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών

Θεοφίλου Στυλιανού

Αριθμός Μητρώου: 1072791

παρουσιάστηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών στις

.... / .... / ....

Ο Επιβλέπων

Σιντόρης Χρήστος, Ε.ΔΙ.Π.

Ο Διευθυντής του Τομέα

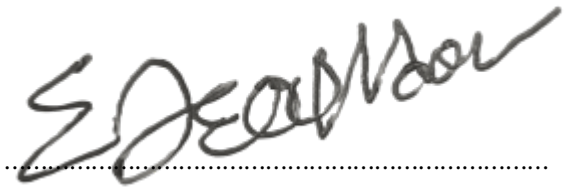
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του τμήματος, του επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Υπεύθυνη Δήλωση

Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην διπλωματική εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών.

(Υπογραφή)



.....

Θεοφίλου Στυλιανός

Σύνοψη

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Λέξεις-κλειδιά: Μάθηση βασισμένη στο παιχνίδι, Κβαντική Υπολογιστική, Κβαντική Μηχανική, Παιχνίδι για κινητά, Flutter

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Keywords: Game-based Learning, Quantum Computing, Quantum Mechanics, Mobile Game, Flutter

Ευχαριστίες

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

1 1 Introduction

1.1 Motivation

Quantum computing utilizes the principles of quantum mechanics to process information and solve complex problems exponentially faster that classical computers. Quantum bits can exist in multiple states at the same time, offering great computational power, beyond the limits of classical computers. The development and widespread use of quantum computing can help in fields such as pharmaceuticals, cryptography, artificial intelligence, materials science and more. (Taylor 2024), (Nagappan 2023), (Wootton 2017) Since quantum computing represents a new era for computer science, opening up new prospects for accelerating scientific discoveries, learning the basic principles of quantum computing is extremely important.

1.2 What is Game-Based Learning

Game-based learning is a very old practice; it did not start with the advance of modern technology. It can be defined as the technique of being educated by playing games. It integrates the characteristics and principles of games such as elements of competition, rewards and active user engagement, into learning activities. Games can be an interactive tool that can simplify challenging concepts and help learners understand complex ideas, engaging them into educational content. (Ledda 2012), (Wirtz 2023)

1.3 Benefits of Game Based Learning

First, game-based learning is more appealing to children, as it appears to be a game on the surface, but in the background it has the ability to stimulate children' s curiosity and capture their imagination. It is a friendlier and more accessible mean of engaging young learners with a subject than traditional methods, as it is fun and motivating.

Game-based learning also has the ability to enhance critical thinking and problem-solving, as they involve human instinct to compete and desire to succeed. Because learners often compete with other players, they have to collaborate and share ideas. They must listen to and evaluate the opinions of other players and take into account the tactics of opposing teams.

Games often require users to react quicker to stimuli, make critical decisions in a short period of time and combine knowledge acquired during the game to solve complex problems. Due to their repetitive and interactive nature, they have the ability to improve retention and increase the brain' s capacity to memorize things.

Also, as games are flexible, they can be adapted to different learning styles, levels and paces, meeting individual needs and can also give instant feedback about where gaps in knowledge are or provide specific tasks for the user to help cover these areas. In this way, they can further help learners to identify their strengths and weaknesses.

Comparing games to traditional textbooks, although the latter have been used for many years with success, their revision and renewal takes a long time and is difficult and costly. The cost of reprinting, redistributing and recycling or storing old textbooks must be taken into account. Even in the case of digital textbooks, there is a significant cost of disposal and renewal. By contrast, games are very versatile, their rules can be adapted easily, and their content can be changed quickly to keep pace with technological and scientific progress.

In summary, game-based learning offers a modern, engaging and flexible approach to education. It is a great way to improve learners' critical thinking and problem-solving skills, boost their creativity and keep them engaged and motivated. Also, unlike traditional textbooks, it can be quickly and cost-effectively updated, in order to reflect new information and technological progress.

(Harding 2023), (Wirtz 2023)

1.4 Research Objectives

The aim of this thesis is to familiarize the learners with the basic principles of quantum computing, such as quantum bits and quantum gates. They should not be distracted or get tired due to the complexity of the game. The aim is to design a simple and accessible educational game, with few rules and clear objectives.

In order to play this educational game, one does not need to have a university background in mathematics or a strong background in quantum physics. The game can be played by anyone who is interested in learning how quantum gates work and how they affect quantum bits.

In addition, it is desirable that the game could be played at any time and in any place, without the need of equipment or a computer. The aim is that the user can play even when he has limited time (e.g. travelling, waiting for public transportation, etc.), and for the game to be suitable for playing in a class, in the context of a lecture.

For these reasons, a classic digital game, for mobile devices, with simple mechanisms and low complexity should be chosen, whose rules and objectives should be adapted to the topic of quantum

computing.

1.5 Thesis Structure

2 Literature Review

2.1 Computer Science Games for Higher Education

https://www.researchgate.net/publication/325046233_Games_for_Teaching_Computing_in_Higher_Education_-_A_Systematic_Review

2.2 Games for Quantum Computing and Quantum Physics

In the literature we can find many games for quantum computers or quantum physics, which have been developed in different contexts. There are games developed in the context of competitions, as university coursework, by research groups, by universities, by companies, by independent enthusiasts and by professors or professional engineers. By extension, there are games that have not been developed for serious purposes, competition games, games designed for research purposes, commercial games and educational games. (Piispanen et al. 2023), (Seskir et al. 2022)

The earliest quantum game that can be found in the literature is an Atari arcade game called 'Quantum' created in 1982 (Piispanen et al. 2023). To date, more than 300 games related to quantum physics, digital and not, have been created (Piispanen et al. 2024). This section summarizes digital games that have been created for educational purposes by companies or universities and are still playable.

To prwto paixnidi ⋯.. introduction: https://www.researchgate.net/publication/373686105_The_History_of_Quantum_G –> paragraph IIIA –> READ THIS ARTICLE, LOT' S OF DATAAAA!!!

add this one https://mmpant.com/author/mmpant/ –> https://mmpant.com/2024/04/05/quantum-games/

list: (Piispanen 2024)

COMPANIES

- IBM
  - Quantum Composer: https://quantum.ibm.com/composer/files/new
  - Hello Quantum Project: IBM & University of Basel

    - https://hello-quantum.soft112.com/
    - (Seskir et al. 2022) –> 4.1 Hello Quantum/Hello Qiskit

- Google
  - The Qubit Game: https://quantumai.google/education/thequbitgame, https://blog.google/technology/researc quantum-day-meet-our-researchers-and-play-qubit-game/
  - Experiments with Google –> Quantum Computing Playground: https://experiments.withgoogle.com/quantu computing-playground, https://en.wikipedia.org/wiki/Google_Chrome_Experiments

- QPlayLearn
  - https://qplaylearn.com/about
  - Uses interactive tools to make the learning process more effective and entertaining for different target groups.
  - Platform with web-based educational games. Created by universities or individuals
    - Particle in a box: https://qplaylearn.com/projects/quantum-physics
    - Quantum Playground: https://qplaylearn.com/projects/quantum-state
    - Q|Cards>: https://qplaylearn.com/projects/qubit
    - TiqTaqToe: https://qplaylearn.com/projects/superposition
    - Psi and Delta: https://qplaylearn.com/projects/wave-like-behaviour
    - Quantum Solitaire: https://qplaylearn.com/projects/entanglement
    - Escape Quantum: https://qplaylearn.com/projects/quantum-measurement
    - Potatoes Quest: https://qplaylearn.com/projects/tunneling
    - QWiz: https://qplaylearn.com/projects/quantum-technologies
    - Saving Photonland: https://qplaylearn.com/projects/time-frequency-modes
  - (Seskir et al. 2022) –> 4.3

- Quarks Interactive
  - Game: Quantum Odyssey
  - https://ed.quantum.ieee.org/quantum-odyssey/
  - https://www.quarksinteractive.com/
  - (Seskir et al. 2022) –> 4.5

- Canon Lab Org
  - https://www.canonlab.org/quander
  - (Piispanen 2024) (Piispanen et al. 2023)

- Quantum Flytrap
  - https://www.quantumlah.org/

  - https://quantumflytrap.com/virtual-lab
  - Supported by Centre for quantum Technologies - National University of Singapore
  - https://www.quantumlah.org/
  - (Seskir et al. 2022) -> 4.4

UNIVERSITIES

- Aalto University

  - https://quantumgames.aalto.fi/
  - Course offered by Aalto University
  - Games are not designed for educational purposes.
  - This course is designed to teach students how to design and develop games and also learn the basic concepts of quantum computing.
  - (Piispanen et al. 2023) (Piispanen 2024)

- Science At Home

  - https://www.scienceathome.org/about-us/
  - Aarhus University
  - Browser and desktop educational games
  - Diverse team of scientists, designers and game developers that create scientific games, aiming at teaching by game-play
  - (Seskir et al. 2022) -> 4.6

- Applied Research Laboratories at the University of Texas at Austin

  - Virtual Quantum Optics Library
  - (Seskir et al. 2022) -> 4.7

Consider: 2 tables -> games & virtual labs -> game table: name, creator, url, platform (desktop/web), concept (entanglement, superposition etc), category/genre/subgenre

-> Say that game categories are described later on thesis

!Consider a chapter explaining how an educational game should be designed! https://papers.ssrn.com/sol3/papers.cfm?abstr

2.2.1 Desktop & Web Games

2.2.2 Mobile Games

Android

iOS

1) Meqanic https://meqanic.com/app/

2.2.3 Non Educational Games ???

1) Game Jams or enthusiasts at itch???

- List Of Quantum Games
- https://itch.io/profile/decodoku
- https://quantumtictactoe.com/about/

3 Gamification and Game-Based Learning

In traditional teaching methods, textbooks and lectures are used and students are assessed through projects, assignments and examinations. Gamification is the integration of game elements, such as point systems, leaderboards, badges or other game-related elements, into conventional learning activities in order to increase student engagement and motivation. On the other hand, game-based learning is the design of learning activities so that the characteristics and principles of games are a key feature of these activities. Game elements can be categorized into three categories, which are presented below. (Seskir et al. 2022), (University of Waterloo 2024)

3.1 Game Elements

3.1.1 Point Systems

Points or Experience Systems reward students for completing tasks, just like conventional grades. They can introduce some useful features in learning environments, such as:

Limitless Points

While conventional grading systems collect learning artifacts that accumulate to one hundred percent of a course grade, point systems can accumulate points without a fixed end, pushing learners to do even better.

Flexible Goals

The courses can be structured to allow learners collect as many points as they want or force them complete a certain number of assignments to reach a desired threshold.

