



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

Ο Διευθυντής του Τομέα

Σιντόρης Χρήστος, Ε.ΔΙ.Π.





ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

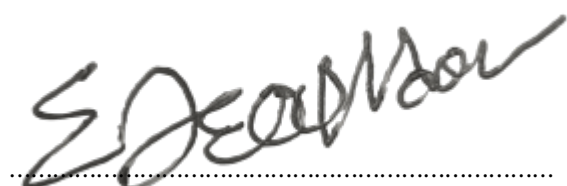
Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του τμήματος, του επιβλέποντα, ή της επιτροπής που την ενέκρινε.

#### 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

<b>1</b>	<b>1 Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	What is Game-Based Learning . . . . .	1
1.3	Benefits of Game Based Learning . . . . .	1
1.4	Research Objectives . . . . .	2
1.5	Thesis Structure . . . . .	3
<b>2</b>	<b>Literature Review</b>	<b>5</b>
2.1	Computer Science Games for Higher Education . . . . .	5
2.2	Quantum Physics Games . . . . .	5
2.2.1	Desktop and Web Games . . . . .	5
2.2.2	Mobile Games . . . . .	6
2.3	Why a mobile game? . . . . .	8
<b>3</b>	<b>Gamification and Game-Based Learning</b>	<b>9</b>
3.1	Game Elements . . . . .	9
3.1.1	Point Systems . . . . .	9
3.1.2	Badges . . . . .	10
3.1.3	Leaderboards . . . . .	10
3.2	Gamification Strategies . . . . .	10
3.2.1	Online Strategies . . . . .	11
3.2.2	Classroom Strategies . . . . .	11
3.2.3	Out-of-class Gamification Strategies . . . . .	11
3.3	Pitfalls . . . . .	12
<b>4</b>	<b>Mobile Game Categories, Genres and Subgenres</b>	<b>13</b>
4.1	Category #1: Casino . . . . .	15
4.1.1	Genre #1.1: Casino . . . . .	15
4.2	Category #2: Casual Games . . . . .	15
4.2.1	Genre #2.1: Hyper Casual . . . . .	15
4.2.2	Genre #2.2: AR / Location Based . . . . .	16

4.2.3	Genre #2.3: Arcade . . . . .	16
4.2.4	Genre #2.4: Lifestyle . . . . .	17
4.2.5	Genre #2.5: Simulation . . . . .	18
4.2.6	Genre #2.6: Puzzle . . . . .	19
4.3	Category #3: Mid-Core . . . . .	20
4.3.1	Genre #3.1: Shooter . . . . .	20
4.3.2	Genre #3.2: Card Games . . . . .	21
4.3.3	Genre #3.3: Role Playing Games (RPG) . . . . .	21
4.3.4	Genre #3.4: Strategy . . . . .	23
4.4	Category #4: Sports and Driving . . . . .	23
4.4.1	Genre #4.1: Sports . . . . .	24
4.4.2	Genre #4.2: Driving . . . . .	24
<b>5</b>	<b>Mobile Application Development</b>	<b>25</b>
5.1	Platforms . . . . .	25
5.1.1	Native Applications . . . . .	25
5.1.2	Cross-Platform Applications . . . . .	26
5.1.3	Hybrid-Web Applications . . . . .	26
5.1.4	Progressive Web Applications . . . . .	26
5.2	Languages and Frameworks . . . . .	27
5.2.1	Android . . . . .	27
5.2.2	iOS . . . . .	27
5.2.3	Cross-Platform Frameworks . . . . .	28
5.3	Game Engines . . . . .	28
5.4	Platform, Framework and Game Engine Choice . . . . .	28
<b>6</b>	<b>Bibliography</b>	<b>31</b>

## Κατάλογος πινάκων





# Κατάλογος σχημάτων

4.1	Game Categories, Genres and Subgenres according to GameRefinery . . . . .	14
-----	---	----



# 1 1 Introduction

## 1.1 Motivation

<https://www.theknowledgeacademy.com/blog/advantages-and-disadvantages-of-quantum-computing/>  
<https://www.linkedin.com/pulse/pros-cons-quantum-computing-athik-nagappan-1nhef/>

