



Asteroid Space

Documentation of the First Assignment Intelligent Virtual Environments

Nipuna Weerasekara, Leduin Cuenca

Software and Systems Master Program
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Madrid, Spain
`{nipunalakshithawe47, leduin.cuenca}@alumnos.upm.es`

January 7, 2024

Contents

1	Introduction	2
2	Play Instructions	2
3	Project Structure	2
3.1	Game Scenes	2
3.2	Assets	3
3.3	Game Objects	4
3.4	Scripting and Game Mechanics	4
3.5	Interaction	5
References		5

1 Introduction



Figure 1: Splash Screen of Asteroid Space Game

the ranking of the scores from the main menu, the background of which is shown in Figure 1.

2 Play Instructions

When the player starts the game, they will be welcomed with a game splash screen and then in the main menu, the player is asked for their name. When the name is provided, the player can start the game. In the game, the player is tasked with controlling the spaceship. The spaceship has the ability to move upwards and downwards using up and down buttons respectively. In addition, the user can use left and right arrows to turn the spaceship while the spaceship has 360 degrees of freedom to move either clockwise or counterclockwise.

When the player sees an asteroid on the game screen, they can point and shoot at the asteroids with the gun equipped with the spaceship by pressing the space bar. Once the bullet hits the asteroid, the asteroid will be destroyed and the player will be given 5 points. If the asteroid hits the spaceship the score will be zero and if an asteroid hits the spaceship three times, the game will be over and the credits scene will be shown to the player.

3 Project Structure

In this section, we discuss the structure of the Unity project. For developing **Asteroid Space**, we used **Unity 2022.3.15f1** for Windows, Mac and Linux.

3.1 Game Scenes

Asteroid Space has 4 game scenes. First, when the player starts the game, the *Main Menu Scene* will be shown. In that scene, the initial game splash screen and the player name input are shown. In addition, there will be three buttons in the Main Menu Scene.

Asteroid Space is a Unity-based game with a retro-themed 80s space action game scenario. In the Asteroid Space game, the player controls a spaceship, where the player has to defend the spaceship against the coming asteroids. The spaceship is equipped with a state-of-the-art laser gun to shoot the coming asteroids. If the asteroid hits the spaceship, the player's score is reset to zero and the player has to start again. The player is given three lives and if all three lives are used, the game is over. Adhering to the 80s gaming theme, the player can see the current high score and

1. High Scores - used to see the high scores and it will load a new game scene called *High Scores Scene*.
2. Start Game - used to go to the Main Game Scene
3. Exit - used to exit the game

When the player enters their name and press *Start Game*, the *Main Game Scene* will be loaded. In the Main Game Scene, the game can be played and once the game is over, the *Game Over Scene* will be shown to the player. In addition, if the player wants, they can return to the Main Menu Scene via a button displayed in the Game Over Scene.

The Figure 2a is a screenshot from the Asteroid Space game where it shows the Main Menu Scene. The Figure 2b shows the screenshot of the High Scores Scene while Figure 2c shows an example frame from the Main Game Scene. The Figure 2d shows the screenshot of the Game Over Scene.