Student Choice

Learners can collect points by completing assignments or tests or any other course-relevant activities. They are free to choose how to collect points.

Tracking

Points can be tracked via grade books and shared with the entire class.

3.1.2 Badges

Using badges can be a great way to reward learners for their work and keep them engaged and motivated. Badges can also be shared with the class to encourage competition.

3.1.3 Leaderboards

The use of leaderboards is a great way to motivate learners, by providing them with constant feedback of where they rank in comparison to other group members. There are two types of leaderboards, absolute and relative.

Absolute

Absolute leaderboards rank all learners by some global measures, such as collected points. When this type of ranking is used, consideration must be given to the feeling of disappointment or discrimination that may be created against those in the bottom positions (Domínguez et al. 2013).

Relative

Relative leaderboards rank and group learners according to relative criteria. A common example is a view in which learners see only those students who are directly above or below them, solving the disappointment and discrimination issues.

3.2 Gamification Strategies

There are three types of gamification strategies: Online Strategies, Classroom Strategies and Out-of-class Strategies (University of Waterloo 2024).

3.2.1 Online Strategies

Discussion Boards

Learners can be prompted to participate in online discussions, before or immediately after class, by making them optional but awarding points for each post or reply to another post. They should be able

to use the collected points as an extra help on an assignment or for improving their final grade.

Quizzes

Instead of presenting a set of seemingly unrelated questions, the usage of interactive and narrative quizzes is suggested. This helps learners see the implications of their answers and also helps them stay engaged.

3.2.2 Classroom Strategies

Jeopardy

Jeopardy-style games can be used for chapter reviewing or midterm preparation and can provide enjoyment and enhance cooperation among learners in a familiar game structure.

Classroom Response Systems

Using classroom response systems can simplify the process of gathering responses from an entire class and can encourage participation by incorporating game elements such as point systems or leaderboards.

3.2.3 Out-of-class Gamification Strategies

Game-based learning environments

This strategy includes all games that are designed for educational purposes and do not require the presence of an instructor to be played.

Game-enhanced learning environments

These environments use commercially available games designed for entertainment purposes. Learners can play these games for fun and then discuss gaming experiences with other learners.

3.3 Pitfalls

There are some potential pitfalls that instructors may encounter when designing game-based learning activities (Moore-Russo, Wiss, and Grabowski 2018). First, game elements may not be connected to learning objectives. Quite often the assignments are introduced without planning how they support

the learning objectives. Also, sometimes the learning objectives are connected, but the context of the game is not. Thus, learners cannot retain knowledge or relate it to the subject. For game-based learning activities to be successful, learners must be able to retain and apply the knowledge they have been exposed to through the game. Furthermore, when students are focused on passing high stakes examinations, participating in a gamified environment can be seen as an unnecessary obstacle (Berkling and Thomas 2013). Learners who are used to learning and being assessed by conventional teaching methods and are focused on achieving high grades may feel deprived in a gamified environment. They will fill uncertainty, discomfort and may find it difficult to adapt to new rules. Another common pitfall is to capture the imagination of students and inspire them. In order to avoid this, opportunities for play, exploration and collaboration for new goals must be introduced (Nicholson 2012).

4 Mobile Game Categories, Genres and Subgenres

Gamerefinery has developed a flexible three-layer classification that allows mobile games to be easily grouped under distinctive genres (Julkunen 2024). Each game is classified into a subgenre, according to its features and mechanics. This subgenre belongs to one genre, which in turn belongs to one category. This three-layered approach acts as a helpful taxonomy for market and game research, as it provides game developers a singular unified approach to categorizing games.

Image 4.1. Game Categories, Genres and Subgenres according to GameRefinery

The image is sourced from (Julkunen 2024).

There are 4 categories, Casino games, Sports, Mid-core and Casual games. Each category has genres, which will be described in detail below. At the genre level, the differences between the games begin to become apparent. At the third level, the subgenre level, the mechanisms of the games become distinct.

4.1 Category #1: Casino

The casino category contains only one genre, the casino genre.

4.1.1 Genre #1.1: Casino

This genre contains traditional casino and gambling games. It has five subgenres.

Bingo

Games about playing bingo with others are included in this subgenre.

Cards

Includes casino card games, such as poker or blackjack.

Slots

This subgenre includes casino games with slot machines.

Casual Casino

Includes games that combine gambling elements with casual gameplay. For example, a game may have a casual casino-style gameplay (e.g. using slot machines) along with elements such as town building.

Other

Casino games that cannot be part of the categories above.

4.2 Category #2: Casual

This category contains six genres.

4.2.1 Genre #2.1: Hyper Casual

These games have very simple controls and are easy to learn. They are designed for short playing sessions and are very straightforward. They are divided into six subgenres.

Puzzle

Games where you have to solve some kind of puzzles.

Tap

These games require timing and precise and fast reactions. The gameplay focuses on tapping or holding one or more fingers on the device' s screen at the right time.

Steer

Includes games that require timing and reaction and the gameplay focuses on steering an object either by tilting the device or with some fingers.

Swipe / Drag

The gameplay focuses on swiping fingers or dragging and releasing objects.

IO

The main idea behind these games is for the player to grow by destroying other players or bots smaller than him and ultimately become the king of the whole gameplay area.

Other

Includes hyper casual games that don' t belong to any of the subgenres described above.

4.2.2 Genre #2.2: AR / Location Based

These games utilize augmented reality elements and location technology. This category does not have subgenres.

4.2.3 Genre #2.3: Arcade

These games have straightforward controls and mechanics for short and casual playing sessions.

Platformer

Casual platformer jumper games, where the player has to get through stages by jumping, running or gliding, while avoiding obstacles and/or enemies.

Shoot Them Up / Beat Them Up

Includes arcade style shooting and fighting games, with simple controls and lots of action, with no real emphasis on precise aiming or tactics.

Tower Defense

These games use tower defense mechanics. Main goal of the players is to prevent the enemies from reaching a certain point or target on the screen. Usually the enemies come in waves of increasing difficulty.

Board Games

Includes classic board game titles, either direct conversions of traditional board games to mobile versions, or games that utilize board games mechanics.

Other

Arcade games that do not fit in any other subgenre.

4.2.4 Genre #2.4: Lifestyle

These are games that revolve around lifestyle themes, such as decorating, fashion or customizing the look and style of models.

Customization

This subgenre includes games that focus on customizing or designing things, such as wardrobes or rooms.

Interactive Story

These games have very light mechanics and emphasize on interactive storytelling. Players' decisions affect the progress of the story.

Music / Band

Includes games where music and rhythm are affecting theme and mechanics. There are many customization options regarding style and look.

4.2.5 Genre #2.5: Simulation

These are casual games focusing on constructing and developing farms, cities, worlds or entities, while completing several tasks and side quests to progress in the game.

Adventures

In these games, players are focusing on completing tasks and collecting various items in order to process. The mechanics are often pretty lightweight, as they are limited to tapping or dragging objects. Emphasis is placed on the story and collecting aspects.

Breeding

Includes games that revolve around breeding creatures with each other, in order to get new, better creatures (e.g. breed two dragons to get a stronger dragon)

Tycoon / Crafting

The mechanics of these games revolve around construction and resource management.

Sandbox

The players of these games are free to roam the world of the game. They can craft things that will help them survive or grow. Emphasis is placed on user-generated content.

Time Management

Includes games where the players have to complete various tasks quickly, accurately and in the right order.

Idler

'Idler mechanics' means that the game plays itself even if the application is closed. When the application is open, players can see progress happening all the time - even if they are not doing anything themselves (e.g. crops growing, money or energy keeps generating etc.).

4.2.6 Genre #2.6: Puzzle

These games are focusing on puzzle solving or trivia and often use traditional board game mechanics.

Match-Three Puzzle

In these games players have to match pieces together to clear them from the board.

Bubble Shooter

They are Match-Three puzzles where you shoot board pieces -instead of swapping them- to make matches and clear the board.

Merge Games

Includes games where the players have to combine similar objects to create new objects of a higher tier. Merging is used to clear boards or upgrade items.

Action Puzzle

This subgenre includes games that require speed, aiming or directing an object in order to solve puzzles.

Word Games

These are games where the players have to solve word puzzles, like constructing or guessing words from given letters.

Trivia

Includes games that test players' general knowledge by asking questions. They often consist of levels of increasing difficulty.

Coloring Games

These games use tap-to-color or swipe-to-color mechanics, allowing players to experience a digital version of a coloring book.

Hidden Objects

Games that revolve around finding and tapping hidden objects in static scenes to progress in the game.

Solitaire

Includes solitaire games, as well as Mahjong Solitaire.

Other

Games focused on puzzle solving that do not belong to any of the subgenres described above.

4.3 Category #3: Mid-Core

Mid-Core category contains four genres.

4.3.1 Genre #3.1: Shooter

These games are focused on shooting targets from either a first or third person perspective. They often offer a Player-Versus-Player (PVP) gameplay.

Battle Royale

This subgenre includes both first-person and third-person shooter games with Battle Royale mechanics, such as shrinking areas, corpse scavenging and last team/player standing victory.

Classic FPS/TPS

Includes first-person shooter games that emphasize on PVP and team gameplay.

Sniper

These are shooters that use a single-player sniper theme, emphasizing a less direct approach to combat, often encouraging players to use stealth and keep their distance from the battlefield.

Tactical Shooter

Games where players control tanks, robots or other vessels from a third-person point of view. They emphasize on PVP and team gameplay.

4.3.2 Genre #3.2: Card Games

This genre has only one subgenre.

Card Battler

In these games, players have or create a deck of cards and battle other players. They emphasize on PVP and card collecting aspects.

4.3.3 Genre #3.3: Role Playing Games (RPG)

These are games where you control and develop some characters and defeat enemies to progress in the game.

Action

These games incorporate elements of adventure games. Players have direct control over characters' movement and use of skills.

Fighting

This subgenre includes games where players control an on-screen character and engages real-time one-on-one close combat. The fights take place in a closed arena setting.

Massive Multiplayer Online RPG (MMORPG)

These games are capable of supporting large numbers of players simultaneously, in the same open world.

Turn-Based

In these games you form and develop a team of characters to fight enemies in a turn-based setting.

Puzzle

This subgenre includes games that are a mix of Match-Three and RPG mechanics. These games mix character development with solving Match-Three puzzles.

Idle

These games play themselves, like casual idler games, while the app is closed. Once the player returns to the game, the characters have developed, collected loot and progressed through the game.

Sovereign

Includes games where player gets the role of a ruler and manages a country, an empire or a kingdom, fictional or real.

Survival

These are games where the players have to survive in the wilderness, often fighting monster or other players, while developing their character' s skills and equipment or building stronger communities around them.