Quantum computing utilizes the principles of quantum mechanics to process information and solve complex problems exponentially faster than 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. 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

<https://decodoku.medium.com/why-we-need-to-make-quantum-games-6f8c7bc4ace7> <https://elearningindustry.com/7-tips-game-based-learning> <https://www.gamedesigning.org/learn/game-based-learning/> <https://bedrocklearning.org/literacy-blogs/the-pros-and-cons-of-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.

## 1.3 Benefits of Game Based Learning

<https://bedrocklearning.org/literacy-blogs/the-pros-and-cons-of-game-based-learning/> <https://www.gamedesigning.org/learn/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.

## 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](https://www.researchgate.net/publication/325046233_Games_for_Teaching_Computing_in_Higher_Education_-_A_Systematic_Review)

### 2.2 Quantum Physics Games

#### 2.2.1 Desktop and Web Games

:memo: **Name:** Name of the game :memo: **Created By:** Name of the creator (university or institution or student or individual) e.g. Student at University of Aalto :memo: **URL:** Where the game is hosted :memo: **Educational:** Was the game designed for educational purposes? True/False :memo: **Competition:** Was the game created for a competition? True/False :memo: **Course:** Was the game created in the scope of a course? True/False :memo: **Platform:** Desktop or Web

#### List Of Quantum Games

##### 1) Finnish Game Jam

- Competition for game developers.
- Desktop games: User must download the source files and build the game.
- Browser games.
- Most of the games are available on [itch.io](https://itch.io):
- Games created by Finnish Game Jam are not designed for educational purposes.
- The target of this competition is to create games with limited time and resources but with unlimited creativity [[link](#)].

##### 2) QPlayLearn

- Platform with web-based educational games.
- Uses interactive tools to make the learning process more effective and entertaining for different target groups [[link](#)].

### 3) [Quantum Games Course by Aalto University](#)

- 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.

### 4) [Science At Home](#)

- 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 [[link](#)]

### 5) [github / gitlab](#)

- Lots of desktop games. Most of them use Jupyter Notebooks.
- The majority of these games are not designed for educational purposes. They have been developed in the context of competitions or workshops.

### 6) [Decodoku](#)

- Has created a lot of quantum games. The games are hosted on [itch.io](#).
- The games does not have an educational purpose.
- Most of them are designed to run on the [IBM Quantum Platform](#)
- Browser and desktop games

Conclusions: - Educational or not? - Quantum Computing or Particle Physics? - Designed by professionals, universities or just for fun? - Desktop or web? - Programming language - Technologies?  
Consider adding table

## 2.2.2 Mobile Games

### 2.2.2.1 Google Play Store (for android devices)

- [Quantum](#)
  - Educational app
  - 13 chapters
  - 4 tests - 60 different questions
- [QuantumQ](#)
  - Puzzle game with theory



- Quantum gates and quantum circuits
- [Learn Quantum Physics](#)
  - Educational app
  - Designed for students and professionals
- [Quantum Computing](#)
  - Educational app
- [Quantum 3:](#)
  - Educational game designed by Michigan State University
  - Puzzle game
  - Particle physics

#### 2.2.2.2 App Store (for iOS devices)

- [Quantum Mechanics](#)
  - Educational app
  - Particle physics
- [Qika Quantum Game](#): Qika is a quantum game which you need to apply quantum gates to qubits in the grid. You need to change the states of qubits by applying gates in order to reach the target measurement.
  - Puzzle game
  - Applying quantum gates to qubits in order to reach a target measurement !!! Basic concept of our game !!!
- [Quantum 3:](#)
  - Same as android game
  - Educational game designed by Michigan State University
  - Puzzle game
  - Particle physics
- [Quantum Kate AR](#):
  - Educational game
  - Casual -> Simulation -> Adventure game
  - Particle physics

