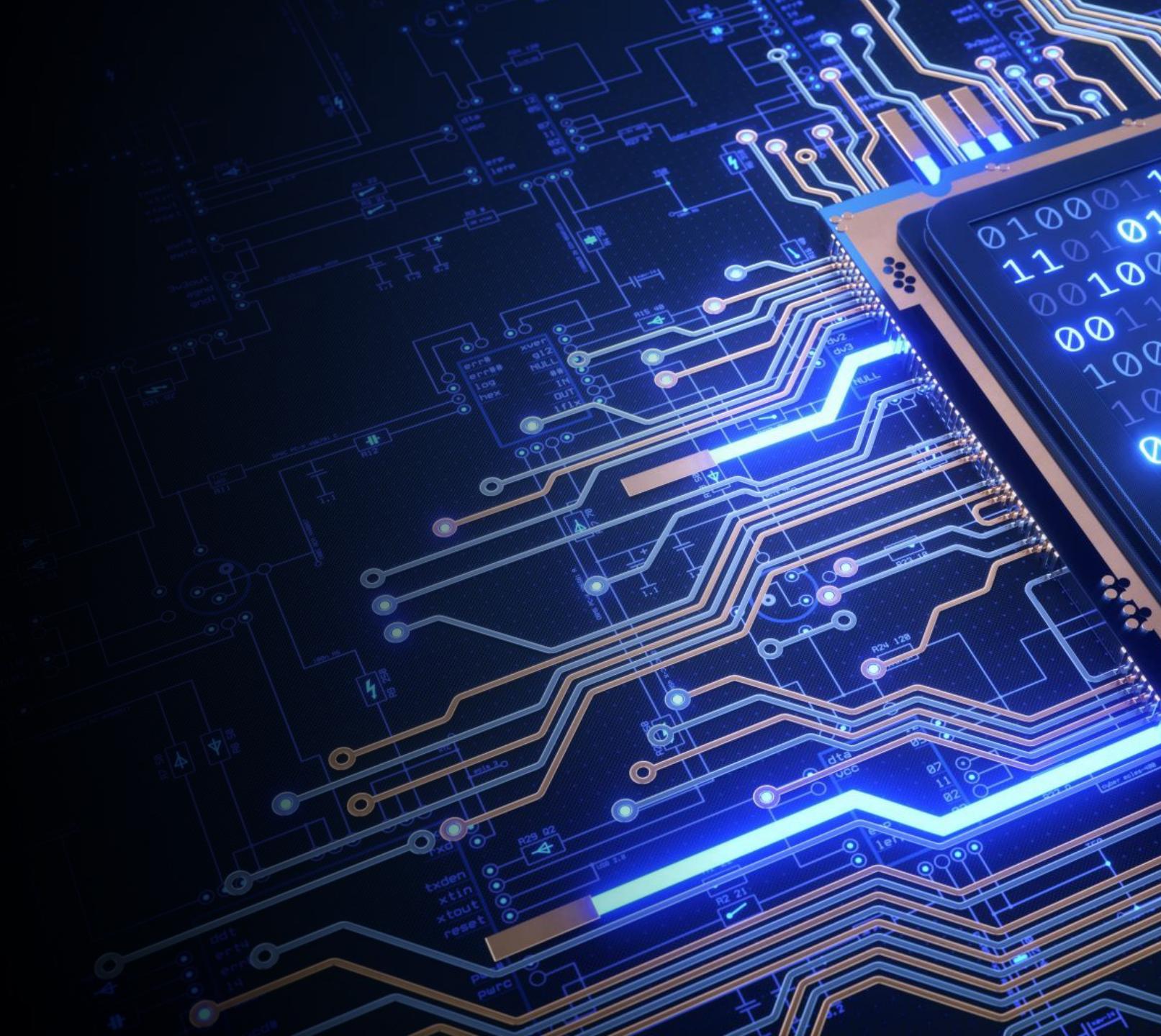




Implementing artificial intelligence mechanisms in mobile video gaming.

This is a presentation regarding the process behind developing a 3D mobile video game, which implements an artificial intelligence agent capable of dynamically altering the level's content generation based on the user's behavior and performance.





Main Points, general context, problem in question, suggested solution:

- The demand for innovative, replayable and easy to play mobile video games is higher than ever.
- The notion of developing mobile video games that provide an endless experience for the user, is widely accepted by the gaming community.
- However, providing the same endless experience to the user on each individual run (e.g. level design and difficulty adjustment), can quickly become simply flat and boring for some users.
- Therefore, the suggested software solution successfully studies the player's previous actions, behavior and gameplay habits, in order to modify the level design and difficulty settings accordingly.
- Thus, the player can experience a unique and personalized level design layout on each individual run.

Technologies Used:



Game Engine: Unity 3D (2019 version).



Code Editor: Visual Studio 2020, Sublime Text.



3D – 2D Design and Object visualization: Blender 3D, Adobe Photoshop, Adobe Illustrator.



Languages: C#, Java (Android), Swift (Apple)

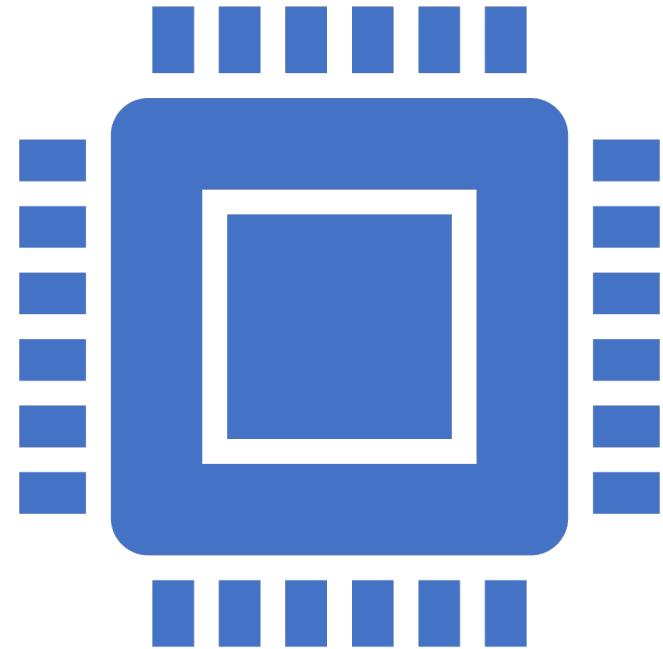


Third Party API's: High Mountains Vibration System API, Gley Plugin API for Apple's and Google's SDK integration.

Background research:

The background research process involved extensive research on the following topics:

- *Implementation of Artificial Intelligence mechanisms in mobile video games (allowed for the conceptualization and integration of the procedural content generation and the dynamic difficulty adjustment mechanisms).*
- *Mobile video game Optimization Techniques (for the optimization of the software).*
- *Available software and engines for mobile video gaming development (proven useful for the technologies that were used).*
- *Physics engine and Unity 3D gameplay traits (for the overall design and mechanisms of the video game).*



User types and characteristics:

- Characteristics of Registered User/Player:

- Age: 4+

- Gender: Female, Male, Other

- Knowledge background: Basic knowledge on how to download applications and games from the app store. Knowledge on using the Google Play, or the Game center online service. Additional but optional knowledge includes previous experience playing infinite runner games.