Tower Defense

This subgenre includes games that combine casual tower defense mechanics with RPG elements. They have characters that can be improved and item progression mechanics. They often have a deep storyline.

4.3.4 Genre #3.4: Strategy

The games of this genre focus on resource management, building construction or army development. At the same time, players can make alliances or fight against other player and clans.

4X Strategy

This subgenre includes games that focus on ‘Exploring - Expanding - Exploiting and Exterminating’. Players have to focus on things like technology research, resource and troops management, open world exploration, base construction and fighting rivals.

Build and Battle

These are games where the players develop their bases, manage resources and create armies to battle both AI and human opponents. Battle mechanics emphasize on tactical thinking, which includes the proper deployment and direction of troops or spell casting.

Asymmetric Survival

Includes synchronous PVP games that use an asymmetric setup in team composition (e.g. 1 versus 4). Usually one side has the role of hunter and the other players/sides are hunted.

Tactical Battler

Includes games where players battle against non-playable characters (NPCs) or other players in a closed arena setup. These games also involve collecting and developing various aspects.

Multiplayer Battle Arena

These games focus on destroying the opposing team, together with your own team, in a closed arena setting.

4.4 Category #4: Sports and Driving

This category contains two distinct genres, sports and driving.

4.4.1 Genre #4.1: Sports

Games in this genre are all about sports or sport-themed action.

Arcade Sports

These games are based on a real-life sport, but with a very casual feel, for example with unrealistic physics or not using the exact rules of the sport in question.

Realistic Sports

This subgenre includes games that have realistic physics and graphics and an accurate rule-set of the sport in question.

4.4.2 Genre #4.2: Driving

Games in this genre are all about racing with cars, motorcycles or other vehicles or have a racing-theme action.

Arcade Driving

Includes racing games with an arcade setup with unrealistic physics.

Realistic Driving

These racing games have realistic physics.

5 Mobile Application Development

Mobile application development is the process of creating software applications that run on mobile devices. The software can be preinstalled on the device, downloaded from an app store or accessed through a web browser. (Amazon Web Services, Inc. 2024a), (IBM Corporation 2024)

Mobile games are digital games designed for mobile devices. They can utilize mobile sensors and hardware (e.g. accelerometers, GPS etc.), or even external peripherals, such as gaming controllers and AR/VR headsets.

5.1 Platforms

There are two dominant operating systems for mobile devices, Google' s Android and Apple' s iOS. iOS is used only on Apple devices while Android is used by several manufacturers.

Developing applications on each of these platforms requires the use of different software development kits (SDKs). There are four approached to mobile app development, which are compared below.

5.1.1 Native Applications

A native mobile application is a software application that is designed for a specific operating system platform (Rouse 2024). Native mobile apps can only work on the platform that they are designed for, because they use the programming languages, frameworks and interfaces that are platform-specific. They run directly on the operating system, so they tend to perform better than other applications that require interaction with the device' s operating system or hardware.

Because native apps are compiled directly into machine code, there have to be a different code base for each version of the same application (i.e. iOS or Android version). This is a requirement that significantly increases the cost and time of development and maintenance. (Amazon Web Services, Inc. 2024a), (Rouse 2024)

Native android apps are built with the Android SDK and use Java or Kotlin. On the other hand, native iOS apps are built with the iOS SDK and use Swift or Objective-C.

5.1.2 Cross-Platform Applications

Cross-platform apps have the ability to operate on different operating systems with little to no modification.Because the use universal coding languages and frameworks, cross-platform apps can run on iOS and Android using the same codebase. These coding languages and frameworks hide from developers the underlying differences between operating systems. (Marshall 2024)

Multiplatform apps reduce the cost for building and maintaining an application that targets different platforms. On the other hand, there are some performance issues and the access to device-specific features is limited, because they are not interacting directly with the operating system or the hardware. (Marshall 2024), (Amazon Web Services, Inc. 2024a)

According to Statista, the most popular frameworks for multiplatform development are Flutter and React-Native and Kotlin Multiplatform. (Vailshery 2024)

5.1.3 Hybrid-Web Applications

A hybrid-web application combines the elements of native and web apps. They are essentially web apps that have a native app shell. They are build with standard web technologies, like JavaScript and HTML and are bundled as native app packages. Hybrid apps are executed inside a container, which wraps the applications and acts as a bridge between the application and the operating system. (Techtarget contributor 2023), (Kohout 2016)

Although these apps reduce development and maintenance cost and user experience is very good, their performance is very low, as they cannot take advantage of many native device features. (Amazon Web Services, Inc. 2024a), (Rouse 2024)

5.1.4 Progressive Web Applications

Progressive Web Apps (PWAs) skip App Store delivery and conventional installation processes - they are accessible via a URL. PWAs are web apps that use browser capabilities to provide an app-like user experience, so they are written using web technologies, such as JavaScript and HTML.

PWAs are able to overcome certain disadvantages of Hybrid Web Apps, as they have better performance and more extensive access to device features. They also have low development and maintenance cost, but the app capabilities are restricted by the browser the use. (Amazon Web Services, Inc. 2024a), (Rouse 2024)

5.2 Languages and Frameworks

5.2.1 Android

Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. Java applications are compiled in bytecode that can run on any Java virtual machine, meaning that compiled Java code can run on all platforms without the need to recompile. It was initially released in 1995. Although Android is built on the Linux kernel, which is largely written in C, the Android SKD uses the Java language as the basis for its applications. (Wikipedia contributors 2024a)

Kotlin

Kotlin is a high-level, statically typed, general-purpose programming language with type interference. Kotlin is designed to interoperate fully with Java and the JVM version of Kotlin's standard library depends on the Java class library. Kotlin, released by JetBrains in 2016, aims to address Java's shortcomings and enhance development productivity. (Wikipedia contributors 2024b), (Fehervari 2024)

Comparison

Both Java and Kotlin compile to byte-code for the JVM, offering similar performance. Java applications tend to consume more memory, while Kotlin has more efficient memory management, with inline and extension functions that can reduce memory footprint. Kotlin has better startup time due to type interference, leading to faster initialization. Both languages support multithreading, but Kotlin's co-routines simplify concurrent code handling. Kotlin offers modern features and capabilities for Android development while Java has stronger community support. (GeeksforGeeks contributor 2024), (Fehervari 2024), (Medium contributor 2023)

5.2.2 iOS

Objective-C

Objective-C is a high-level, general-purpose, object-oriented programming language that first appeared in 1984. It is influenced by C and Smalltalk and it was primarily selected by NeXT for NeXTSTEP operating system. Apple chose Objective-C as the main programming language for iOS and macOS, because macOS was based on NeXTSTEP. (Wikipedia contributors 2024c)

Swift

Swift is a high-level, general-purpose, multi-paradigm programming language created in 2010 by Apple. Swift is intended to support the core concepts of Objective-C, but in a safer way. It compiles to byte-code and uses an LLVM compiler. (Wikipedia contributors 2024d)

Comparison

Objective-C has more complex and verbose syntax than Swift. It uses square brackets and has longer syntax for method and property definitions, while Swift is more concise and readable, with a syntax that resembles natural language. Swift is also significantly faster than Objective-C and offers a modern framework (SwiftUI) to build user interfaces. On the other hand, Objective-C has many well-documented, third-party frameworks and is well-tested and more stable. Also, it is a superset of C, so it works smoothly with C and C++ code. (Kaur 2023), (Popko 2024)

5.2.3 Cross-Platform Frameworks

Flutter

Flutter is a user interface (UI) SDK developed and released by Google in 2017. It can be used to create natively compiled mobile, web and desktop apps from a single codebase. It uses its own rendering engine to draw widgets on the screen, unlike other UI frameworks that rely on the platform' s rendering engine or manipulate the platform' s built-in UI stack. Flutter also provides access to native APIs. The Dart programming language is used to write applications in Flutter and the applications are compiled ahead-of-time (AOT) on all platforms except the web, where the code is transpiled to JavaScript or WebAssembly. (Wikipedia contributors 2024e), (JetBrains s.r.o. 2024), (Medium contributor 2024)

React-Native

React-Native is a UI SDK released by Meta Platforms (formerly Facebook Inc.) in 2015 and can be used to develop apps for mobile devices, Android TV, tvOS, web applications and desktop applications. Its components wrap existing native code and can interact with native APIs. React-Native apps are written in JavaScript or TypeScript. (Wikipedia contributors 2024h), (JetBrains s.r.o. 2024), (Medium contributor 2024)

Comparison

React-Native is easier to learn, because it uses JavaScript as a programming language and has greater community support. On the other hand, Flutter has better documentation and its command line interface (CLI) offers tools that allow Continuous Integration (CI) and Continuous Development (CD) to be created more easily than React-Native. Flutter' s CLI also offers the ability to automate application deployment in the app stores. Both frameworks feature hot-reload functionality, which allows developers to see changes instantly while modifying their code, without having to recompile.

Flutter uses its own on widgets and libraries and its own rendering engine and compiles directly to native code, while React-Native depends on the underlying platform and uses multiple JavaScript layers before compiling to native code. Also, React-Native requires the developers to use third-party libraries, both for development and testing. These facts make Flutter significantly faster than React-Native, while allowing Flutter to use less CPU and memory, have a smaller package size and have more consistent UI across platforms. (Shah 2024), (Bat 2024)

5.3 Platform, Framework and Game Engine Choice

The application that will be developed will be a cross-platform mobile application, in order to target users using both Android and iOS. We chose to develop a cross-platform application to get optimal performance, not rely on browser limitations and have a single codebase. This choice also reduces development and maintenance costs and provides a great user experience.

Between Flutter and React-Native, we chose to use Flutter, because it is lighter, faster, produces smaller packages and automates the application deployment in the stores. Also, Flutter has its own game engine, the Flame engine, which will make the development of our game easier.

In addition, Flutter has libraries (e.g. qartvm, quantools) for performing quantum calculations and simulating quantum circuits, which may be useful in the game development process, while there are no corresponding libraries in React-Native.

