

CSD3156 Mobile Group Project



Members

Leong Jia Yi Celine	2002392
Tan Ek Hern	2100948
Bryan Boh	2100716
Chua Zhen Xun	2102308
Huang Wei Jhin	2101192

Table of Contents

App Proposal	3
Objective & Description	3
Design Breakdown & Functionality	4
Software Architecture	9
Key Features	10
Development Process	12
Work Delegation & Contribution	12
References	12
Video Links	12

App Proposal

In today's bustling world, filled with hectic schedules and constant demands, it's crucial to provide people with an avenue for entertainment and relaxation.

Introducing "**Space Shooters**," a thrilling and immersive 2D space shooter game designed to captivate users and provide an exhilarating escape from the stresses of daily life.

Objective & Description

"Space Shooters" is a 2D space shooter game that combines cute graphics and engaging gameplay. In a galaxy, players step into the shoes of a skilled space alien pilot tasked with defending the universe from the impending doom of falling asteroids. The game offers a perfect blend of action, strategy, and excitement, providing players with an enjoyable and challenging gaming experience.

How to Play

Movement: Tap to shoot, tilt the phone to move

Pick-Up Items: Shield (invulnerable for 10 seconds), 4 guns (different bullet sizes, direction and speed), health

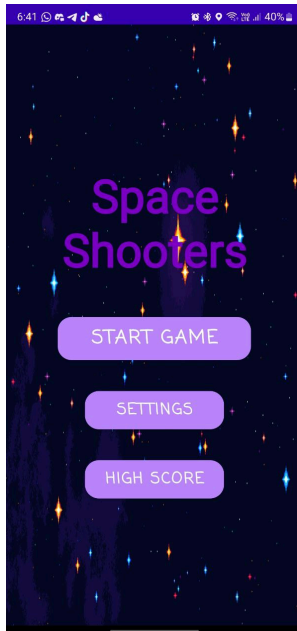
Score: Calculated by: $\text{Time} + \text{Shots} - \text{Hits}$

Asteroids falling gradually get faster as the game progresses, increasing the challenge of the game. If the player shoots the asteroid the asteroid's falling speed will increase even greater. Thus the player needs to decide if they want to shoot and gain points or avoid keeping the pace slower.

Pick-up items appear randomly during the game. The Shield makes the player invulnerable for 10 seconds, with a blue bubble that will decrease in size when the time is almost up. Health pick-up adds 1 life to the player's health but does not decrease the number of hits which will be used to calculate the final score of the game. 4 different gun pick-ups are available and change the bullets the player shoots. Players can pick up the weapon to change weapons during the game. Each gun has a unique shooting sound.

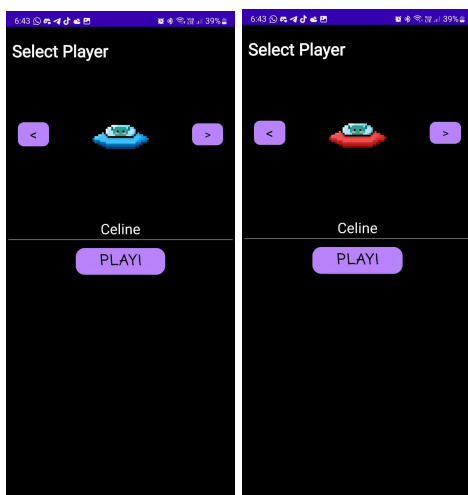
Design Breakdown & Functionality

1) Main Menu

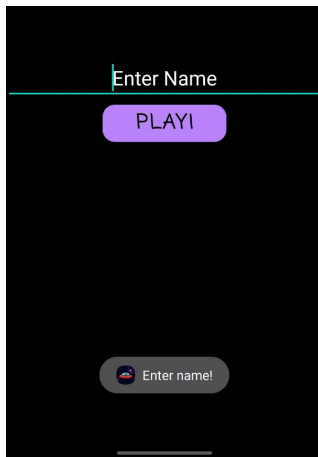


The Main Menu consists of 3 buttons, 'START GAME', 'SETTINGS', and 'HIGH SCORE'. Taking players to the respective screens.