- Characteristics of Unregistered User/Player:

- Age: 4+

- Gender: Female, Male, Other

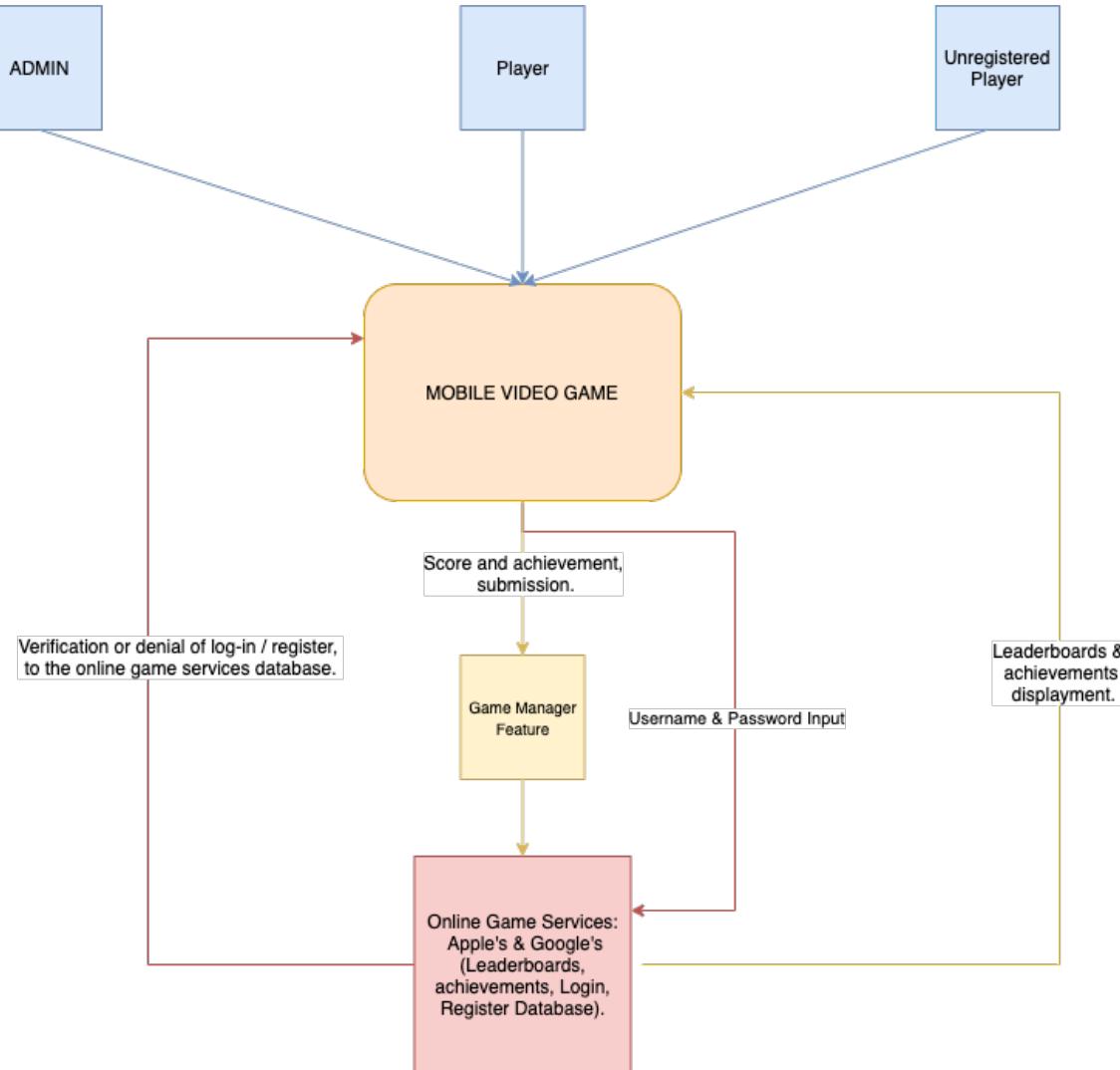
- Knowledge background: Basic knowledge on how to download applications and games from the app store. Moreover, some basic optional knowledge includes registering and creating a profile on the google play or apple's game center services. Additional but also optional knowledge includes previous experience playing infinite runner games.

- Characteristics of Administrator

- Age: 18 +

- Gender: Female, Male, Other

- Knowledge: Coding skills, Internet Programming skills, general knowledge on database data manipulation, an apple developer account and a google developer account, in order to be able to publish the game, including future updates. Finally, other skills and general knowledge are required for using Apple's and Google's developer pages, where he/she can alter settings and preferences regarding the state of the videogame on the store.

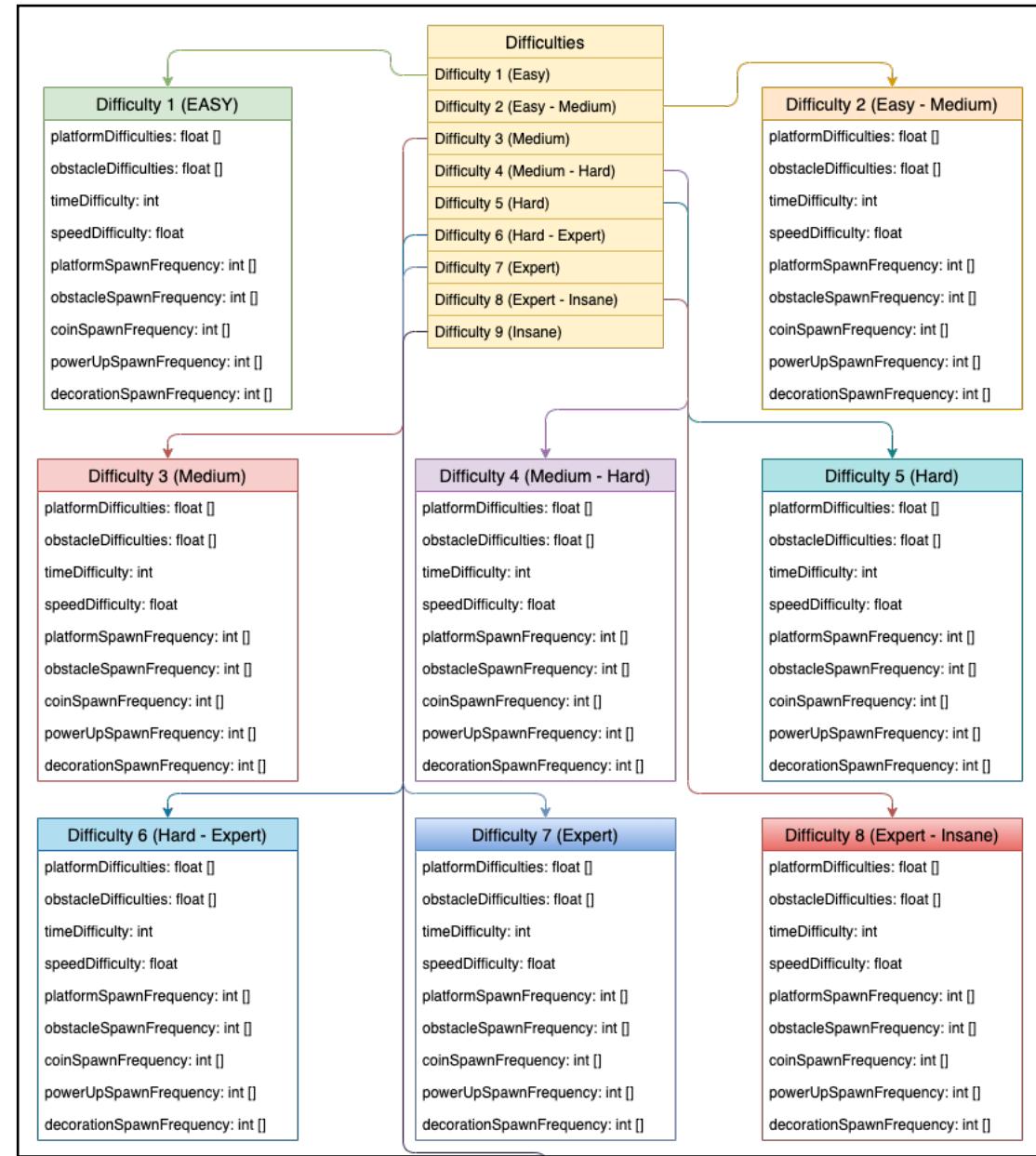


ACD and interactivity:

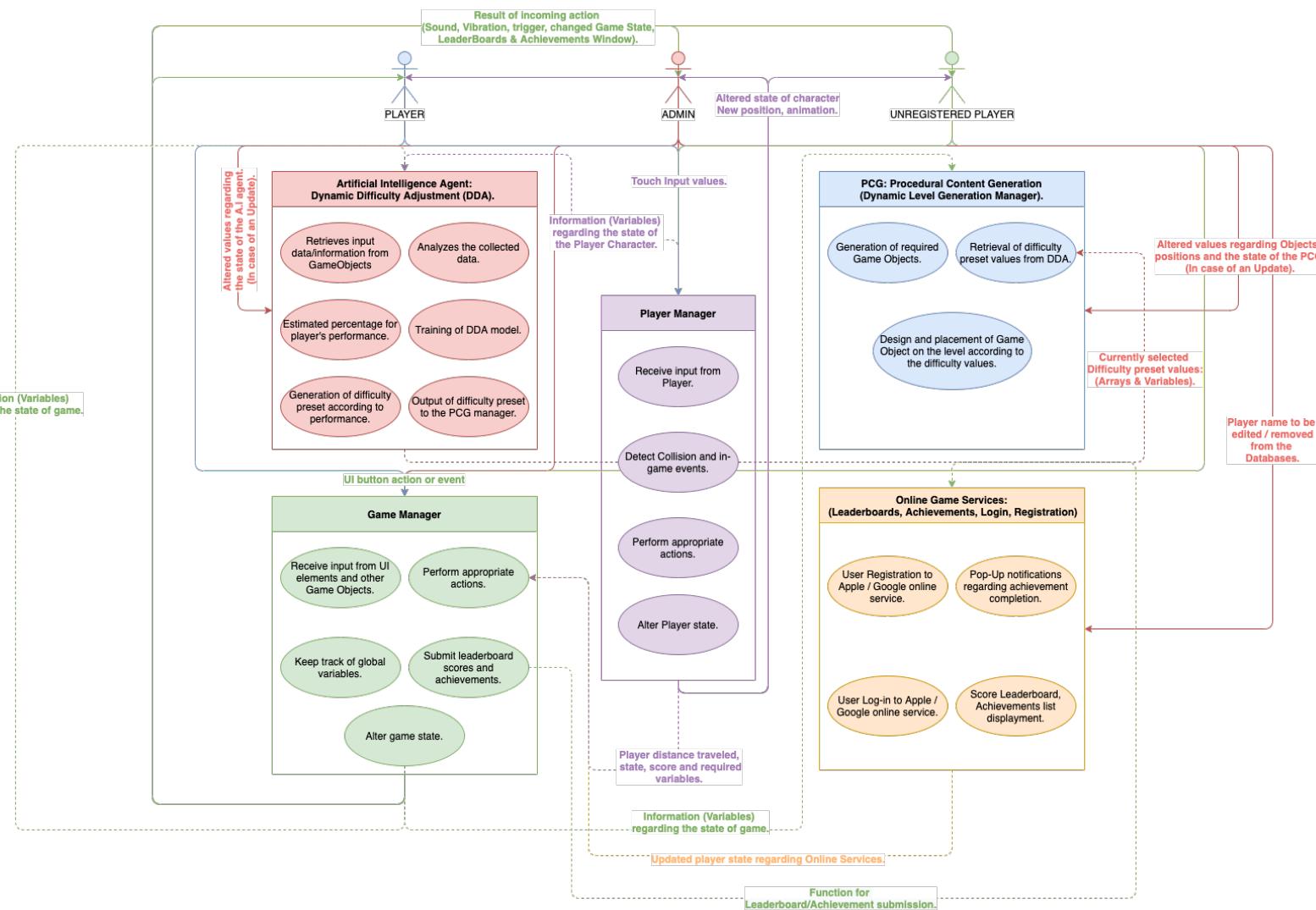
Features:

- 1) Online Game Services (Leaderboards, Achievements, Sign-in, Register)
- 2) PCG: Procedural Content Generator (Dynamic Level Generation)
- 3) Game Manager
- 4) Player Manager
- 5) Artificial intelligence Agent / Dynamic Difficulty Adjustment (DDA)

Interactivity & ERD:

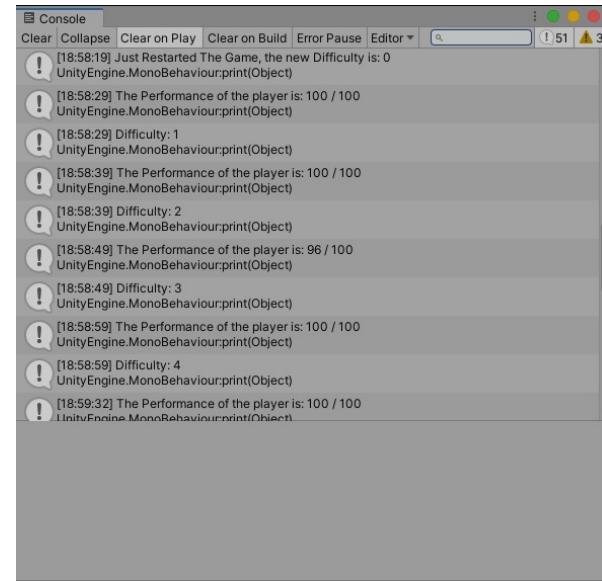


System structure:



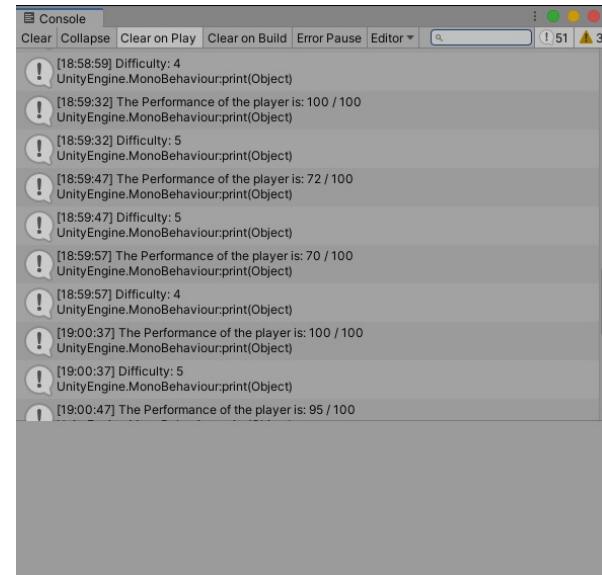
Testing strategy and proof:

- The testing strategy that was followed for every single component and feature during the development period of the video game, was a trial and error method. Each time that a single new feature or component was implemented, the appropriate testing procedure was then applied, in order to test the functionality and the validity of the specific feature.



