



**RMIT International University Vietnam**  
**Assignment 2 - Individual Project - Space Game Report**

<b>Subject Code:</b>	<b>COSC2659</b>
<b>Subject Name:</b>	<b>iOS Development</b>
<b>Location &amp; Campus</b>	<b>SGS Campus</b>
<b>Student name:</b>	<b>Nguyen Quoc An</b>
<b>Student Number:</b>	<b>s3938278</b>
<b>Lecturer's name:</b>	<b>Mr. Tom Huynh</b>

*"I declare that in submitting all work for this assessment I have read, understood and agreed to the content and expectations of the Assessment Declaration."*

*Disclaimer: All data provided in this application and report is for the purpose of this assessment only, and any act of spreading or commercializing this data is strictly prohibited.*

<b>I. Introduction.....</b>	<b>2</b>
<b>II. Motivation and inspiration.....</b>	<b>2</b>
<b>III. How to play the game.....</b>	<b>2</b>
Rules of the game:.....	2
1. Defend your spaceship.....	2
2. Shoot the enemy.....	2
3. Upgrade your spaceship.....	2
4. Defeat the boss.....	3
5. Unlock achievements.....	3
Tips and tricks:.....	3
1. Aim Precise.....	3
2. Timing is everything.....	3
3. Watch for patterns.....	3
4. Collect crystal and upgrade.....	3
<b>IV. Game Features.....</b>	<b>3</b>
GameManager.....	3
Register Scene.....	3
Main Menu Scene.....	4
Game Scene.....	4
1. Player.....	4
2. Health bar.....	4
3. Score.....	4
4. Crystal and crystal label.....	4
5. Bullet.....	5
6. Enemy.....	5
7. Spawn Manager.....	6
Game Win Scene.....	6
Game Over Scene.....	6
Map Scene.....	6
Shop Scene.....	6
Leaderboard Scene.....	6
Achievement Scene.....	7
How To Play Scene.....	7
<b>V. Testing Instructions.....</b>	<b>7</b>
<b>VI. Project Demo.....</b>	<b>7</b>
<b>VII. Conclusion and Future Development.....</b>	<b>7</b>
<b>VIII. Resources used in this project.....</b>	<b>7</b>
<b>IX. Appendix.....</b>	<b>8</b>

## **I. Introduction**

Space Game is a mobile shooting game of the type “Shoot’ em up” game, inspired by a very popular video game in the 2000s called "Chicken Invader". The gameplay, in summary, is fairly simple to the player, as stated in its category, shoot all the enemies, upgrade your spaceship, and advance to the next level. Throughout the playing journey, your play history is recorded, which can be shown in the leaderboard, and unlock achievements.

## **II. Motivation and inspiration**

The choice of re-making a well-established shooting game is because the space-shooter game genre has always been an icon in the history of gaming. Starting from the original arcade game machine and long-lasting until now. These games provide the player with excitement in action gameplay and the challenging feeling of advancing through the game.

Personally, I found this type of game has become an unforgettable childhood memory for everyone and for me. So through this project, I want to bring back the nostalgia feeling of the old days by this Space Game. This is also a great opportunity for me to dive into the realm of game development and design, learning and understanding the mechanics and principles of designing a successful space shooting game.

In the following section, I will go through the gameplay and present an overall analysis of the game implementation, as well as the challenges encountered during the development.

## **III. How to play the game**

In any game, understanding the rules and mastering the game mechanics is the key to victory. Below is an overview of how to play Space Game, along with valuable tips and tricks to help players advance through the game.

### **Rules of the game:**

#### **1. Defend your spaceship**

When playing the game, the player will act as a space pilot to fly their spaceship to fight the incoming waves of enemy ships. There will be 7 different types of enemy spaceships with different properties which will be discussed later. Those enemies will try to move, shoot, and try to take you down by any chance. So the first objective is simple, survive through all the enemies after a specific amount of time, and you will win the level.

#### **2. Shoot the enemy**

To combat against enemies, the player’s spaceship is equipped with a gun shooting continuously forward. So use your precise aiming skills to shoot down the enemy while avoiding their incoming fire.