6 Quantum Computing

Quantum computers are computers that perform calculations by taking advantage of quantum phenomena, such as superposition and entanglement. The quantum properties of the microcosm provide the ability to store and process larger amounts of information and perform specific calculations at higher speeds than conventional computers. Information is processed using quantum gates and quantum algorithms, analogs of which cannot exist in conventional computers. (Karafillidis 2015), (Wikipedia contributors 2024f)

6.1 Quantum Bits

Quantum computers store information as bits. A quantum bit is a two-state system based on properties of the microcosm such as spin, energy state or the way particles oscillate and is the basic unit of information storage. (Wikipedia contributors 2024f)

State 0 is represented as $|0\rangle$ and state 1 as $|1\rangle$ and are called basic or basis states and are orthogonal to each other. Because the two states belong to a vector space, Hilbert space, they can be multiplied by a number and added together, and the result will be a valid state. Superposition is based on this fact. (Wikipedia contributors 2024f)

Each valid state can be written as:

$$|q\rangle = a|0\rangle + b|1\rangle$$

where a and b are called probability amplitudes and are complex numbers. The magnitudes of a and b are always less than one, and it holds that:

$$|a|^2 + |b|^2 = 1$$

The two basic states of a qubit can be represented as matrices:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The state $|q\rangle = a|0\rangle + b|1\rangle$ can be represented using matrices as follows:

$$|q\rangle = a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

The probability amplitudes, a and b, are complex numbers, so we can write the last equation in a more general form:

$$|q\rangle = e^{i\phi_a} \sin\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi_b} \cos\left(\frac{\theta}{2}\right)|1\rangle \Leftrightarrow$$

$$|q\rangle = e^{i\phi_a}\left(\sin\left(\frac{\theta}{2}\right)|0\rangle + e^{i(\phi_b - \phi_a)} \cos\left(\frac{\theta}{2}\right)|1\rangle\right)$$

If the global phase term $e^{i\phi_a}$ is omitted while the phase difference $\phi_b - \phi_a$ is called ϕ, the last equation simplifies to the following:

$$|q\rangle = \sin\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \cos\left(\frac{\theta}{2}\right)|1\rangle$$

The angle θ determines the magnitude of the probability amplitudes a and b, while the angle ϕ is called the phase angle and does not affect the measurement outcome. This means that two qubits differing only by a phase angle cannot be distinguished by a measurement. However, the phase angle should not be omitted, as it affects quantum computations.

The term $e^{i\phi_a}$ is called the global phase and is a mathematical artifact that can be safely ignored.

(Karafillidis 2015)

6.1.1 Superposition

Superposition is based on the addition of two states, similar to how we would add two waves, and gives quantum computers the ability to perform parallel computations.

A classical bit can have two distinct states, 0 or 1, and can be stored by any system that has two distinct states, closed or open. All information is analyzed, stored, and processed as a sequence of 0s and 1s by classical computers.

A quantum system can be in both states simultaneously. Before we measure its state, it tends to be in the state $|q\rangle$ or $|1\rangle$. The quantum state $|q\rangle = a|0\rangle + b|1\rangle$ is a superposition of the two basic states. Measuring the state of the system destroys the superposition, so a qubit can only be found in one of the two basis states.

The outcome of the measurement is impossible to predict with certainty, as all we know are the probabilities of it being in one of the two basis states, which are given by the square of the probability amplitudes, a and b.

(Amazon Web Services, Inc. 2024b), (Karafillidis 2015)

6.1.2 Quantum Entanglement

Quantum entanglement is defined as the state of two quantum systems when it cannot be written as a tensor product of their basic states. Quantum entanglement is a physical resource that can be used for the development of quantum algorithms and the execution of quantum computations. There is no classical analog of this state. When two systems are entangled, measuring the state of one reveals the state of the other, regardless of the distance between them. Essentially, the state of one quantum system depends on the state of the other. (Amazon Web Services, Inc. 2024b), (Karafillidis 2015)

6.1.3 Decoherence

Superposition states are unstable and decohere so that the system becomes stable. Decoherence is an irreversible process and can be caused by external factors such as an increase in temperature or radiation. (Amazon Web Services, Inc. 2024b)

6.2 Quantum Registers

Quantum registers are an array of qubits used as memory and represent a superposition of 2^n quantum states, where n is the number of qubits. The numbering of the qubits is done from right to left or from bottom to top. Quantum computers do not have classical circuits but perform operations by acting on bits that are within quantum registers.

The state of a quantum register with n qubits is defined as the tensor product of the states of the qubits that comprise it:

$$|q_r\rangle = |q_{n-1}\rangle \otimes ... \otimes |q_1\rangle \otimes |q_0\rangle$$

(Karafillidis 2015)

6.3 Quantum Gates

https://quantum.microsoft.com/en-us/insights/education/concepts/single-qubit-gates#:~:text=The%20Y%20gate%20perf

https://www.sharetechnote.com/html/QC/QuantumComputing_Gate_X.html

Quantum gates are not physical systems like classical gates, but physical processes applied to quantum bits (qubits) and registers that change their state. Additionally, information does not pass through quantum gates as it does with classical gates, since they are not part of any physical circuit with conductors. The information remains within the quantum registers, and the gates act on them by rotating their state vectors. A quantum gate can be a laser pulse or a magnetic field. (Karafillidis 2015), (Wikipedia contributors 2024f)

Since quantum bits are vectors in Hilbert space, quantum gates must be operators in Hilbert space. However, not all operators in Hilbert space are suitable for representing quantum gates. For an operator to be characterized as a quantum gate, it must not change the length of the state vector, only its angle, and it must not change the values of the inner products between state vectors. Thus, only unitary operators can constitute quantum gates. (Karafillidis 2015), (Wikipedia contributors 2024f)

The table below presents some of the most well-known quantum gates and the effects of their actions.

Πίνακας 6.1: Quantum Gates (Wikipedia contributors 2024g)

Name(s)	# of qubits	Symbol(s)	Matrix
Identity, No-Op	any	I	$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$
Pauli-X, NOT, Bit Flip	1	X, NOT, σ_x	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y	1	Y, σ_y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z, Phase Flip	1	Z, σ_z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Phase	1	S, P, $\sqrt{Z}$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

Name(s)	# of qubits	Symbol(s)	Matrix
Hadamard	1	H	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$
Controlled-NOT, Controlled-X, Controlled Bit Flip	2	CNOT, XOR, CX	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}$
Controlled-NOT, Controlled-X, Controlled Bit Flip	2	CNOT, XOR, CX	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}$
Swap	2	SWAP	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$

6.3.1 Identity

The identity quantum gate does not affect the state of the qubit. Its symbol is often omitted in quantum circuits.

Πίνακας 6.2: Identity Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$
$a\|0\rangle + b\|1\rangle$	$a\|0\rangle + b\|1\rangle$

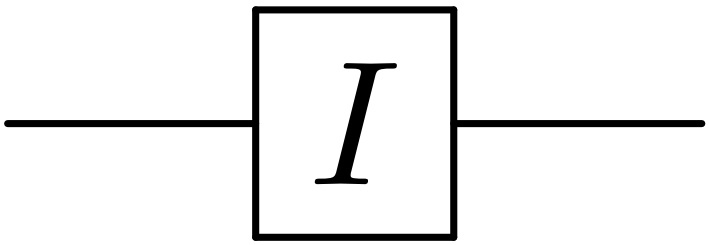


Image 6.1. Identity Gate Circuit Diagram

6.3.2 Pauli-X

The quantum X gate functions similarly to the classical NOT gate, rotating the state vector by 180 degrees around the X-axis. This operation changes the state from $|0\rangle$ to $|1\rangle$ and vice versa.

Πίνακας 6.3: Pauli-X Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$
$a\|0\rangle + b\|1\rangle$	$a\|1\rangle + b\|0\rangle$

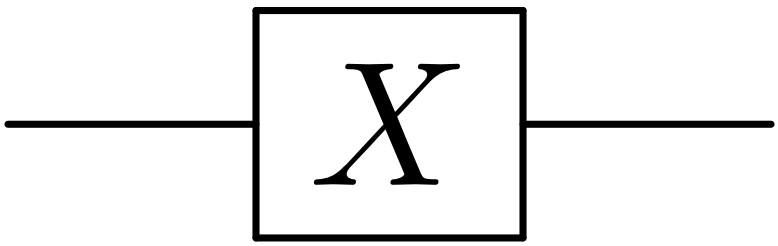


Image 6.2. Pauli-X Gate Circuit Diagram

6.3.3 Pauli-Y

The quantum Y gate operates similarly to the X gate but rotates the state vector by 180 degrees around the Y-axis. It changes the state from $|0\rangle$ to $|1\rangle$ and vice versa, and additionally shifts the phase of the

$|0\rangle$ state by 90 degrees and the phase of the $|1\rangle$ state by -90 degrees.

Πίνακας 6.4: Pauli-Y Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$i\|1\rangle$
$\|1\rangle$	$-i\|0\rangle$
$a\|0\rangle + b\|1\rangle$	$ia\|1\rangle - ib\|0\rangle$

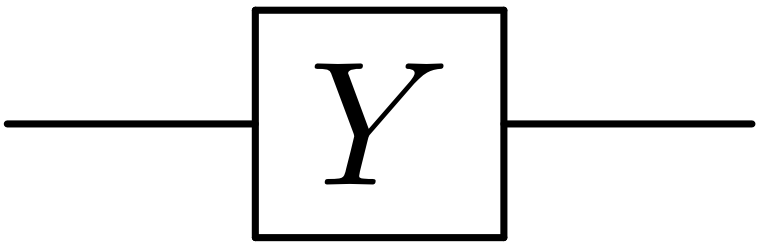


Image 6.3. Pauli-Y Gate Circuit Diagram

6.3.4 Pauli-Z

The quantum Z gate rotates the state vector by 180 degrees around the Z-axis. It shifts the phase of the $|1\rangle$ state by 180 degrees and does not affect the state of the $|0\rangle$ state.

Πίνακας 6.5: Pauli-Z Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$-\|1\rangle$
$a\|0\rangle + b\|1\rangle$	$a\|0\rangle - b\|1\rangle$

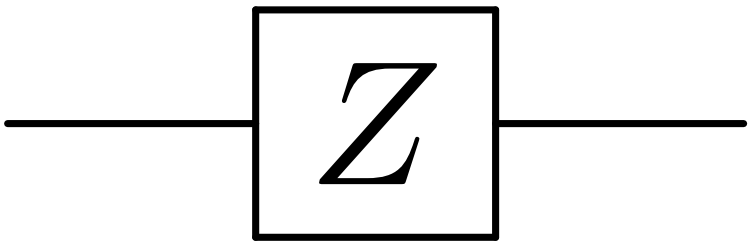


Image 6.4. Pauli-Z Gate Circuit Diagram

6.3.5 Phase

The S gate shifts the phase of the $|1\rangle$ state by 90 degrees.