Console

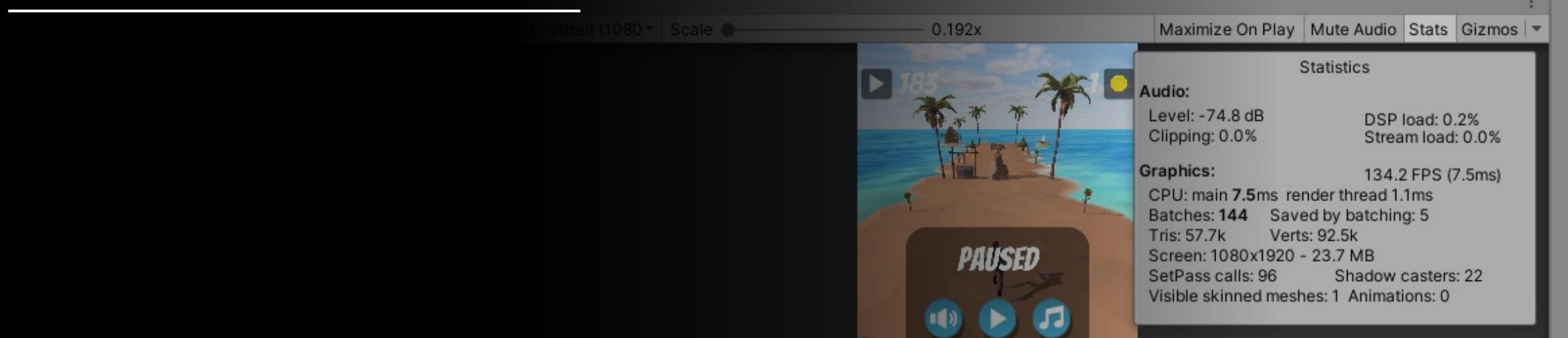
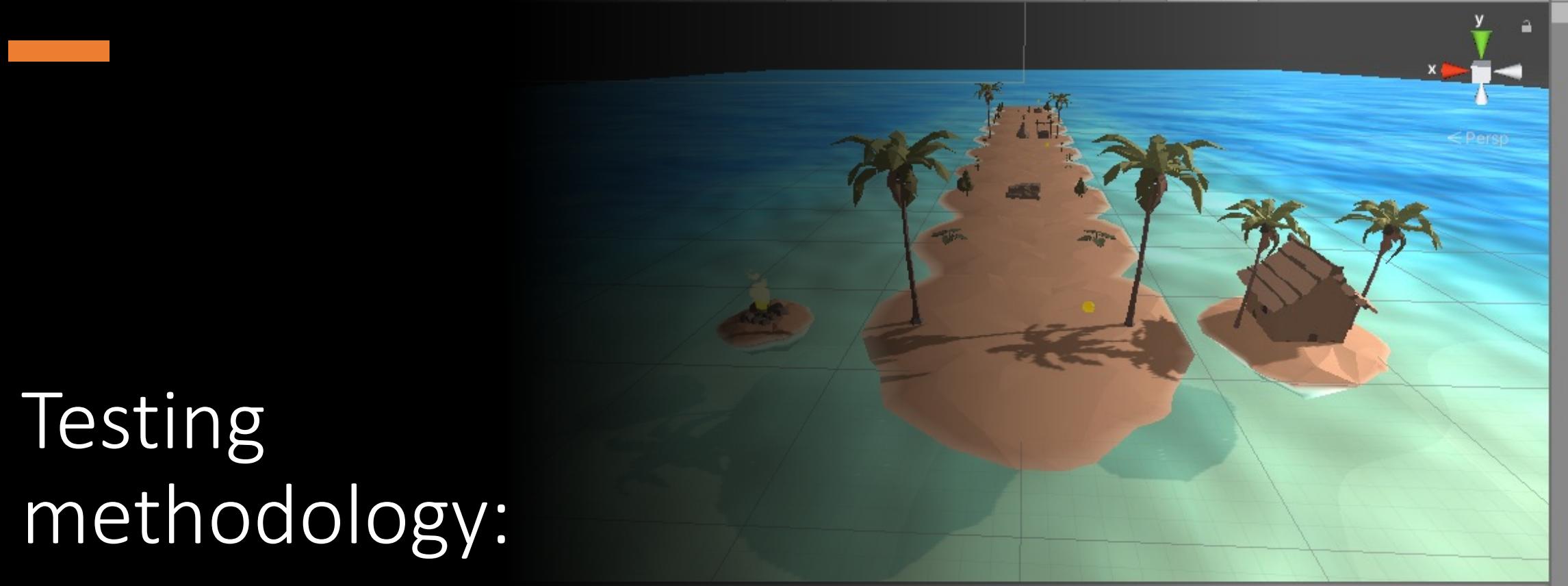
[18:58:19] Just Restarted The Game, the new Difficulty is: 0
UnityEngine.MonoBehaviour:print(Object)
[18:58:29] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:58:29] Difficulty: 1
UnityEngine.MonoBehaviour:print(Object)
[18:58:39] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:58:39] Difficulty: 2
UnityEngine.MonoBehaviour:print(Object)
[18:58:49] The Performance of the player is: 96 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:58:49] Difficulty: 3
UnityEngine.MonoBehaviour:print(Object)
[18:58:59] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:58:59] Difficulty: 4
UnityEngine.MonoBehaviour:print(Object)
[18:59:32] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)