2) Start Game

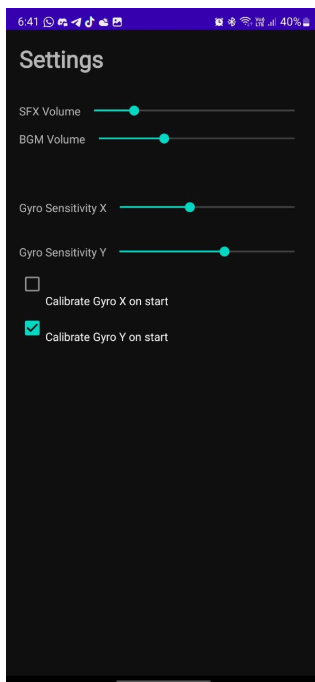


Upon clicking the start game button, the user will be able to enter their name and choose the color of their UFOs (red, yellow, blue). The player's name and spaceship used will be saved as user preference in the preference data store.



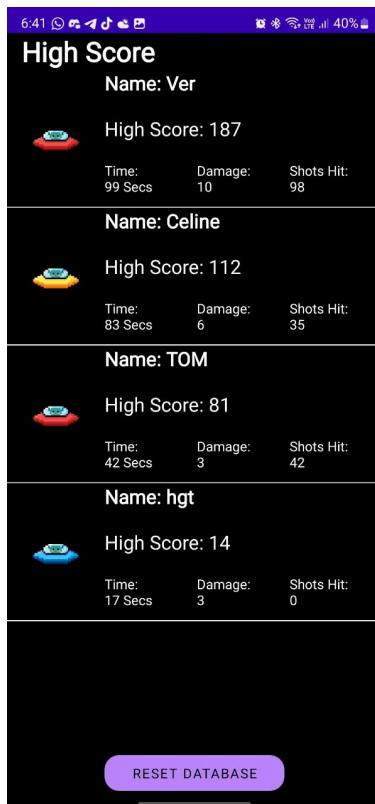
An error message appears when no name is entered upon pressing the play button. The game will not be able to commence till done so as the player name is used as the primary key to save their score to the database.

3) Settings



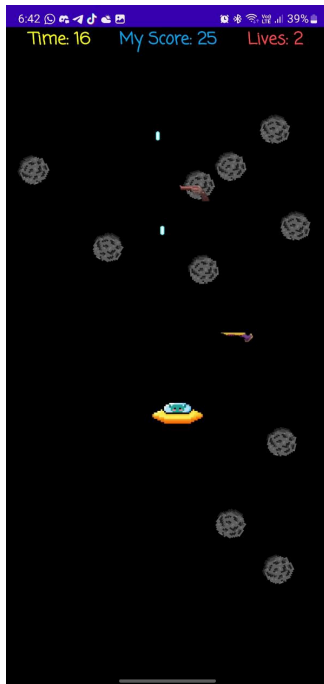
The user can adjust the game volume according to their liking. This preference will be saved. The user can also choose to calibrate their neutral position of the gyroscope when the game starts by checking calibrate X or calibrate Y. By default, the neutral position is at a 45-degree angle. There is also a hidden Endless Mode button somewhere~.

4) High Score



The high score screen sorts the data according to the highest score to the lowest score. It saves the player's choice of UFO color, high score, player's name, time, damage and shots hit. Players can compare their results to other players and motivate themselves to do better. A button at the bottom of the screen "RESET DATABASE" allows the user to clear all highscore data and start afresh. The player's score will only be entered in the database if they have achieved a higher score than their previous game (same name) or if it is their first time.

5) Gameplay



The gameplay screen shows the scoreboard and the game screen itself. The scoreboard consists of the time, score and lives left. The player's movements are controlled by the gyroscope by tilting their phone. Tilting sideways moves the player left or right. Tilting forwards or backwards moves the player front and back. The user has to maneuver around to avoid the falling asteroids or their lives will decrease. The asteroids gradually become faster as the game progresses, increasing the difficulty of the game. The player can also tap anywhere on the screen to shoot bullets to destroy the asteroids if they choose to do so. However, if they shoot an asteroid its speed will increase further when it spawns again. Hence players need to balance if they want to score the point of shooting an asteroid or keep the game pace slower by avoiding it. Players can also pick up weapons to change the bullet styles and sound effects. Furthermore, there are health and shield pickups to aid players in the game.