Πίνακας 6.6: Phase Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$i\|1\rangle$
$a\|0\rangle + b\|1\rangle$	$a\|0\rangle + ib\|1\rangle$

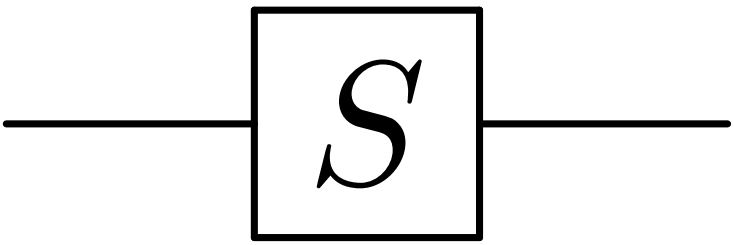


Image 6.5. Phase Gate Circuit Diagram

6.3.6 Hadamard

The Hadamard gate, when acting on a qubit in one of the two basis states, places it in a superposition of the two basis states. Conversely, when it acts on a qubit that is in a superposition of the two basis states, it returns it to one of the basis states.

Πίνακας 6.7: Hadamard Gate Truth Table

$\|q_i\rangle$	$\|q_o\rangle$
$\|0\rangle$	$\frac{1}{\sqrt{2}}\|0\rangle + \frac{1}{\sqrt{2}}\|1\rangle$
$\|1\rangle$	$\frac{1}{\sqrt{2}}\|0\rangle - \frac{1}{\sqrt{2}}\|1\rangle$
$\frac{1}{\sqrt{2}}\|0\rangle + \frac{1}{\sqrt{2}}\|1\rangle$	$\|0\rangle$
$\frac{1}{\sqrt{2}}\|0\rangle - \frac{1}{\sqrt{2}}\|1\rangle$	$\|1\rangle$
$a\|0\rangle + b\|1\rangle$	$\frac{1}{\sqrt{2}}(a+b)\|0\rangle + \frac{1}{\sqrt{2}}(a-b)\|1\rangle$

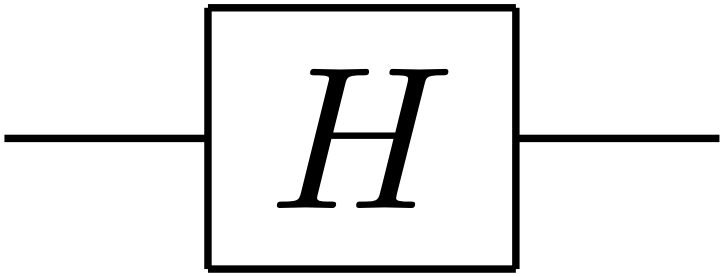


Image 6.6. Hadamard Gate Circuit Diagram

6.3.7 Controlled-NOT

The CNOT gate inverts the target bit when the control bit is set to 1. It functions as an X gate controlled by the control bit.

Πίνακας 6.8: Controlled-NOT Gate Truth Table - Second bit as control bit

$\|q_{control}q_{target}\rangle$	$\|q_{control}q_{target}\rangle$
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$
$a\|00\rangle + b\|01\rangle + c\|10\rangle + d\|11\rangle$	$a\|00\rangle + b\|01\rangle + c\|11\rangle + d\|10\rangle$

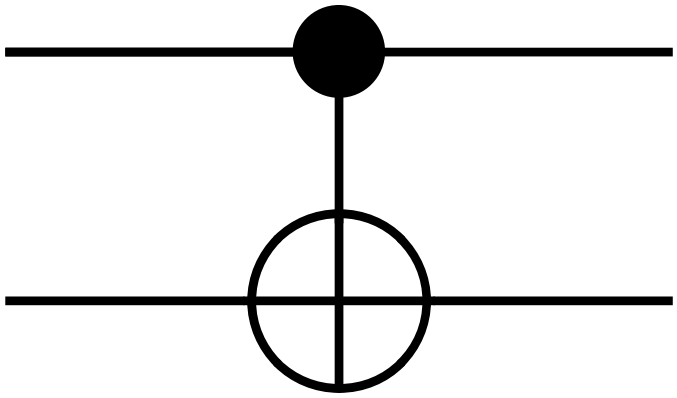


Image 6.7. CNOT Gate Circuit Diagram - Second bit as Control bit

Πίνακας 6.9: Controlled-NOT Gate Truth Table - First bit as control bit

$\|q_{control}q_{target}\rangle$	$\|q_{control}q_{target}\rangle$
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|11\rangle$
$\|10\rangle$	$\|10\rangle$
$\|11\rangle$	$\|00\rangle$
$a\|00\rangle + b\|01\rangle + c\|10\rangle + d\|11\rangle$	$a\|00\rangle + b\|11\rangle + c\|10\rangle + d\|11\rangle$

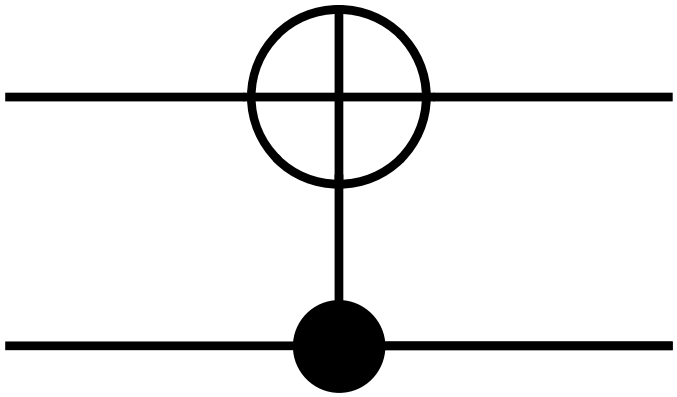


Image 6.8. CNOT Gate Circuit Diagram - First bit as Control bit

6.3.8 Swap

The SWAP gate exchanges the states of two qubits.

Πίνακας 6.10: SWAP Gate Truth Table

$\|q_{i1}q_{i0}\rangle$	$\|q_{01}q_{o0}\rangle$
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|10\rangle$
$\|10\rangle$	$\|01\rangle$
$\|11\rangle$	$\|11\rangle$
$a\|00\rangle + b\|01\rangle + c\|10\rangle + d\|11\rangle$	$a\|00\rangle + b\|10\rangle + c\|01\rangle + d\|11\rangle$

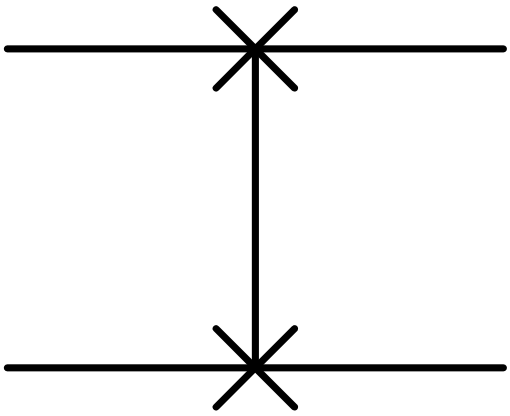


Image 6.9. SWAP Gate Circuit Diagram

References

The details about quantum gates and the tables are based on the following sources:

- (Karafillidis 2015)
- (Microsoft Corporation 2024a)
- (Microsoft Corporation 2024b)
- (Wikipedia contributors 2024g)

The images are sourced from (Wikipedia contributors 2024g).

7 Educational Game Development

In the previous chapters, we discussed gamification and presented the basic theory of quantum computing and some of the most well-known quantum gates. Next, we will present the educational game we designed to teach learners the concept of superposition and to familiarize them with the effects of some quantum games.

The game developed is called *Qubity* and is a variant of the arcade video game *Space Invaders*. It combines the characteristics of a "Shoot 'em Up" game with puzzle solving. The game is aimed at university students, as some levels require knowledge of complex numbers to solve the puzzle. However, the levels present the necessary theory for solving the puzzle, so a player with a flair for mathematics could complete the level without having any knowledge of complex numbers, as demonstrated by the evaluation of the game.

7.1 Game Description

The game consists of 30 levels, in which the player is asked to solve a puzzle using certain quantum gates -a subset of those presented in chapter 6. The player operates a spaceship whose position is determined by the state of a quantum register. The player can change the position of the spaceship by manipulating a qubit with one of the available quantum gates.

The first 17 levels have a single qubit register and the available gates are Pauli-X, Pauli-Y, Pauli-Z and Hadamard. The remaining 13 levels have a 2-qubit register and the available gates are Pauli-X, Pauli-Z and Hadamard. In each level there are 2, 4, or 8 different states in which the spaceship can be in, depending on the number of qubits and the available gates.

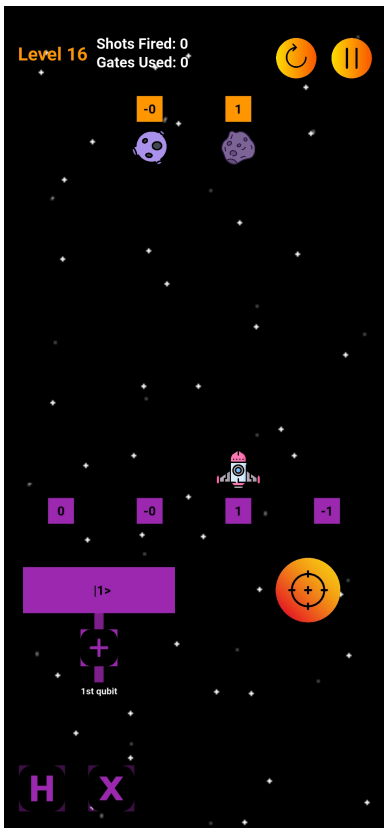


Image 7.1. Screenshot from Level 16

In each level, there are one or more asteroid targets. The player shall move the spaceship under the targets by altering the state of the register. Then, they have to destroy the asteroid by launching missiles. Several levels have more than one solution, but to score maximum points they must use the minimum number of gates and as few missiles as possible.

The game is accompanied by a 12-question quiz that allows players to test their knowledge gained by the game.

7.2 Screens

7.2.1 Home Screen

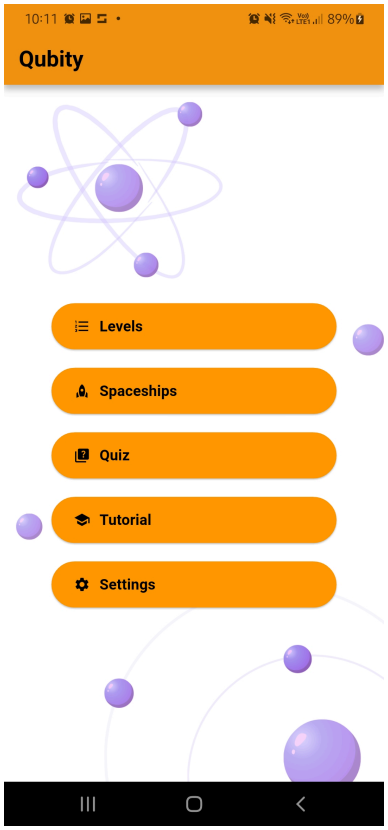


Image 7.2. Home Screen

From the app' s home screen, the user can access the game levels, available spaceships, the quiz, a tutorial level and game settings.