Console

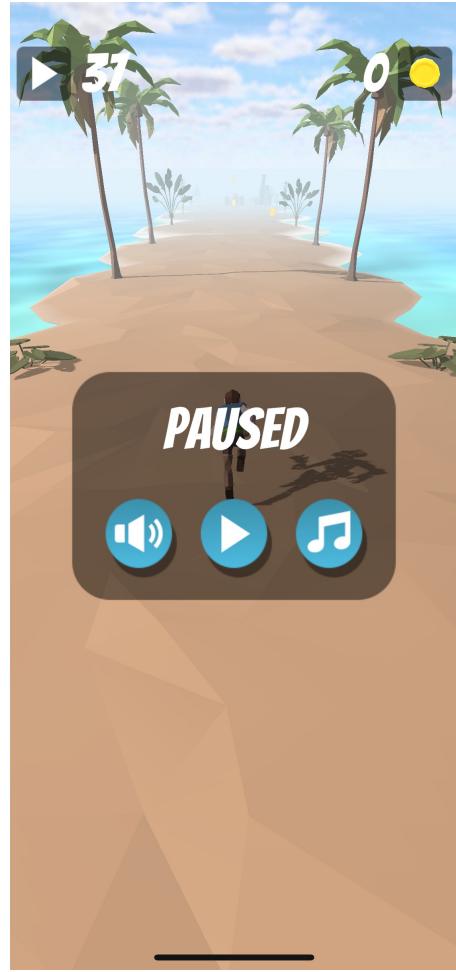
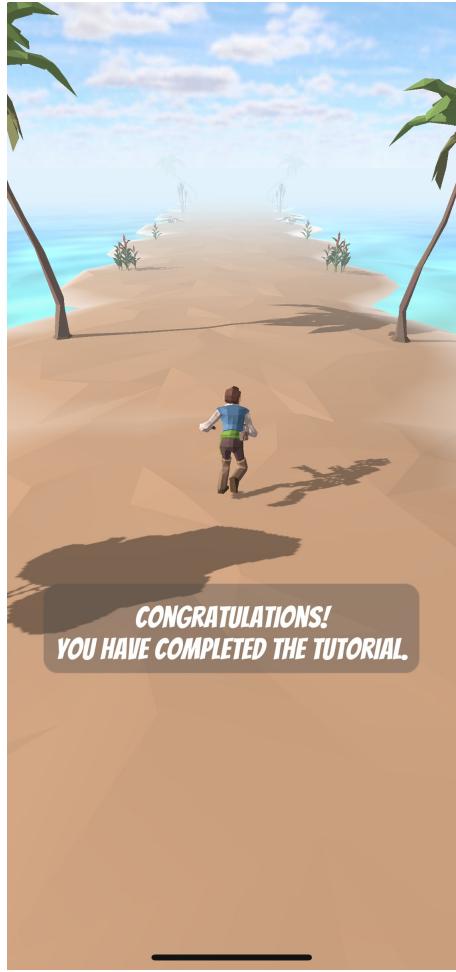
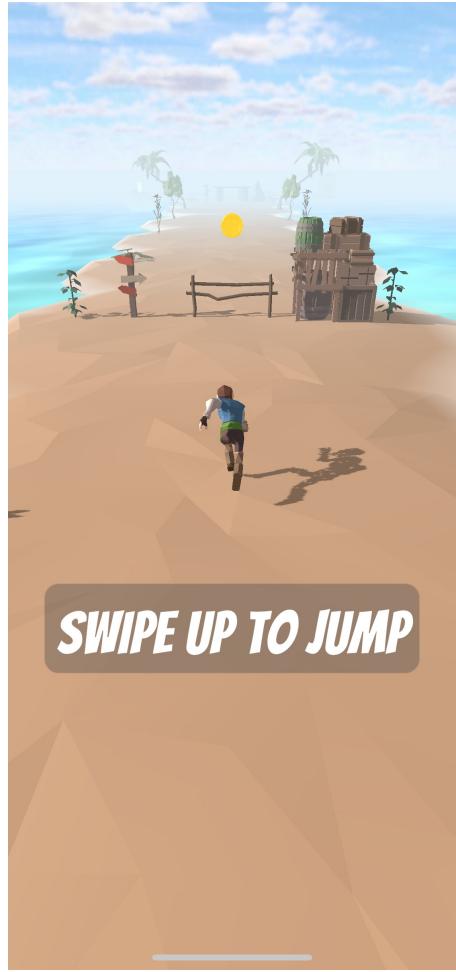
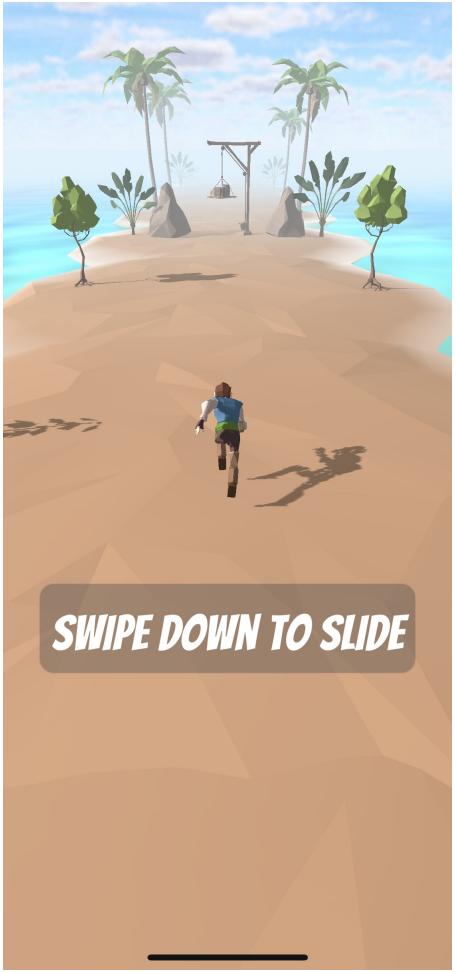
[18:58:59] Difficulty: 4
UnityEngine.MonoBehaviour:print(Object)
[18:59:32] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:59:32] Difficulty: 5
UnityEngine.MonoBehaviour:print(Object)
[18:59:47] The Performance of the player is: 72 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:59:47] Difficulty: 5
UnityEngine.MonoBehaviour:print(Object)
[18:59:57] The Performance of the player is: 70 / 100
UnityEngine.MonoBehaviour:print(Object)
[18:59:57] Difficulty: 4
UnityEngine.MonoBehaviour:print(Object)
[19:00:37] The Performance of the player is: 100 / 100
UnityEngine.MonoBehaviour:print(Object)
[19:00:37] Difficulty: 5
UnityEngine.MonoBehaviour:print(Object)
[19:00:47] The Performance of the player is: 95 / 100
UnityEngine.MonoBehaviour:print(Object)

