Meteor Game Documentation

Intelligent Virtual Environments course at ETSIINF UPM

Game Instructions

Controls

Ship Movement: Use either the WASD keys or the Arrow Keys to navigate the spaceship.

Shoot Lasers: Press the SPACE BAR to unleash powerful lasers from your ship.

Exit back to menu: Press ESC to return to the main menu.

Objectives

Your mission is to destroy a total of ten meteors using your ship's laser gun.

Meteors

Meteors randomly spawn across the top of the game environment, adding an element of unpredictability to each playthrough. After each successful hit the meteor gets destroyed and the score is increased by one. After being hit by a meteor, your score is reset to 0, increasing the challenge and requiring consistent skill.

Scoreboard

Highscores are recorded everytime player is hit by a meteor or destroys the 10th meteor and win the game. They are stored in PlayerPrefs registry under the distinctive name "PlayerX".

Project Structure

Game Scenes

1. Start Scene:

The introductory menu provides options to:

- Initiate the game
- Access detailed instructions
- View the scoreboard
- Quit the game

2. Instructions Scene:

- This scene shows vital information about the game, like controls and overall gameplay.
- Includes a button for navigation back to the start menu.

3. Scoreboard Scene:

- Displays the current high score stored in PlayerPrefs registry, identified by the name "PlayerX."
- Includes a button for navigation back to the start menu.

4. Game Scene:

- The primary gaming arena where players control the spaceship, tasked with obliterating meteors.

- Contains key game elements such as the ship, meteors, bullets and the ScoreManager.

5. End Scene:

- Credits the game's author(s).

- Provides a button for players to gracefully exit the game.

Assets and Game Objects

1. Ship:

- Represents the player's controllable spaceship, serving as the main conduit for interaction.

2. Meteor:

- An entity that poses a threat to the player; meteors must be destroyed to progress in the game.

3. Bullets:

- Fired from the ship, bullets are the primary means by which meteors are annihilated.

4. ScoreManager:

- A crucial script responsible for managing the scoring system and persisting highscores through PlayerPrefs.

5. Row Entry for Score:

- A graphical component displaying individual score entries within the scoreboard.

Scripts

PlayerController Script:

- Orchestrates player input for ship movement and laser firing. Resets the score on collision with meteor. Updates the score to registry. Changes to the end scene after 10 successfully destroyed meteors.

MeteorBehaviour Script:

- Destroys the meteor after 3 seconds prior to spawning.

BulletBehaviour Script:

- Dictates the behavior of bullets, determining how they interact with the game environment. Increases the current score by 1 after bullet collide and destroys a meteor.

ScoreManager Script:

- Provides methods to retrieve high scores sorted in descending order, add new scores, and save the updated scores to PlayerPrefs upon destruction or when explicitly called. The script utilizes JsonUtility for serialization and deserialization to store and manage player scores in the game.

EndMenu Script:

- Contains a function called Quit that exits the application when called by a quit button.

EnemySpawner Script:

- Spawns a meteor at specified rate with a random position on the top of the game scene.

EscToExit Script:

- Checks for the 'Escape' key press in the Update function and loads the Start menu scene using SceneManager when the key is detected.

InstructionsMenu Script:

- Contains a function called GoBackToMenu that, when invoked, loads the Start menu scene using SceneManager, facilitating navigation back to the main menu from the instructions menu.

RowUI Script:

- Defines a user interface (UI) row with TextMeshPro elements for displaying rank, name, and score, allowing for dynamic updating of these UI components in response to changes in the game.

Score Script:

- Defines a Score class, marked as serializable, with properties for the player's name and their corresponding score, facilitating the storage and manipulation of player scores within the game. The constructor initializes these properties upon instantiation.

ScoreData Script:

- Defines a ScoreData class containing a List of Score objects, providing a structured container for storing and managing player scores within the game. The constructor initializes an empty list upon instantiation.

ScoreUI Script:

- Populates a UI element (RowUI) with high scores retrieved from the ScoreManager during the Start phase. The script instantiates RowUI objects for each score, displaying the rank, player name, and score value on the UI. The scores are fetched from the ScoreManager and arranged in descending order before being displayed.

ScreenWrap Script:

- Enables the continuous wrapping of a GameObject to the opposite side of the screen when it moves outside the viewport boundaries. It utilizes the main camera to determine the viewport position and checks whether the GameObject is outside the viewport. If so, it calculates the new position on the opposite side and updates the GameObject's position accordingly, creating a seamless wrapping effect.

StartMenu Script:

- Contains methods to load different scenes based on player interactions. StartGame(): Loads the Game scene when the player initiates the game. GoToInstructions(): Transitions to the Instructions scene when the player selects the instructions option. GoToScoreBoard(): Directs the player to the ScoreBoard scene when they choose to view the scoreboard.