7.2.2 Levels

Image 7.3. Levels Screen

On this screen the player can select any of the available levels. Each level' s card shows the number of qubits, the available gates and the score.

7.2.3 Spaceships

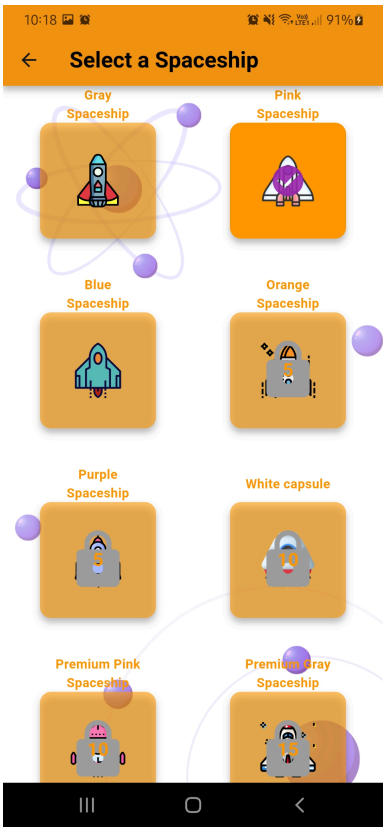


Image 7.4. Spaceships Screen

From this screen the player can select one of the available spaceships. The spaceships are decorative in nature, as they all have the same capabilities, but are a reward for the player' s progress in the game. Initially only 3 are available, new spaceships are unlocked after completing more levels.

7.2.4 Quiz

< image >

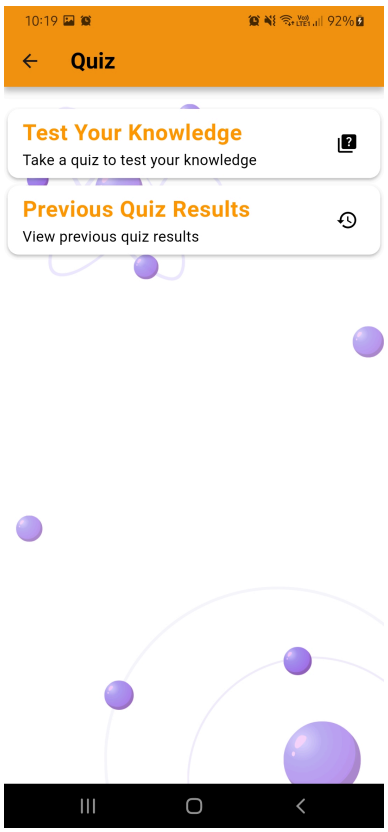


Image 7.5. Quiz Screen

This screen gives access to the knowledge check quiz. The player can answer a short quiz or view their score from previous attempts. Each quiz consists of 12 questions, 3 theoretical and 9 related to quantum gates.

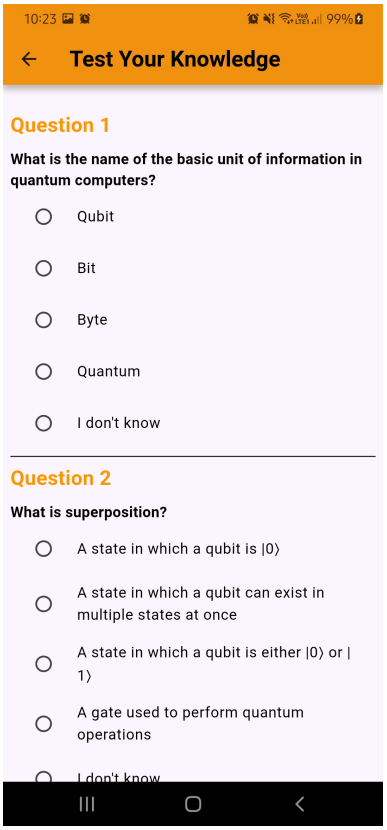


Image 7.6. Quiz Form

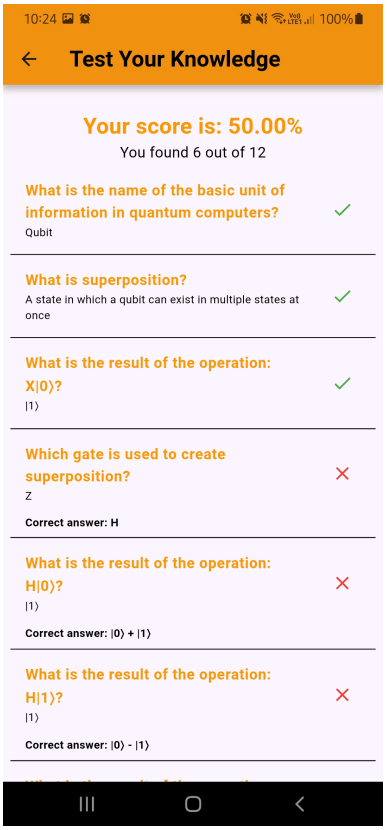


Image 7.7. Quiz Results

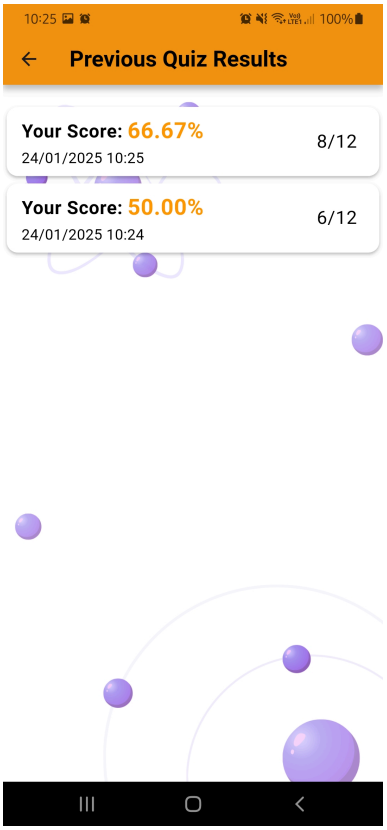


Image 7.8. Quiz History

7.2.5 Tutorial

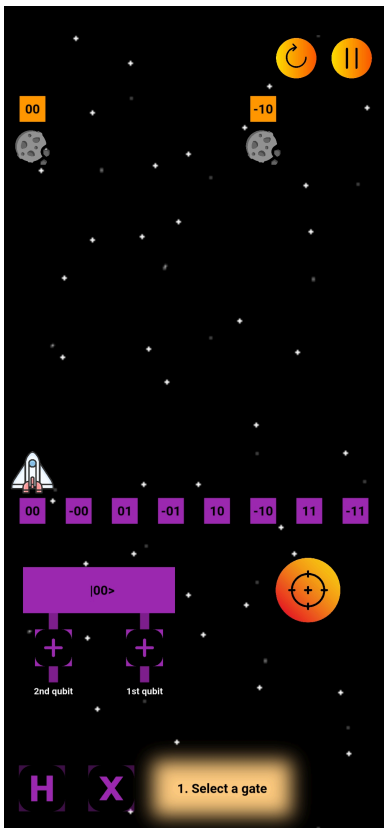


Image 7.9. Tutorial Screen

This screen loads a tutorial level, for which no score is calculated. First, a few slides are presented with basic quantum computing theory. In particular, the player is introduced to basic theory around qubits, quantum registers and quantum gates, as well as the concept of superposition. The slides consist of short sentences that summarize the theory discussed in chapter 6. Then there are 4 overlays with instructions, which demonstrate to the player how to use the gates to control the spaceship and how to destroy the asteroids.

7.2.6 Settings

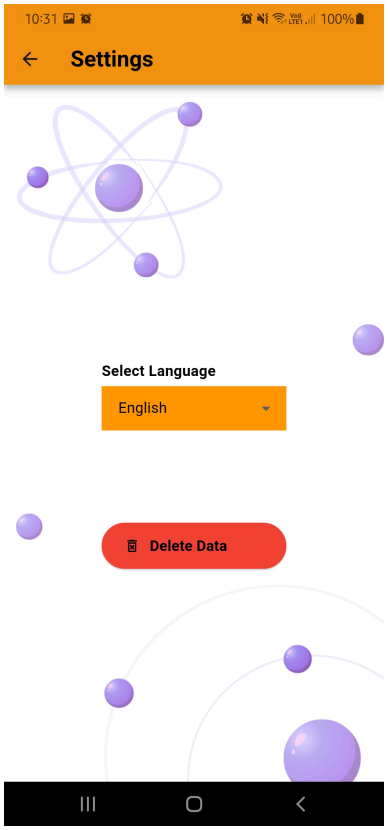


Image 7.10. Settings Screen

From this screen, the user can choose between the available languages and delete their progress in the game. Available languages are English and Greek, with the former being the default.

7.3 Levels

Each level has a different number of possible states, depending on the available gates and the number of qubits.

At the beginning of some levels, there are some slides with theory about the quantum gates available in this level. The first levels are very simple, requiring only 1 use of a gate to put the register into the target state. As the game progresses, the number of gates and combinations increases. In some levels there are gates that do not to be used. The player needs to use what they have learned to solve the puzzle using the minimum number of gates.

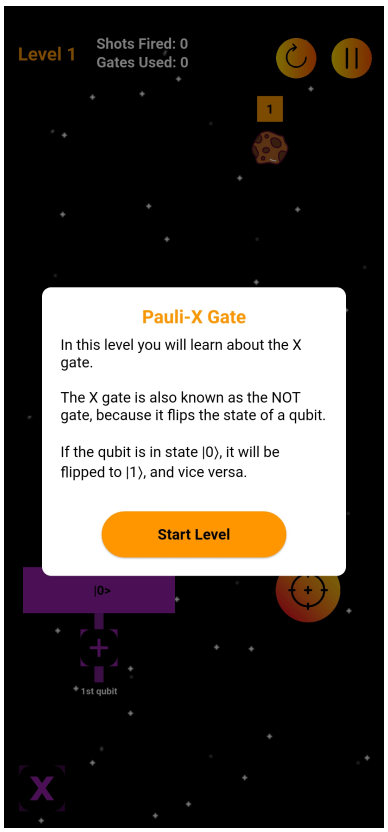


Image 7.11. Theory Slide

The first 9 levels introduce the Pauli Gates (X, Y and Z). In level 10, the theory of superposition is reintroduced to increase information retention. Also, Hadamard Gate (H) is introduced. In the next 7 levels, the player is asked to solve the puzzle by combining his previous knowledge of X and H gates.

From level 18, the number of qubits is increased to 2. From this level, no new theory is introduced; the player is just asked to combine their previous knowledge to solve the puzzles. In addition, the level of difficulty increases as the player has to decide which qubit to apply the gate to.