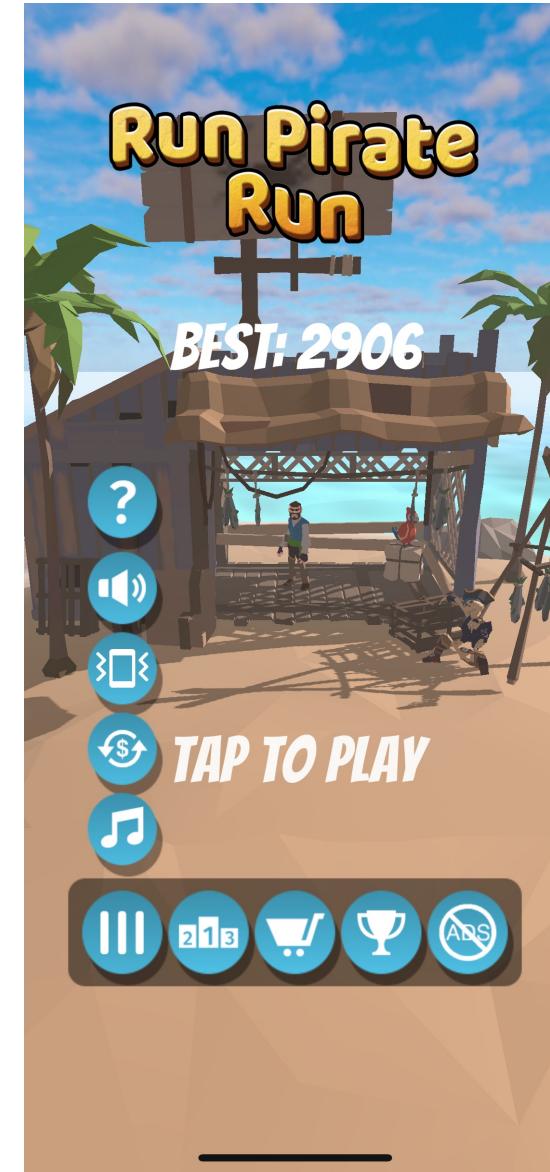
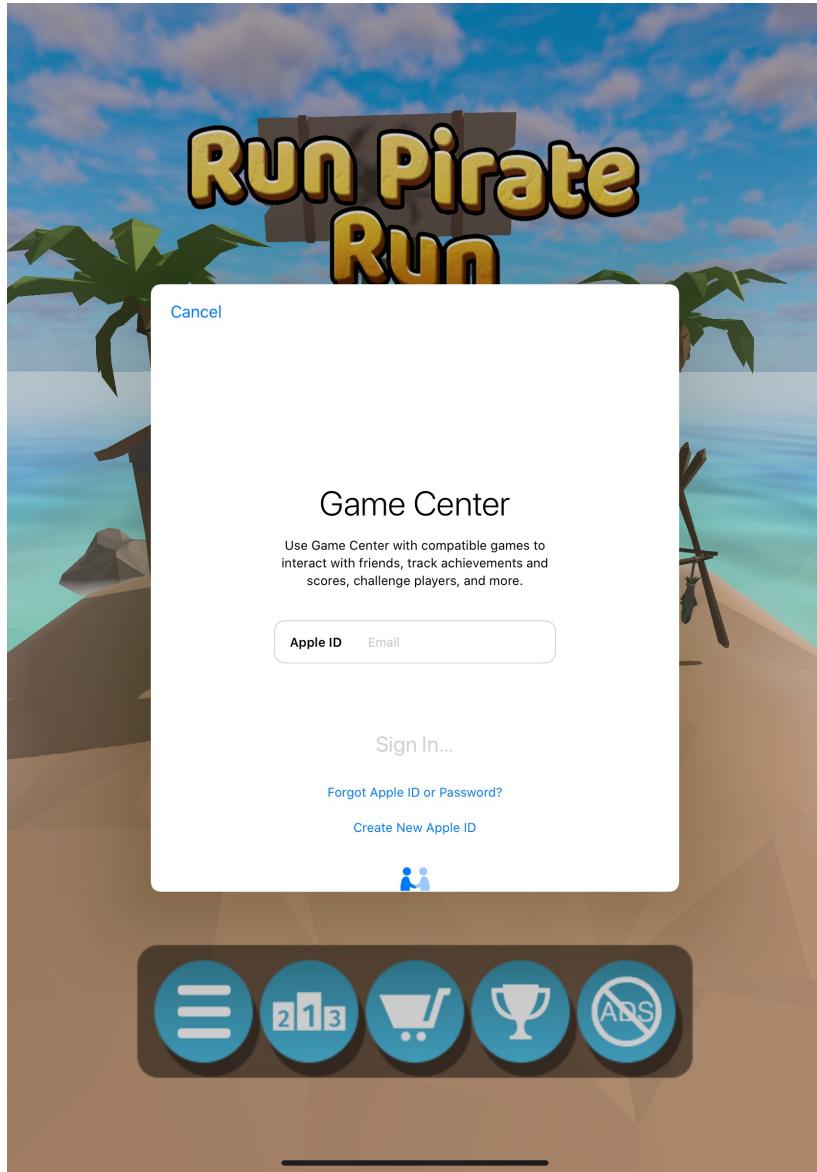
Testing methodology:



Proof of
working
features:







Cancel

Game Center

Use Game Center with compatible games to interact with friends, track achievements and scores, challenge players, and more.



Your gameplay information, including the games you play and who you invite and play with, is used to provide and improve Game Center features.
[See how your data is managed...](#)

Cancel

Game Center

Use Game Center with compatible games to interact with friends, track achievements and scores, challenge players, and more.

Apple ID	mavros_assassin@hotmail....
Password	••••••••

[Sign In...](#)

A virtual keyboard is displayed at the bottom of the screen, showing the letters q through p in the top row and a through l in the second row. Below these are rows for punctuation and numbers. The 'space' key is prominent, followed by a blue 'go' button.

Cancel

Two-Factor Authentication

A message with a verification code has been sent to your trusted devices. Enter the code to continue.

[Didn't get a verification code?](#)

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
0		✖

Cancel

Game Center

Use Game Center with compatible games to interact with friends, track achievements and

The Apple ID you entered couldn't be found or your password was incorrect. Please try again.

[OK](#)

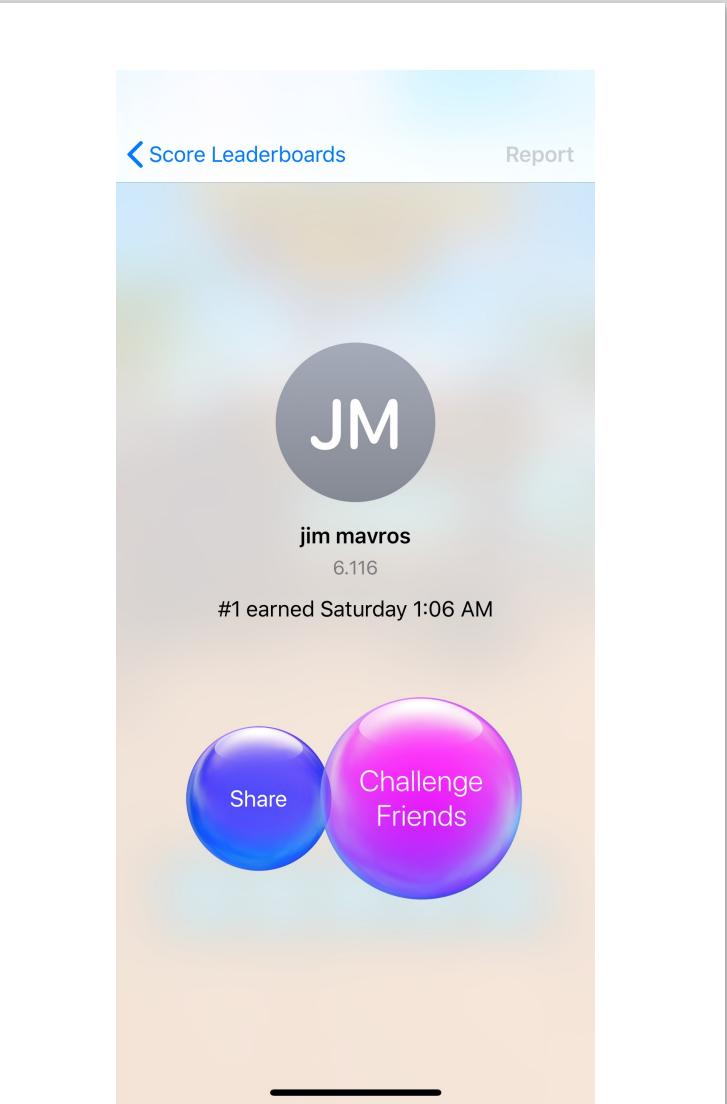
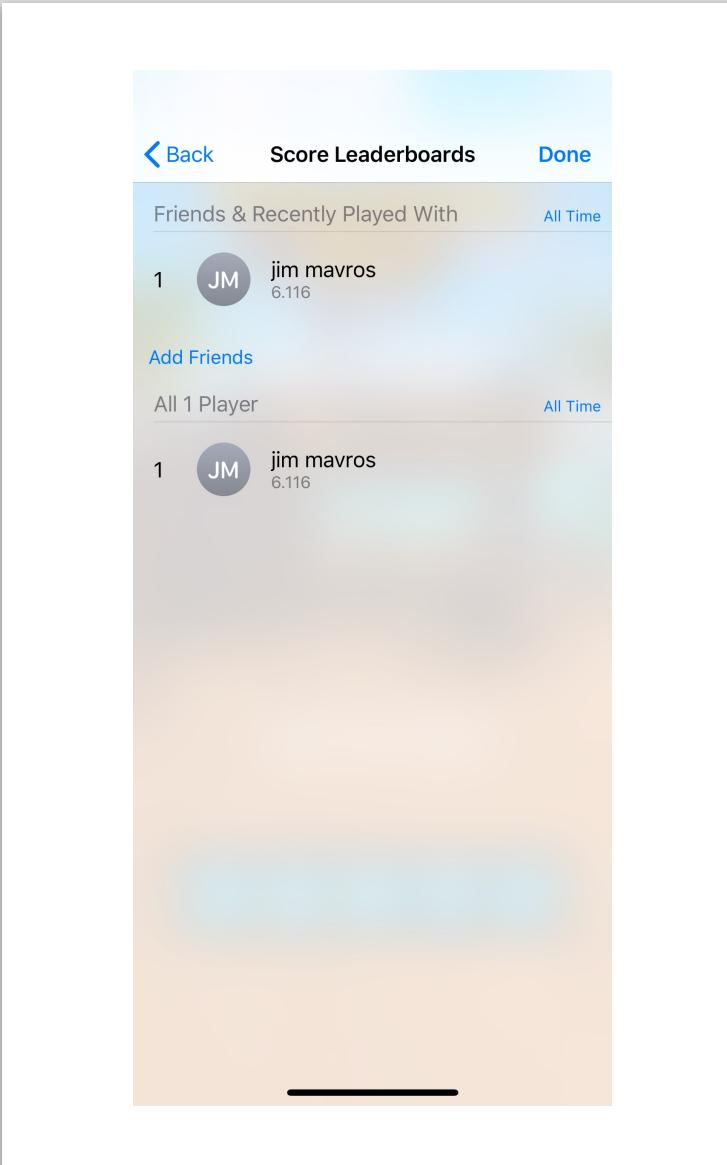
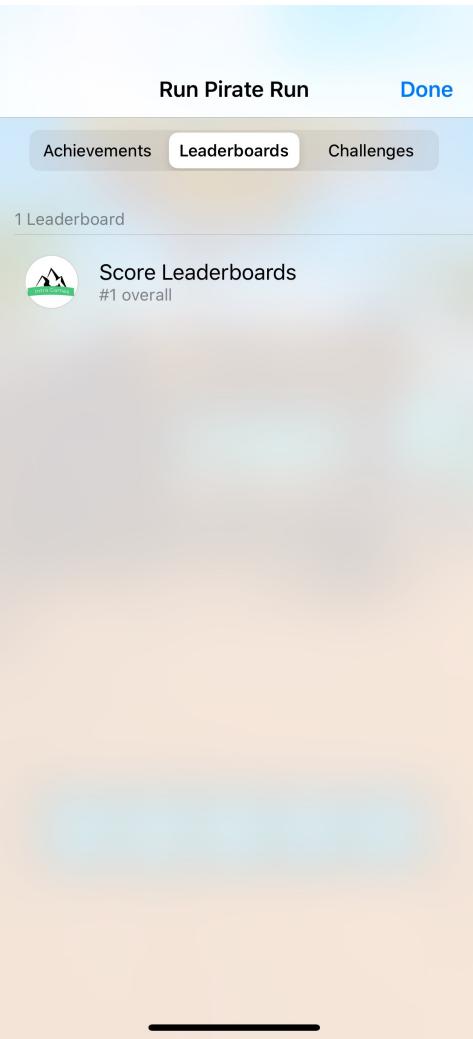
[Sign In...](#)