The game ends when the player has no more lives left. The final score is calculated by: total Time played + asteroids Shot down - asteroids Hits.

The game also has a tutorial mode that shows hints in the toast box that will appear in the game. This tutorial is only shown once when they first play the game. However, they can enable it again on the settings page.

Pickable Items



Gun 1: Normal bullet



Gun 2: Slower + Thicker bullet



Gun 3: Faster + Thinner bullet



Gun 4: Normal + Random direction

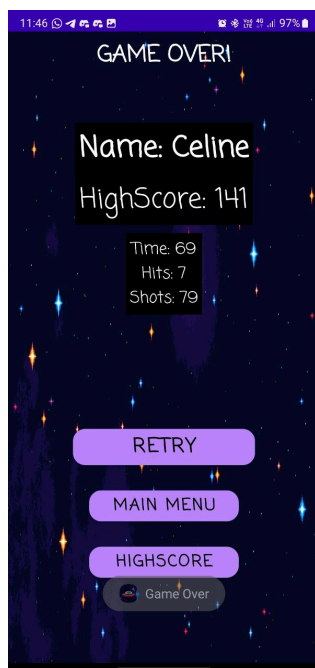


Health: +1 life



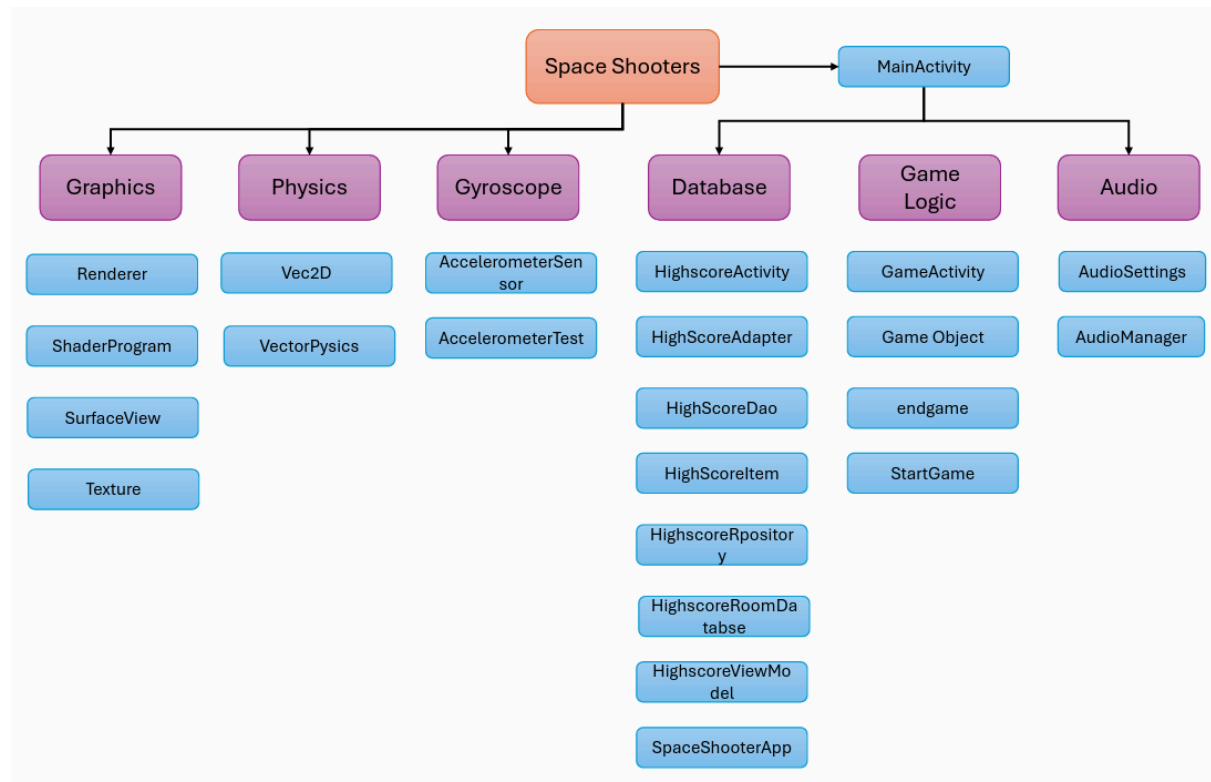
Shield: Invulnerable for 10 seconds

6) End Game



After the game ends, you will see the endgame screen. Which will display the name you entered, the high score for the run of the game, the time you survived, the number of times hit by an asteroid and the number of shots that hit the asteroid. You are also given the option to retry the game, go see the updated high score or go back to the main menu.