## 2.3 Why a mobile game?

---

Conclusions: - Educational or not? - Quantum Computing or Particle Physics? - Designed by professional, Universities or just for fun? - Android or iOS? - Programming language - Technologies?  
Consider adding table

## 2.3 Why a mobile game?

<https://www.ncfe.org.uk/all-articles/how-educational-games-are-changing-the-way-we-learn/>

[https://www.researchgate.net/figure/Core-educational-value-of-mobile-games\\_fig5\\_360277617](https://www.researchgate.net/figure/Core-educational-value-of-mobile-games_fig5_360277617)

<https://www.linkedin.com/pulse/top-7-reasons-using-mobile-apps-education-e-learning-industry-ved-raj>

## 3 Gamification and Game-Based Learning

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. In other words, it integrates game elements into existing learning activities.

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.

### 3.1 Game Elements

University of Waterloo ( “[Gamification and Game-Based Learning](#)” 2024) categorizes game elements into three main categories: Point Systems, Badges and Leaderboards.

#### 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 ( “Gamification and Game-Based Learning” 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 (Deborah Moore-Russo 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 ( “Game Categories, Genres & Subgenres” 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.



**Εικόνα 4.1.** Game Categories, Genres and Subgenres according to GameRefinery

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 Games

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. ( “[Mobile Application Development](#)” 2024), ( “[What Is Mobile Application Development?](#)” 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 approaches 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 ( “[Native Mobile App](#)” 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. ( “[Mobile Application Development](#)” 2024), ( “[Native Mobile App](#)” 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 they 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. ( [“Cross-Platform” 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. ( [“Cross-Platform” 2024](#) ), ( [“Mobile Application Development” 2024](#) )

Some of the most popular frameworks for multiplatform development are [Flutter](#), [React-Native](#), [Ionic](#) and [Kotlin Multiplatform](#) ( [“Top 10 Cross-Platform App Development Frameworks for 2024” 2024](#) ), ( [“The Six Most Popular Cross-Platform App Development Frameworks” 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 built 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. ( [“Hybrid Application” 2023](#) ), ( [“What Is a Mobile Application Containerization, or Wrapper, and Why Must It Die?” 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. ( [“Mobile Application Development” 2024](#) ), ( [“Native Mobile App” 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 they use. ( [“Mobile Application Development” 2024](#) ), ( [“Native Mobile App” 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 SDK uses the Java language as the basis for its applications. ( [“Java \(Programming Language\)” 2024](#))

#### 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. ( [“Kotlin \(Programming Language\)” 2024](#)), ( [“Another Guide on Performance: Java Vs Kotlin” 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. ( [“Kotlin Vs Java - Which One Should i Choose for Android Development” 2024](#)), ( [“Another Guide on Performance: Java Vs Kotlin” 2024](#)), ( [“Kotlin Vs. Java: A Comparison of Features and Performance” 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](#)

## 5.3 Game Engines

---

operating system. Apple chose Objective-C as the main programming language for iOS and macOS, because macOS was based on NeXTSTEP. ( [“Objective-c” 2024](#))

### 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](#). ( [“Swift \(Programming Language\)” 2024](#))

### 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. ( [“Difference Between Objective c and Swift” 2023](#)), ( [“Objective-c Vs Swift: iOS Comparison” 2024](#))

### 5.2.3 Cross-Platform Frameworks

(flutter vs react-native vs ionic vs kotlin multiplatform)

#### Flutter

#### React-Native

#### Ionic

#### Kotlin-Multiplatform

## 5.3 Game Engines

## 5.4 Platform, Framework and Game Engine Choice

-> iOS and Android -> why? -> target all users -> Cross Platform Approach -> why? -> Single code base, reduced costs of development and maintenance, very good user experience and sufficient performance -> Flame engine -> why?