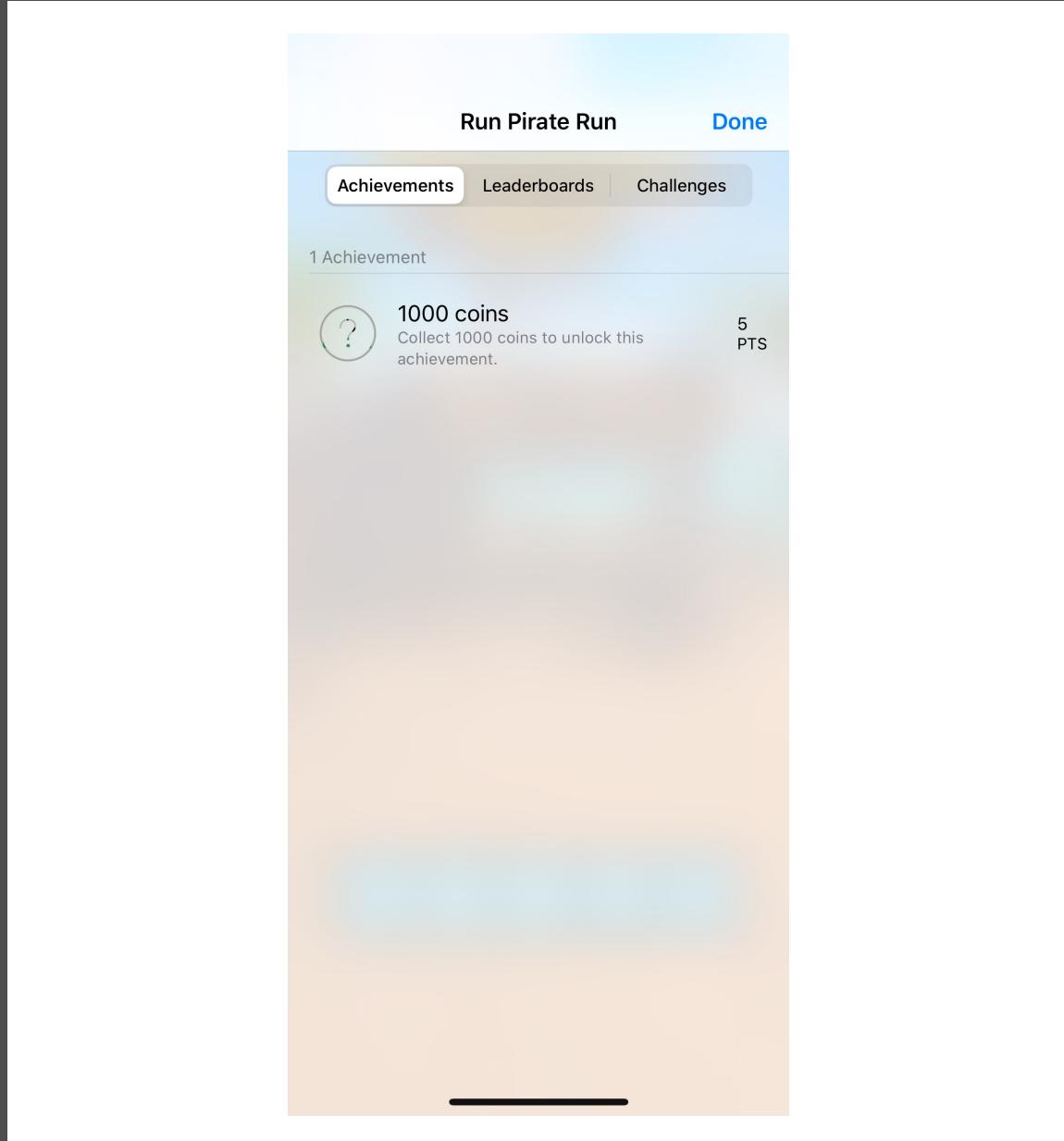
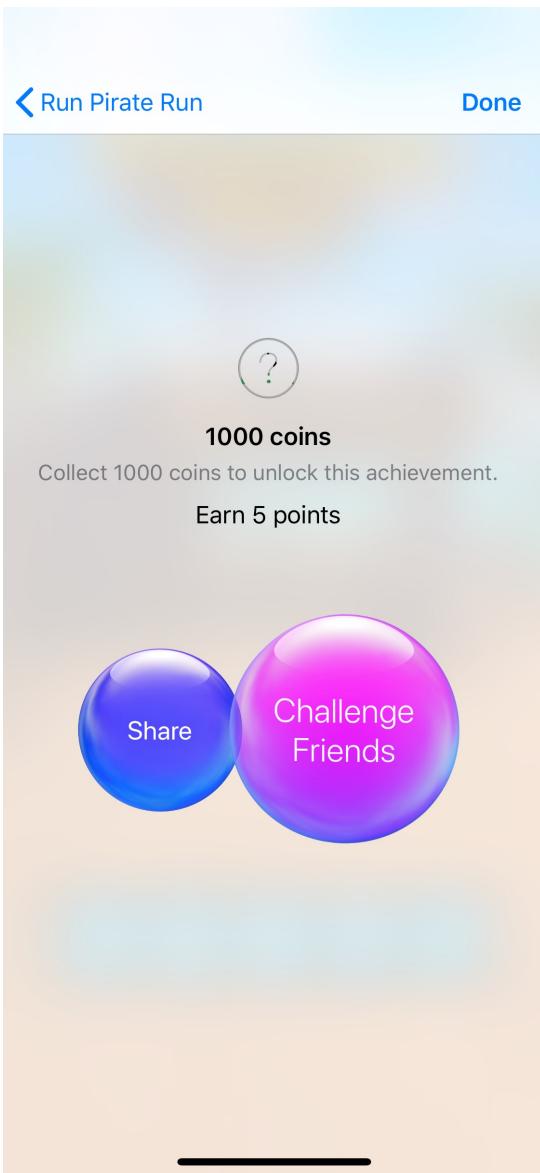
[Forgot Apple ID or Password?](#)