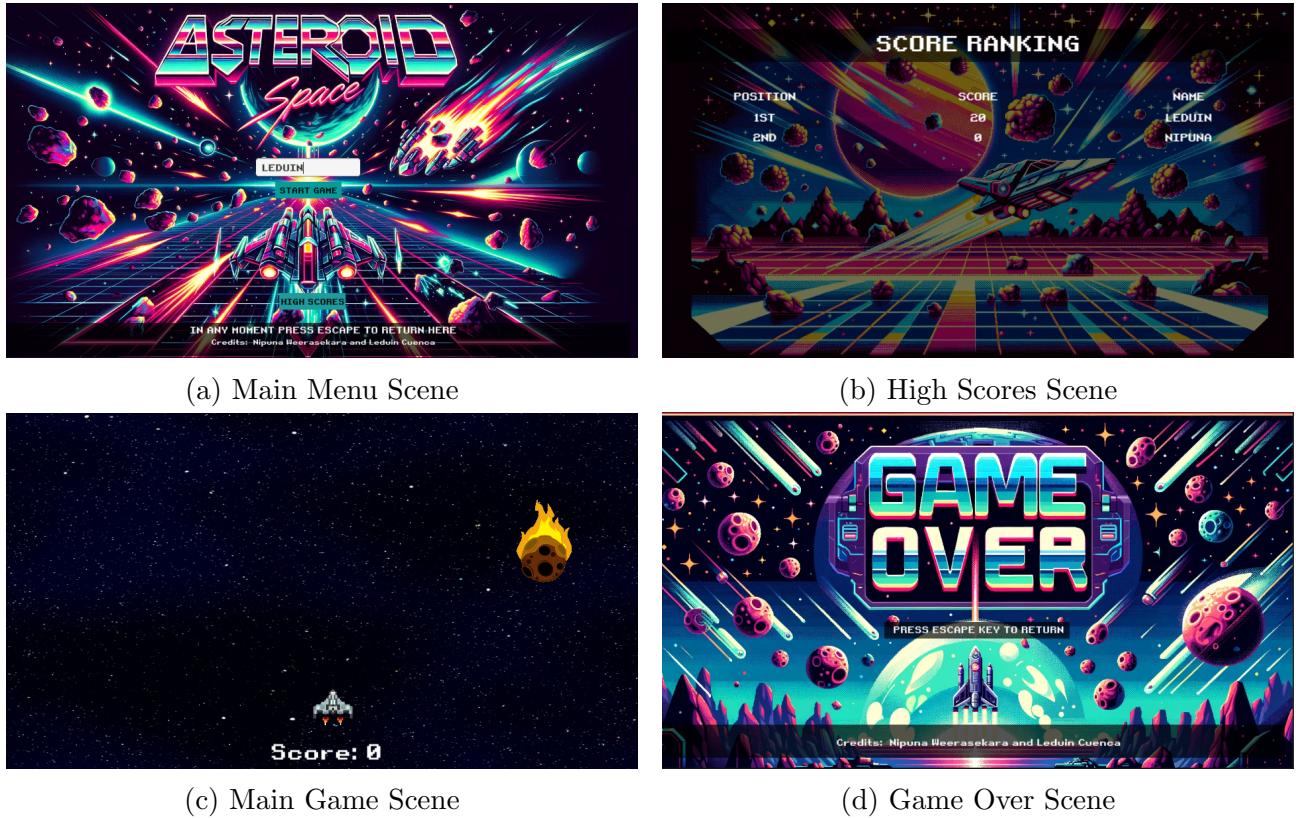


Figure 2: Screenshots from various scenes of the game

3.2 Assets

In our game **Asteroid Space**, we use several images as background images and Sprites. In addition, we use sound effects such as bullet firing sound effects, asteroid explosion sound effects, asteroid-spaceship impact sound effects, and game-over sound effects. To maintain the 80s retro theme, we use Synthwave music effects for the main menu scene, high scores scene

and the game-over scene. For the Synthwave music effects, we use royalty-free music for games from Pixabay [1]. In addition, for in-game sound effects, we use Pixabay and FreeSoundEffects.com [2]. The background images for the game were created using DALL·E 3 AI image generation model [3].

3.3 Game Objects

In our game, we utilise several game objects to develop each scene. **Player** game object is used to denote the spaceship where it has a Capsule collider component and a RigidBody 3D component. In addition, it has three audio source components for the sound effects such as bullet firing sound effect, spaceship-asteroid impact sound effect and game over sound effect.

We use orthographic **Main Camera** to deliver the game scenes and we use **Directional Light** game object for the lighting of the scenes. For the main background of the main game scene, we use **Space** game object with a Sprite Renderer. To spawn the asteroids, we use **EnemySpawner** game object. We use **Score** and **ScoreManager** game objects to handle the scoring of the player.

In the *ScoreRanking* scene, we used **Canvas** game object to have a backdrop and to view the score ranking. In addition, we used **Music** game object to play background music for each game scenes.

3.4 Scripting and Game Mechanics

In *Asteroid Space* game, the player is given a spaceship to handle. This is done by giving horizontal movement (rotation) via left and right buttons and the vertical movement via up and down buttons. For handling these inputs, we use a script called **Player** and with this script, we initialise the main player's rigidbody (the spaceship) and the constraints preventing the spaceship's movements. The spaceship will be bound inside the camera viewport. The bullets will be shot from a place designated in the spaceship and the above script is handling the instantiating of the bullets and the bullets will be disappear after a given time. The spaceship's collision detection is set to a capsule collision body and the bullet's collision detection is also set to a capsule collision body. When the meteor hit the spaceship, the spaceship will incur damages, which relates to resetting the game. If the bullet hits the meteor, it will incur damages, which relates to increasing the player's scores by five points. After the game has been reset three times, the game over screen will be shown and this is also handled with this **Player** script.

The **EnemySpawner** script is used to spawn meteors. These meteors will be spawn at a rate that increases over time within the bounds of the game screen. It calculates the spawning position of the meteors based on the camera's size and aspect ratio, instantiating meteors at random locations along the top edge of the screen. The **HighScoreTable** script manages the high score table in the game, loading and saving high scores using *PlayerPrefs* and displaying them dynamically in the game screen. It adds new scores, sorts them in descending order, keeps only the top ten, and refreshes the display to include ranks, scores, and names of the players while giving the feeling of a 80s retro arcade game.

3.5 Interaction

As for the interactions in the **Asteroid Space** game, the player can use the up and down buttons and the left and right buttons for vertical and horizontal movements respectively. When the player wants to move to the main menu, they press the escape button at any moment. In addition, this information is visible to the player when they turn on the game and first see the main menu of the game. In each game scene, the user can use the mouse to click the buttons and keyboard to enter input. For the main menu, we ask the player to give their name so that it would be visible in the high score menu. For this, the player can input their name on the keyboard. The keyboard input is taken as case insensitive and in the high score menu, the name would be shown as all-capitalised.

To inform the player via feedbacks, we employ music cues. When the game starts, we play a 80s retro-synthwave music to inform the player that they have entered to the game. Then when the player enter to the game playing mode, we will play another synthwave tune, while keeping in mind that the music shouldn't interfere with the player's concentration. Then, while playing the game, for each bullet the player shoots, the game will make a bullet firing sound and when the bullet hits a meteor, it will make an impact sound. In addition, when the meteor hits the spaceship, the game will play an explosion sound effect. To inform the player that the game will be over after three resets, when a meteor hits the spaceship for the third time (after two game resets), the game will play a short game over sound effect. When the player is shown the game over screen, the game will play another retro-synthwave music. Finally, if the player want to see the high score menu, they can go to main menu and see the high score menu. While viewing the high score menu, the game will play another synthwave music which relates to high score menus (triumphant music). All of these sound effects are obtained from royalty free sources [1, 2].

References

- [1] *80s Synthwave Game Music*. <https://pixabay.com/music/synthwave-80s-synthwave-game-music-112662/>. Royalty-Free Music - Pixabay.
- [2] *FreeSoundEffects*. <https://www.freesoundeffects.com/>.
- [3] *DALL·E 3*. <https://openai.com/dall-e-3>.