#### **3. Upgrade your spaceship**

The game difficulty will increase significantly as the game progresses. Therefore, the game has brought you a spaceship upgrade system. In each game, when destroying enemy ships,

you will have a rate of 33% to drop 1 to 5 crystals. Collect these crystals and use them later in the shop to buy a more powerful spaceship, which will give you more health.

#### **4. Defeat the boss**

The ultimate and final challenge of Space Game will be the boss waiting for you at level 12. The boss will be much more powerful than the normal enemies, so you will need to prepare yourself with skills and strategy to defeat it.

#### **5. Unlock achievements**

While you are progressing through the game, your score and play history will be recorded. You can view your rank in the leaderboard compared to other players, see your statistics, and unlock 5 different achievements in the achievement menu.

### **Tips and tricks:**

#### **1. Aim Precise**

Taking time to line up your shots and taking down incoming enemies, can boost your chance of winning significantly.

#### **2. Timing is everything**

Despite not destroying all the enemies, you will still win the level after an amount of time passed. So if your firepower is not enough, prioritize surviving until the timer.

#### **3. Watch for patterns**

The incoming enemies will come in waves, they move and shoot in a pattern. Observe and remember their pattern to plan the attack and find a safe zone for you.

#### **4. Collect crystal and upgrade**

Upgrading your ship will grant you more health, and there will be more room for your errors. So collect as many crystals as you can and use them wisely.

## **IV. Game Features**

Space Game is made using 3 native libraries of iOS: SpriteKit, SwiftUI, and UIKit, with SpriteKit being used the most to provide in-game graphics, animation, game logic, and physics. UIKit is used for the register screen, and SwiftUI is used for some built-in functions.

### **GameManager**

The GameManager class is a class to stores all UserDefaults data, global variables, and functions that will be used repeatedly in other classes. This class is declared as a singleton class, to ensure consistency of data in the game.

### **Register Scene**

The first time the player accesses the game, the game will check the UserDefaults data for key "playerName" in GameManager, if it's equal to nil then the register scene will be displayed for the player to enter their desired name (Appendix 1). After that, the name will be saved and the user will be directed directly to the main menu scene.

## **Main Menu Scene**

The Main Menu scene is the first entrance of the player when they enter the game after they have registered their name (Appendix 2). This scene will have 4 main buttons, to navigate to another scene:

- **Start Button:** This button will take the player to the Game Scene at the highest level that they are currently at.
- **Map Button:** This button will take the player to the Map Scene, to choose which level to play.
- **Ship Shop Button:** This button will take the player to the Shop Scene, so they can view and buy different spaceships.
- **LeaderboardButton:** This button will take the player to the LeaderboardScene to view their current ranking.

Besides that, there are 3 additional buttons at the bottom right corner of the screen:

- **Mute Music Button:** Toggle this button will mute the background music of the whole game. The default value is set to off.
- **Mute Sound Button:** Toggle this button will mute the sound effects of the whole game. The default value is set to off.
- **How To Play Button:** This button will take the player to the How To Play Scene.

## **Game Scene**

Game Scene is where the game will occur, this scene will continuously render all the in-game objects and check for game logic (Appendix 3), these objects will be discussed below.

### **1. Player**

The player object includes health, skin, and trail emitter attributes. The values of those attributes are stored by UserDefaults data in the Game Manager class and initialized when the player enters the Game Scene. At the start of every level, those attributes are read, and the player's position will be rendered at the bottom of the screen.

### **2. Health bar**

The health bar will be rendered at the top left corner of the Game Scene, depending on the player's health. The health value will be deducted if the player is hit by the enemy or the enemy's bullet.

### **3. Score**

The score label will be rendered at the top right corner of the Game Scene, this label will be reset to 0 every time the player enters a new level. The player gets 1 score for every bullet hit to the enemy, and 5 scores for every enemy kill. At the end of every game, this score will be checked if its value is larger than the "highScore" UserDefaults data, then this score will become the high score and updated in the leaderboard.