[Create New Apple ID](#)



Your gameplay information, including the games you play and who you invite and play with, is used to provide and improve Game Center features.
[See how your data is managed...](#)





2:27



◀ Apps



Run Pirate Run

Version 0.1.1 (1)
Expires in 90 days

INSTALL

What to Test

Software Development Capstone Project

Developer: Dimitrios Mavrofrydis

Supervisor: Prof. E. Vagianou

more

✉ Send Beta Feedback

Information

App Details >

Notifications Push, Email >

Previous Builds >

Stop Testing >

From the Developer



Run Pirate Run

Version 0.1.1 (1)
Expires in 89 days

What to Test

Software Development Capstone Project

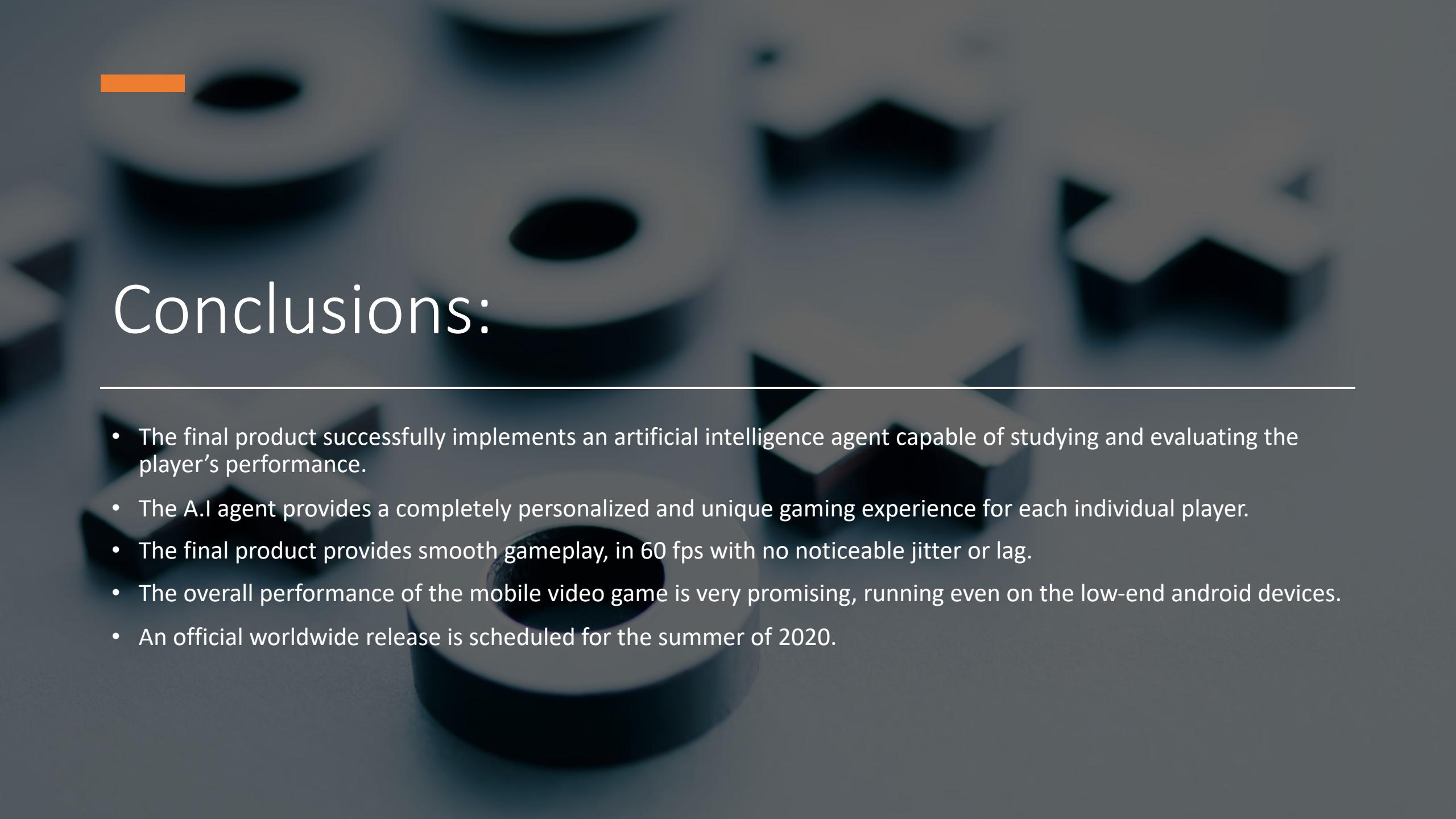
Developer: Dimitrios Mavrofrydis

Supervisor: Prof. E. Vagianou

The American College of Greece – Deree

Spring Semester 2020

Next



Conclusions:

- The final product successfully implements an artificial intelligence agent capable of studying and evaluating the player's performance.
- The A.I agent provides a completely personalized and unique gaming experience for each individual player.
- The final product provides smooth gameplay, in 60 fps with no noticeable jitter or lag.
- The overall performance of the mobile video game is very promising, running even on the low-end android devices.
- An official worldwide release is scheduled for the summer of 2020.

Implementing artificial intelligence mechanisms in mobile video gaming.

Prepared by: Dimitrios Mavrofrydis
Supervised by Prof. E. Vagianou
ITC 4918 – Software Development Capstone
Project Spring Semester 2020