[https://medium.com/\(evincedevelop/top-10-cross-platform-app-development-frameworks-for-2024-1d812fdcf776?\)](https://medium.com/(evincedevelop/top-10-cross-platform-app-development-frameworks-for-2024-1d812fdcf776?)) See ##Cross-Platform for reference - Medium

Why we selected flutter? - cross-platform: Υπήρχαν πιο πολλές εφαρμογές android. Εμείς θέλουμε όλοι οι χρήστες να μπορούν να παίξουν, ανεξάρτητα από τη συσκευή που έχουν. - Άλλα πλεονεκτήματα flutter π.χ. απόδοση/ταχύτητα σε σχέση με τις άλλες επιλογές??? - Επιπλέον της υποστήριξης από android και iOS, η εφαρμογή μπορεί εύκολα να διατεθεί και ως desktop ή web app. - Βιβλιοθήκες για κβαντικούς υπολογισμούς (συμβατές με το Dart version 3) \* <https://pub.dev/packages/qartvm> \* <https://pub.dev/packages/quantools> - Βιβλιοθήκες για πίνακες \* [https://api.flutter.dev/flutter/vector\\_math/vector\\_math-library.html](https://api.flutter.dev/flutter/vector_math/vector_math-library.html) \* [https://pub.dev/packages/advance\\_math](https://pub.dev/packages/advance_math) (supports complex numbers and matrices)





## 6 Bibliography

- “Another Guide on Performance: Java Vs Kotlin.” 2024. Medium. June 2024. <https://medium.com/@fhrvri.mmxiv/another-guide-on-performance-java-vs-kotlin-40117fa93560>.
- 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.” [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](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).
- “Cross-Platform.” 2024. Techopedia. May 2024. <https://www.techopedia.com/definition/17056/cross-platform>.
- Deborah Moore-Russo, Andrew Wiss, and Jeremiah Grabowski. 2018. “Integration of Gamification into Course Design: A Noble Endeavor with Potential Pitfalls.” *College Teaching*. Vol. 66. <http://www.tandfonline.com/doi/abs/10.1080/87567555.2017.1295016>.
- “Difference Between Objective c and Swift.” 2023. Medium. October 2023. <https://medium.com/swifty/difference-between-objective-c-and-swift-e53369ee2d4f>.
- 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](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).
- “Game Categories, Genres & Subgenres.” 2024. GameRefinery Oy. April 2024. <https://docs.gamerefinery.com/en/collections/112330-game-categories-genres-subgenres>.
- “Gamification and Game-Based Learning.” 2024. University of Waterloo. June 2024. <https://uwaterloo.ca/centre-for-teaching-excellence/catalogs/tip-sheets/gamification-and-game-based-learning>.
- “Hybrid Application.” 2023. January 2023. <https://www.techtarget.com/searchsoftwarequality/definition/hybrid-application-hybrid-app>.

- 
- “Java (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).
- “Kotlin (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)).
- “Kotlin Vs Java - Which One Should i Choose for Android Development.” 2024. July 2024. <https://www.geeksforgeeks.org/kotlin-vs-java/>.
- “Kotlin Vs. Java: A Comparison of Features and Performance.” 2023. Medium. August 2023. <https://medium.com/@midoripig1009/kotlin-vs-java-a-comparison-of-features-and-performance-fe9eaac8b2c2>.
- “Mobile Application Development.” 2024. Amazon Web Services, Inc. July 2024. <https://aws.amazon.com/mobile/mobile-application-development/>.
- “Native Mobile App.” 2024. Techopedia. July 2024. <https://www.techopedia.com/definition/27568/native-mobile-app>.
- 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/scholar\\_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](https://scholar.google.com/scholar_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).
- “Objective-c.” 2024. Wikipedia. July 2024. <https://en.wikipedia.org/wiki/Objective-C>.
- “Objective-c Vs Swift: iOS Comparison.” 2024. Netguru. April 2024. <https://www.netguru.com/blog/objective-c-vs-swift>.
- “Swift (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)).
- “The Six Most Popular Cross-Platform App Development Frameworks.” 2024. JetBrains s.r.o. April 2024. <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html>.
- “Top 10 Cross-Platform App Development Frameworks for 2024.” 2024. Medium. May 2024. <https://medium.com/@evincedevlop/top-10-cross-platform-app-development-frameworks-for-2024-1d812fdcf776>.
- “What Is a Mobile Application Containerization, or Wrapper, and Why Must It Die?” 2016. Teska Labs. September 2016. <https://teskalabs.com/blog/mobile-application-containerization-wrapping>.
- “What Is Mobile Application Development?” 2024. IBM Corporation. July 2024. <https://www.ibm.com/topics/mobile-application-development>.