7.4 Development

Development of the game began in November 2024. Google' s *Material Design 3* and *Flame Game Engine* were user to create the graphical elements of the application. The spaceships, asteroids, gates and several other bitmaps used to create the sprite components of the game are from *Freepik*. The background image was created by *ChatGPT* and was slightly modified.

The first step was the selection of the quantum gates that would be presented in the game, and 30 scenarios were created to form the levels of the game. Each scenario consists of the initial state of the

quantum register (initial position of the spaceship), the target state (positions of the target asteroids), the available gates and the minimum number of gates to be used to solve the puzzle. The scenarios are encoded in a YAML file in a manner that is human-readable while allowing for easy addition of new levels.

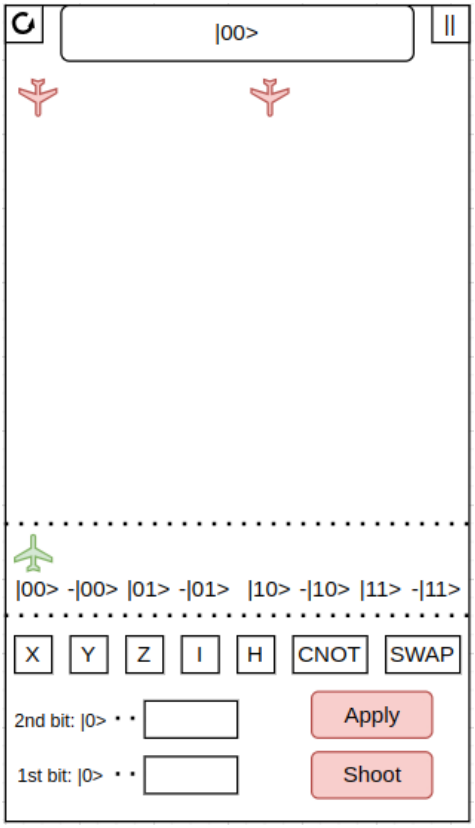


Image 7.12. Wireframe

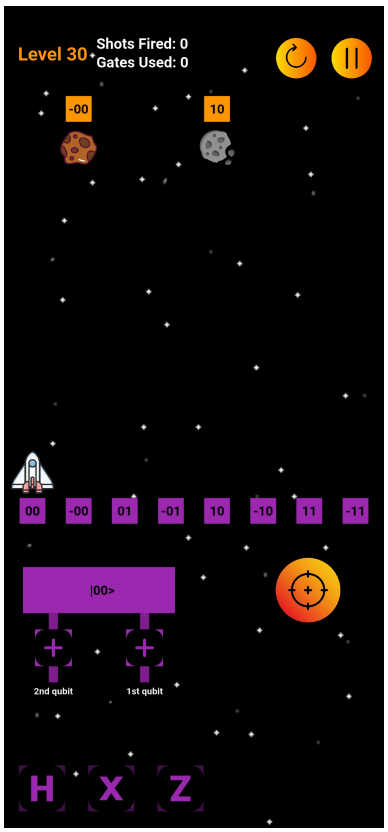


Image 7.13. Screenshot from Level 30

A prototype level was then designed in *diagrams.net*, which formed the foundation for creating the game' s graphical interface. As shown in the images above, the original approach did not include a graphical element for the quantum register, but rather 2 discrete qubits to which the quantum gate would be applied. There were also two buttons, one for applying the gate and one for launching the missiles. The target state was placed at the top of the screen. Finally, the targets were enemy spaceships, as in the original *Space Invaders* game.

By the first week of December, the basic elements of the user interface were in place and the spaceship control mechanism was working. Then, user testing was conducted, involving 5 users. Based on their feedback, the following corrections were made:

1. A graphical element was created to represent a quantum register with one or two outputs, replacing the discrete qubits. The outputs of the register represent the first or the second qubit.
2. The apply button has been removed. The gate action is applied to the register as soon as one of the register outputs is selected.
3. Targets were replaced with asteroids.

4. An overlay has been added to show the current level ID, gate used and missiles launched.

5. The target state display has been changed to make it easier of the player to understand where they need to place their spaceship.

In the last week of December, Greek language support was added, the tutorial level was created, and theory slides were added at the beginning of some levels. Also, a help menu was added on each level. Finally, quiz questions were created, as well as the quiz menu, quiz and quiz history screens.

In the following days, several bugs were fixed and UI issues were addresses after the application was tested by 2 regular users.

7.5 Score

In order to avoid the common pitfall that can lead to player frustration, no scoring system has been used to compare the player with other players. For each level, the player receives between half and three stars, with a perfect score of three. The optimal score is achieved by using exactly as many missiles as the targets of each level, and also by using as many gates as the optimal solution of each level requires.

To further motivate players as they progress through the game and complete levels, more spaceships become available as rewards. When a spaceship is unlocked, an informative toast is displayed.

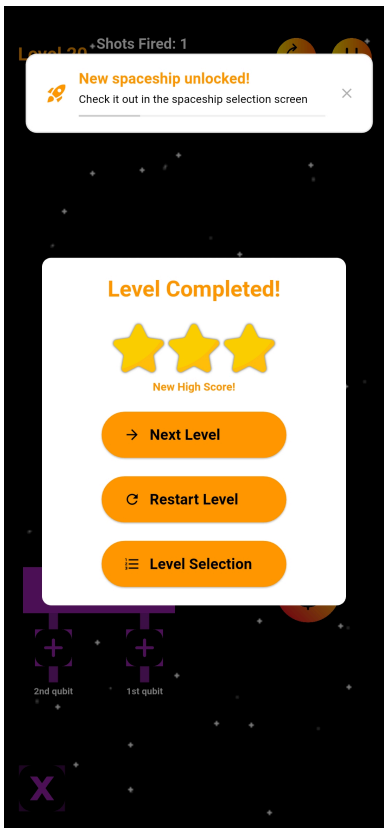


Image 7.14. Level score & spaceship reward

7.6 Quiz

To measure the educational value of the game, a 12-question quiz was designed. The quiz consists of 3 theoretical questions and 9 questions regarding quantum gates actions. The theoretical questions relate to the theory presented to the player via slides. The quantum gate questions ask the player to calculate the new state of a qubit after the gate has been applied to it, or ask the player to select the appropriate gate to perform a quantum calculation. The questions are designed to prove that the player has learned to use -some of- the gates introduced in chapter 6 and understands the concept of superposition.

7.7 Evaluation

10 university students volunteered to participate in the evaluation of the game, which was carried out in 2 phases. The volunteers claimed to have a strong background in mathematics, with 4 of them having knowledge of complex numbers.

In the first phase, the game was shared with the players, and they were asked to complete the quiz before playing the game. Then, they were asked to complete the tutorial level and then complete the 30 levels of the game and retake the quiz. Players were given as much time as they wanted and were asked to record how long it took to complete the game. There was no further guidance on how the game works and no external help was provided to solve the levels. The results of the first evaluation phase are presented below:

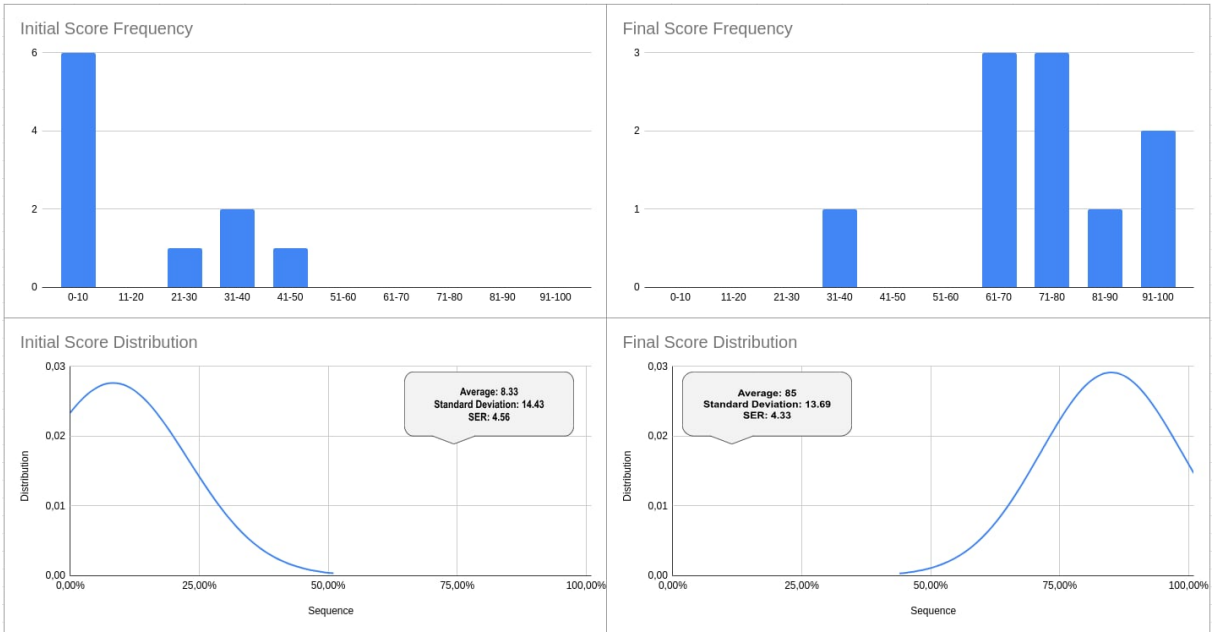


Image 7.15. Statistical Analysis

As shown in the graphs above, the average score on the quiz was 8.33% before playing the game and 85% after completing the game. It is clear that players improved their knowledge of quantum computing by playing the game.

Statistical analysis showed no correlation between the time required to complete the game and the improvement in the quiz score. The average time needed to complete the game was 21 minutes. This makes the game suitable to be played in the context of a college lecture to increase student engagement in the classroom or to allow students to test their knowledge in a fun and interactive way.

In the second phage, 8 out of 10 players from the fist phase participated in a focus group. The questions they were asked to discuss were related to 4 areas; *User Interface & User Experience* (UI & UX), *Game Mechanics*, *Difficulty* and *Scoring & Rewards*. They were also free to discuss with each other and comment on their overall experience of the application. Below are the focus group findings by area:

7.7.1 User Interface & User Experience

- One player requested an extra slide in the tutorial or in the first level, explaining how each level' s score is calculated.
- One player requested that the *Next Level* button should be a different color to others (on the level completion overlay).
- One player requested to group levels in the level selection menu by difficulty.
- Players agreed that the game menu was east to use and that the navigation worked well and was predictable.
- Players agreed that the state of the application was flawless and was updating correctly.
- Some players requested to add background music and sounds when launching missiles and hitting the targets.