### **4. Crystal and crystal label**

Crystal objects are spawned at the position of an enemy when they are killed. There is roughly a 33% rate of dropping crystals when a player kills an enemy, and there will be

randomly 1 to 5 crystals will be dropped. This crystal will spread in random directions, fall down, and disappear after 3 seconds of existence. Players can obtain these crystals by touching them (Appendix 4).

The score label will be rendered at the top right corner of the Game Scene, under the score label. This label will keep track of the amount of crystal obtained by the player at the current level and will be added to the “playerMoney” UserDefaults value after every level, and they can be used to buy more spaceships in the shop.

## **5. Bullet**

Bullet objects are spawned when the player or an enemy shoots, they will move in a designated pattern. Each bullet object consists of a skin, sound effect, and damage, but to simplify the game's hardness and auditory effects, in the current build of the game, all bullets will cause 1 damage, and only player bullets have sound effects.

## **6. Enemy**

There are 7 types of enemies which are inheritance of the parent class Enemy, they all have a health bar but each has a different skin, health, moving pattern, bullet, and shooting pattern (Appendix 5).

- **VerticalMovementEnemy:** This enemy will be spawned in a random position at the top of the screen and move forward until the end of the screen. They will shoot bullets in a circular spread shape.
- **HorizontalLeftMovementEnemy:** This enemy will be spawned in a random position from the left side of the screen and they will move to the right side of the screen. This enemy will not shoot any bullets.
- **HorizontalRightMovementEnemy:** This enemy will be spawned in a random position from the right side of the screen and they will move to the left side of the screen. This enemy will not shoot any bullets.
- **FixedMovementEnemy:** This enemy will be spawned at the top of the screen, and positioned evenly across the screen width. They will move vertically down the screen to a certain point while jiggling horizontally. They will shoot bullets that go to the player's current position and disappear.
- **RandomMovementEnemy:** This enemy type is a projectile, it will be spawned at the top of the screen with a random position, and move to another random position at the bottom of the screen.
- **CircularMovementEnemy:** This enemy will be spawned from the right side of the screen, and move in a circular motion across the screen, back and forth until it is destroyed. This enemy will shoot bullets in a straight line from its position to the end of the screen.
- **BossEnemy:** This enemy will be spawned on level 12, its moving pattern is the same as FixedMovementEnemy. It has a bigger size and shoots all 3 types of bullets of FixedMovementEnemy, CircularMovementEnemy, and VerticalMovementEnemy.

All enemy types have 5 health, except for RandomMovementEnemy have 10 health, and BossEnemy have 400 health. All have a health bar underneath itself, which will be updated according to their current health.

## **7. Spawn Manager**

Spawn Manager is a designated class used to spawn enemy waves depending on the level. When a Game Scene is loaded, it has a level parameter passed to from the Map Scene. The spawn manager will read this parameter and spawn the enemies accordingly, afterward, they will end the level with a win action if all the enemies are spawned and the player still survived.

## **Game Win Scene**

This scene will appear after the player has successfully survived a level, it will display the score as well as the high score of the player (Appendix 6). There will also be 3 buttons in this scene:

- **Replay Button:** This button will replay the previous level the player just played.
- **Continue Button:** This button will direct the player to the map scene.
- **Close Button:** This button will direct the player to the main menu scene.

## **Game Over Scene**

This scene is similar to the Game Win Scene, but the star will be gray (Appendix 7).

## **Map Scene**

The Map Scene will display all 12 levels, and the path connected between them. When accessed, the map scene will read the UserDefaults data of key “currentUnlockLevel”, and render the level node. If the level attribute of the node is less than the key value, which means the level is unlocked, the map will render the node in full color, else it will be grayed out (Appendix 8).

The player can navigate through the map by scrolling the map up and down and clicking on the level node to play the desired level. If the level is unlocked, then they will be directed to the Game Scene, else they will receive an alert saying the level is locked (Appendix 9). If the player wins the current highest level, then the next level will be unlocked.