- 
- “Another Guide on Performance: Java Vs Kotlin.” 2024. Medium. June 2024. <https://medium.com/@fhrvri.mmxiv/another-guide-on-performance-java-vs-kotlin-40117fa93560>.
- 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.” [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](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).
- “Cross-Platform.” 2024. Techopedia. May 2024. <https://www.techopedia.com/definition/17056/cross-platform>.
- Deborah Moore-Russo, Andrew Wiss, and Jeremiah Grabowski. 2018. “Integration of Gamification into Course Design: A Noble Endeavor with Potential Pitfalls.” *College Teaching*. Vol. 66. <http://www.tandfonline.com/doi/abs/10.1080/87567555.2017.1295016>.
- “Difference Between Objective c and Swift.” 2023. Medium. October 2023. <https://medium.com/swifty/difference-between-objective-c-and-swift-e53369ee2d4f>.
- 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](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).
- “Game Categories, Genres & Subgenres.” 2024. GameRefinery Oy. April 2024. <https://docs.gamerefinery.com/en/collections/112330-game-categories-genres-subgenres>.
- “Gamification and Game-Based Learning.” 2024. University of Waterloo. June 2024. <https://uwaterloo.ca/centre-for-teaching-excellence/catalogs/tip-sheets/gamification-and-game-based-learning>.
- “Hybrid Application.” 2023. January 2023. <https://www.techtarget.com/searchsoftwarequality/definition/hybrid-application-hybrid-app>.
- “Java (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).
- “Kotlin (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)).
- “Kotlin Vs Java - Which One Should i Choose for Android Development.” 2024. July 2024. <https://www.geeksforgeeks.org/kotlin-vs-java/>.

- 
- “Kotlin Vs. Java: A Comparison of Features and Performance.” 2023. Medium. August 2023. <https://medium.com/@midoripig1009/kotlin-vs-java-a-comparison-of-features-and-performance-fe9eaac8b2c2>.
- “Mobile Application Development.” 2024. Amazon Web Services, Inc. July 2024. <https://aws.amazon.com/mobile/mobile-application-development/>.
- “Native Mobile App.” 2024. Techopedia. July 2024. <https://www.techopedia.com/definition/27568/native-mobile-app>.
- 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/scholar\\_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](https://scholar.google.com/scholar_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).
- “Objective-c.” 2024. Wikipedia. July 2024. <https://en.wikipedia.org/wiki/Objective-C>.
- “Objective-c Vs Swift: iOS Comparison.” 2024. Netguru. April 2024. <https://www.netguru.com/blog/objective-c-vs-swift>.
- “Swift (Programming Language).” 2024. Wikipedia. July 2024. [https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)).
- “The Six Most Popular Cross-Platform App Development Frameworks.” 2024. JetBrains s.r.o. April 2024. <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html>.
- “Top 10 Cross-Platform App Development Frameworks for 2024.” 2024. Medium. May 2024. <https://medium.com/@evincedevelop/top-10-cross-platform-app-development-frameworks-for-2024-1d812fdcf776>.
- “What Is a Mobile Application Containerization, or Wrapper, and Why Must It Die?” 2016. Teska Labs. September 2016. <https://teskalabs.com/blog/mobile-application-containerization-wrapping>.
- “What Is Mobile Application Development?” 2024. IBM Corporation. July 2024. <https://www.ibm.com/topics/mobile-application-development>.