Software Architecture



The software architecture for our 2D vertical space shooting game, developed using Kotlin in Android Studio, adheres to a modular and performance-oriented design. The physics and collision system is intricately engineered to manage object dynamics and implement precise collision detection algorithms, ensuring a fluid and responsive gaming experience. Leveraging OpenGL graphics, the rendering pipeline optimally utilizes hardware acceleration, enhancing visual fidelity. The main activity encapsulates the game logic, orchestrating core functionalities and managing state transitions. Persistent high-score tracking is achieved through the integration of a RoomDB database system. The gyroscope functionality is seamlessly integrated to capture device motion data, allowing for the implementation of dynamic gameplay features. The audio subsystem is intricately designed, utilizing Android's AudioManager and MediaPlayer API to handle interactive sound effects. This architectural framework is meticulously crafted to harmonize these components, delivering a scalable and maintainable foundation for our 2D vertical shooting game on the Android platform.

Key Features

- **OpenGL**

The **Graphics** feature renders and updates the game objects. The animations help to make the game look more alive. The graphics system also uses 5 layers to allow layering of objects and ensures that some objects are always drawn on top.

- **Physics**

The **physics system** feature helps to calculate the acceleration force applied by the tilt (gyroscope) to maneuver the player's UFO. The physics system also checks for collisions when the UFO collides with an asteroid using a circle-circle collision. The position of each object is calculated by current position + velocity.

- **Gyro**

The **Gyroscope** is a system that uses the phone's gyroscope to detect movement which helps to contribute to the game's mechanics. When a tilt is detected, the player's UFO will move left or right according to the detected tilt.

The gyroscope functionality enhances user engagement by incorporating device motion for dynamic gameplay interactions. The gyroscope can be calibrated when the game starts by checking the calibrate X or Y in settings. This allows players to choose their neutral position and can even play lying down. By default, it is at a 45-degree angle.

- **Database**

The **database** keeps track of the high scores of all the players and their names, spaceships used, time elapsed, damage and shots. This feature helps to motivate the players to challenge themselves and beat their friends. The player's score will only be entered in the database if they have achieved a higher score than their previous game (same name) or if it is their first time. The user can clear the list of high scores if they wish to do so.

- **Game**

The **Game** starts at the main menu activity. From there players can go to the settings activity, high score activity or start the game. This will bring them to the start game activity where they can choose their spaceship and enter their name. Then the game activity begins and the player can play the game. Upon using up all their 3 lives, players will go to the end game activity which will show their score, time, damage and shots. From there they can navigate to the main menu, high score or restart the game.

- **Shared Preferences**

The settings pages allow users to configure their preferences for the game such as volume, gyro sensitivity and gyro calibration options. These preferences will be saved per user so that they will be the same the next time they open up the game. Additionally, the player's name and spaceship choice are also saved.

Development Process

1. Research and Analysis: Analyze the features and functionalities of existing vertical shooting game apps to identify gaps and opportunities for innovation
2. Create a plan for the app's features, functionalities, and user interface. Design the app's layout. Define the app's architecture, database structure and server requirements
3. Development of the app's front-end and back-end functionalities such as the UI menu and the game logic
4. Testing the app's functionalities, features and user interface. Address any issues and bugs identified during testing

Work Delegation & Contribution

Celine Leong: Project Set Up & menu, Documentation

Tan Ek Hern: Physics and Collision, Game Demo Video

Bryan Boh: Project Lead, Graphics, Game over screen

Chua Zhen Xun: Audio, Gyro

Huang Wei Jhin: Gameplay Logic, Database, Saved preference datastore, Gyro settings

References

The game's art is from Wei Jhin's software engineering game project

The game's audio is from Zhen Xun's software engineering game project

Video Links

Game Demo Video (link): <https://youtu.be/P2PJV030uCw>

Presentation Video (link): <https://youtu.be/xNtQtMbZWRg>