7.7.2 Game Mechanics

- Players found the mechanics simple and casual and commented that this simplicity helped them focus on solving the puzzles.

7.7.3 Difficulty

- Players agreed that the game is easy and intended of users with no quantum computer experience. All of them could complete all 30 levels. They said they would like to try more difficult puzzles.
- Most players said that the hardest level was level 30 (the final one), as it required a combination of 3 gates to solve the puzzle.
- One player did not understand how the Hadamard gate acts on 2 qubits. Half of the players said they would like more information about superposition, but agreed that the theory presented was sufficient to solve the puzzles.
- One player suggested that there should be hints to use in case you find it difficult to solve the puzzle.

7.7.4 Scoring & Rewards

- Players agreed that having a score system encouraged them to improve. 7 out of 8 players tried to complete all levels with 3 stars.

- Players agreed that they felt rewarded and motivated to keep playing by unlocking new spaceships as they progressed.

7.8 Conclusions

Based on the feedback and data collected from the initial evaluation and focus group, several conclusions can be drawn:

1. The number of participants is small, only 10 people, so a complementary evaluation phase with a larger number of participants is suggested, for more reliable conclusions. However, the result of the initial evaluation is promising.
2. The duration of the game makes it suitable to be used as a teaching tool in the context of a lecture.
3. The UX should be improved, based on the feedback from the focus group.
4. The theory around superposition should be extended.
5. The game could be extended with more levels and quantum gates.

8 Bibliography

Amazon Web Services, Inc. 2024a. "Mobile Application Development." Amazon Web Services, Inc. https://aws.amazon.com/mobile/mobile-application-development/.

——. 2024b. "What Is Quantum Computing?" Amazon Web Services, Inc. https://aws.amazon.com/what-is/quantum-computing/.

Bat, Natesh. 2024. "Flutter Vs React Native : Performance Benchmarks You Can't Miss!" Medium. %22https://nateshmbhat.medium.com/flutter-vs-react-native-performance-benchmarks-you-cant-miss-%EF%B8%8F-2e31905df9b4%22.

Berkling, Kay, and Christoph Thomas. 2013. "Gamification of a Software Engineering Course and a Detailed Analysis of the Factors That Lead to It's Failure." In *2013 International Conference on Interactive Collaborative Learning (ICL)*, 525–30. https://scholar.google.com/scholar_lookup?hl=en&publication_year=2013&author=K.+Berkling&author=C.+Thomas&title=Gamification+of+a+Software+Engineering+Course+and+a+Detailed+Analysis+of+the+Factors+that+Lead+to+its+Failure#d=gs_cit&t=1722676216480&u=%2Fscholar%3Fq%3Dinfo%3AmDp34HDvUywJ%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D0%26hl%3Den.

Domínguez, A., J. Saenz-De-Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J. Martínez-Herráiz. 2013. "Gamifying Learning Experiences: Practical Implications and Outcomes." *Computer & Education*. Vol. 63. https://scholar.google.com/scholar_lookup?hl=en&volume=63&publication_year=2013&pages=380-92&journal=Computers+%26+Education&author=A.+Dom%C3%ADnguez&author=J.+Saenz-De-Navarrete&author=L.+De-Marcos&author=L.+Fern%C3%A1ndez-Sanz&author=C.+Pag%C3%A9s&author=J.+Mart%C3%ADnez-Herr%C3%A1iz&title=Gamifying+Learning+Experiences%3A+Practical+Implications+and+Outcomes.

Fehervari, Zoltan. 2024. "Another Guide on Performance: Java Vs Kotlin." Medium. https://medium.com/@fhrvri.mmxiv/another-guide-on-performance-java-vs-kotlin-40117fa93560.

GeeksforGeeks contributor. 2024. "Kotlin Vs Java - Which One Should i Choose for Android Development." https://www.geeksforgeeks.org/kotlin-vs-java/.

Harding, Eve. 2023. "The Pros and Cons of Game-Based Learning." Bedrock Learning. https://bedrocklearning.org/literacy-blogs/the-pros-and-cons-of-game-based-learning/.

IBM Corporation. 2024. “What Is Mobile Application Development?” IBM Corporation. https://www.ibm.com/topics/mobile-application-development.

JetBrains s.r.o. 2024. “The Six Most Popular Cross-Platform App Development Frameworks.” JetBrains s.r.o. https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html.

Julkunen, Joel. 2024. “Game Categories, Genres & Subgenres.” GameRefinery Oy. https://docs.gamerefinery.com/en/collections/112330-game-categories-genres-subgenres.

Karafillidis, Ioannis. 2015. “Quantum Computing.” Kallipos - Open Academic Editions. %22https://repository.kallipos.gr/handle/11419/216?&locale=en%22.

Kaur, Baljit. 2023. “Difference Between Objective c and Swift.” Medium. https://medium.com/swiftfy/difference-between-objective-c-and-swift-e53369ee2d4f.

Kohout, Jiri. 2016. “What Is a Mobile Application Containerization, or Wrapper, and Why Must It Die?” Teska Labs. https://teskalabs.com/blog/mobile-application-containerization-wrapping.

Ledda, Rosalie. 2012. “7 Tips for a Game-Based Learning Success.” eLearning Industry. https://elearningindustry.com/7-tips-game-based-learning.

Marshall, Gunnell. 2024. “Cross-Platform.” Techopedia. https://www.techopedia.com/definition/17056/cross-platform.

Medium contributor. 2023. “Kotlin Vs. Java: A Comparison of Features and Performance.” Medium. https://medium.com/@midoripig1009/kotlin-vs-java-a-comparison-of-features-and-performance-fe9eaac8b2c2.

——. 2024. “Top 10 Cross-Platform App Development Frameworks for 2024.” Medium. https://medium.com/@evincedevelop/top-10-cross-platform-app-development-frameworks-for-2024-1d812fdfc776.

Microsoft Corporation. 2024a. “Explore Quantum - Multi-Qubit Gates.” Microsoft Corporation. https://quantum.microsoft.com/en-us/insights/education/concepts/multi-qubit-gates.

——. 2024b. “Explore Quantum - Single-Qubit Gates.” Microsoft Corporation. https://quantum.microsoft.com/en-us/insights/education/concepts/single-qubit-gates.

Moore-Russo, Deborah, Andrew Wiss, and Jeremiah Grabowski. 2018. “Integration of Gamification into Course Design: A Noble Endeavor with Potential Pitfalls.” *College Teaching*. Vol. 66. http://www.tandfmisc.com/doi/abs/10.1080/87567555.2017.1295016.

Nagappan, Atic. 2023. “Pros and Cons of Quantum Computing.” https://www.linkedin.com/pulse/pros-cons-quantum-computing-athik-nagappan-1nhef/.

Nicholson, Scott. 2012. “A User-Centered Theoretical Framework for Meaningful Gamification, Paper Presented at the Games+ Learning+ Society 8.0.” *8.0, Madison, USA*. https://scholar.google.com/s

cholar_lookup?hl=en&volume=8&publication_year=2012&pages=223-30&journal=Games%2B+Learning%2B+Society&issue=1&author=S.+Nicholson&title=A+User-Centered+Theoretical+Framework+for+Meaningful+Gamification.

Piispanen, Laura. 2024. "List of Quantum Games." Aalto University. https://kiedos.art/quantum-games-list/.

Piispanen, Laura, Edward Morrell, Solip Park, Marcell Pfaffhauser, and Kultima Annakaisa. 2023. "The History of Quantum Games." Aalto University.

Piispanen, Laura, Marcel Pfaffjauser, James Wootton, Julian Togelius, and Annakaisa Kultima. 2024. "Defining Quantum Games." Aalto University. https://arxiv.org/abs/2206.00089.

Popko, Aleksander. 2024. "Objective-c Vs Swift: iOS Comparison." Netguru. https://www.netguru.com/blog/objective-c-vs-swift.

Rouse, Margaret. 2024. "Native Mobile App." Techopedia. https://www.techopedia.com/definition/27568/native-mobile-app.

Seskir, Z. C., P. Migdal, C. Weidner, A. Anuopam, N. Case, N. Davis, C. Decaroli, et al. 2022. "Quantum Games and Interactive Tools for Quantum Technologies Outreach and Education." *Optical Engineering*. Vol. 61. SPIE. https://doi.org/10.1117/1.OE.61.8.081809%7D.

Shah, Disha. 2024. "What Should You Choose from Flutter Vs. React Native in 2024?" Radixweb. https://radixweb.com/blog/flutter-vs-react-native.

Taylor, Eliza. 2024. "Advantages and Disadvantages of Quantum Computing." The Knowledge Academy. https://www.theknowledgeacademy.com/blog/advantages-and-disadvantages-of-quantum-computing/.

Techtarget contributor. 2023. "Hybrid Application." Techtarget. https://www.techtarget.com/searchsoftwarequality/definition/hybrid-application-hybrid-app.

University of Waterloo. 2024. "Gamification and Game-Based Learning." University of Waterloo. https://uwaterloo.ca/centre-for-teaching-excellence/catalogs/tip-sheets/gamification-and-game-based-learning.

Vailshery, Lionel Sujay. 2024. "Cross-Platform Mobile Frameworks Used by Software Developers Worldwide from 2019 to 2023." Statista. https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/.

Wikipedia contributors. 2024a. "Java (Programming Language)." Wikipedia. https://en.wikipedia.org/wiki/Java_(programming_language).

——. 2024b. "Kotlin (Programming Language)." Wikipedia. https://en.wikipedia.org/wiki/Kotlin_(programming_language).

——. 2024c. “Objective-c.” Wikipedia. https://en.wikipedia.org/wiki/Objective-C.

——. 2024d. “Swift (Programming Language).” Wikipedia. https://en.wikipedia.org/wiki/Swift_(programming_language).

——. 2024e. “Flutter (Software).” Wikipedia. https://en.wikipedia.org/wiki/Flutter_(software).

——. 2024f. “Quantum Computing.” Wikipedia. https://en.wikipedia.org/wiki/Quantum_computing.

——. 2024g. “Quantum Gate.” Wikipedia. https://en.wikipedia.org/wiki/List_of_quantum_logic_gates.

——. 2024h. “React Native.” Wikipedia. https://en.wikipedia.org/wiki/React_Native.

Wirtz, Bryan. 2023. “Why Game-Based Learning: Pros, Cons and How It Helps Students Retain Basic Knowledge.” Game Designing. https://www.gamedesigning.org/learn/game-based-learning/.

Wootton, James. 2017. “Why We Need to Make Quantum Games.” Decodoku. https://decodoku.medium.com/why-we-need-to-make-quantum-games-6f8c7bc4ace7.