## **Shop Scene**

The Shop Scene will display all 5 spaceships and their health value as well as the number of crystals the player currently has. There will be a back and forward button to navigate to view different spaceships. The first spaceship is owned by default, and the other 4 will have a price for the player to buy with their crystals. If they bought the spaceship, they can use the equip button to choose that spaceship, else they will receive an alert saying they have not bought the spaceship (Appendix 10). This Scene will read the UserDefaults value for key “shipBought”, which is an array of numbers, to identify which ship has been owned.

## **Leaderboard Scene**

The Leaderboard Scene will be the place for the player to see their ranking compare to other players. The rankings are placed depending on the player's high score, and the player can

navigate through the leaderboard by pressing the back and forward buttons (Appendix 11). This scene also contains an achievement button which will direct the player to the Achievement Scene.

### **Achievement Scene**

The Achievement Scene will display 5 achievement badges, that the player can obtain, along with their progress which is saved in the UserDefaults data. The badges are grayed out by default and will be rendered in full color if the player achieves it (Appendix 12).

### **How To Play Scene**

The how-to-play scene will display the instructions and the rules of the game. The player can navigate through different pages by pressing the back and forward buttons (Appendix 13).

## **V. Testing Instructions**

The game can be run normally on Xcode without any additional configuration. However, there is an additional button called the Unlock All button in the How To Play scene, used for testing purposes. This button will grant the player 10,000 crystals and unlock all levels immediately.

## **VI. Project Demo**

Please refer to this link for the demo video: [COSC2659 - iOS Development - Space Game Demo - YouTube](#)

## **VII. Conclusion and Future Development**

In summary, the Space Game has effectively delivered core gameplay features, offering an engaging experience for players. Looking ahead to future development, I plan to introduce an enhanced weapon system, providing players with more upgrade choices and ways to spend their crystals. Additionally, the introduction of in-game power-ups will add an element of unpredictability and further elevate the player's excitement. To diversify the gameplay, I am committed to incorporating additional boss levels and mini-boss encounters, to ensure a diverse gaming experience.

## **VIII. Resources used in this project**

Ethnocentric Font: [Ethnocentric Font | dafont.com](#)

Void - Main Ship (1.0): <https://foozlecc.itch.io/void-main-ship>

Space Shooter Asset Pack: [\[8x8\] Space Shooter Asset Pack by Gustavo Vituri \(itch.io\)](#)

FREE SPACE SHOOTER GAME GUI: [Free Space Shooter Game GUI - CraftPix.net](#)

Background: [\(3\) Part 1 - Solo Mission \(Space Invaders\) - Make A Full iPhone Game In Xcode - YouTube](#)

Achievement: [Steam Community :: Apex Legends :: Achievements](#)



## IX. Appendix

### Appendix 1: Register Scene



Source: Nguyen, QA 2023

### Appendix 2: Main Menu Scene



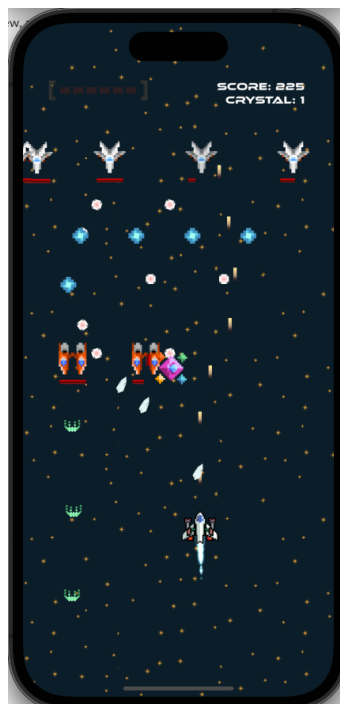
Source: Nguyen, QA 2023

### Appendix 3: Game Scene



Source: Nguyen, QA 2023

### Appendix 4: Crystal



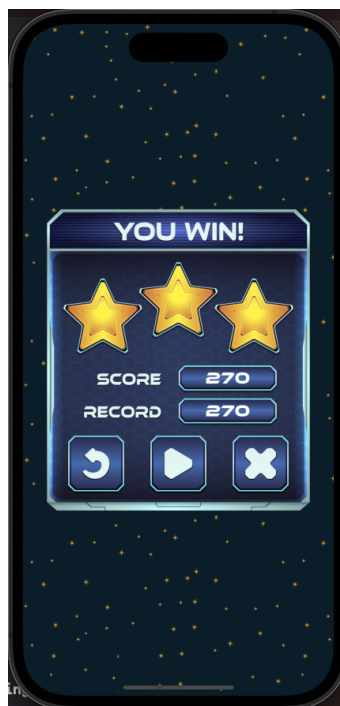
Source: Nguyen, QA 2023

### Appendix 5: Enemy



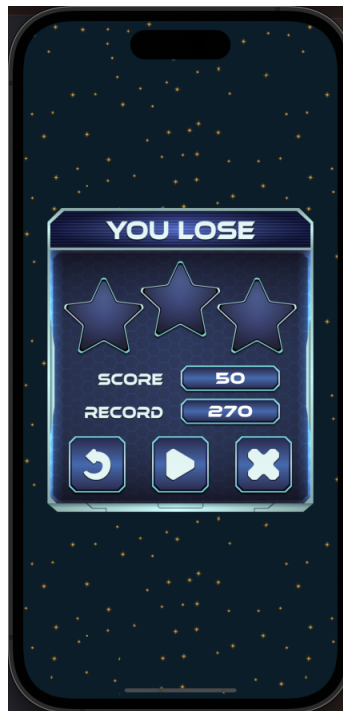
Source: Nguyen, QA 2023

## Appendix 6: Game Win Scene



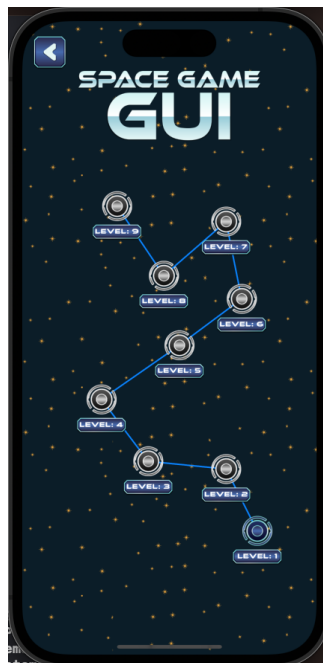
Source: Nguyen, QA 2023

## Appendix 7: Game Over Scene



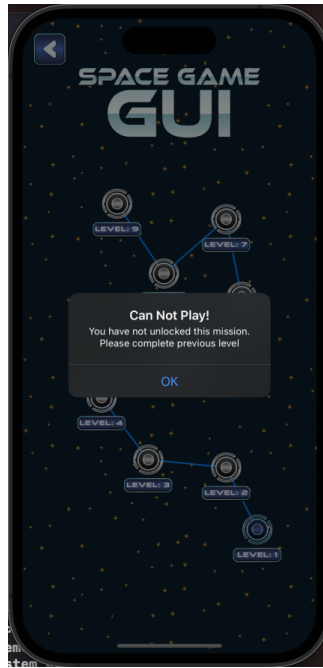
Source: Nguyen, QA 2023

#### Appendix 8: Map Scene



Source: Nguyen, QA 2023

#### Appendix 9: Level Locked



Source: Nguyen, QA 2023

#### Appendix 10: Shop Scene



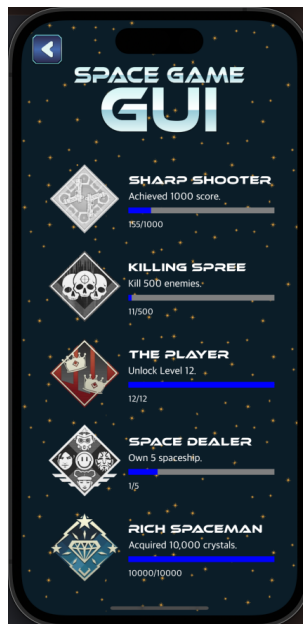
Source: Nguyen, QA 2023

#### Appendix 11: Leaderboard Scene



Source: Nguyen, QA 2023

## Appendix 12: Achievement Scene



Source: Nguyen, QA 2023

## Appendix 13: How To Play Scene



Source: Nguyen, QA 2023