

Interactive Multimedia – Team Project (Space Shooter) – Contribution Statement

By

B00148227 John Layson, B00147637 Adam O'Shea

Contributions made by B00148227 John Layson

- Added the Set Boundary code to prevent the player from traversing beyond the game's borders (as part of the PlayerController script)
- Created the PlayerController script to set the player movement and boundary (later expanded and edited by my teammate).
- Created the GameManager script to organize the game's main setting such as starting game, setting difficulty values, referring to different scripts and components (e.g. SpawnManager script), updating score, showing game over, moving from different scenes and loading URL to the GitHub repository.
- Created the SpawnManager script to set the number and wait in seconds rate of enemies spawning in the scene (later expanded and edited by my teammate).
- Created different game scenes for the main menu, difficulty, and settings.
- Created UI with text and buttons on main menu, difficulty, and settings.
- Added DetectCollision script to make sure something is destroyed when objects touch each other in the game (later expanded and edited by my teammate).
- Created Score UI and its text on the main gameplay (later adjusted by my teammate)
- Added background music for the game scenes.
- Added MoveForward script to create basic projectile movement and speed, allowing it to move forward (later slightly edited by my teammate).
- Created the prefabs for the player and enemies.
- Created Projectile prefab with red textures

- Created a list of enemies to spawn in the SpawnManager
- Added UpdateScore() method on the DetectCollision to add score points for every enemy kill.
- Added the Sphere Colliders on the game objects such as the spaceships
- Created, condensed and encapsulated codes from the void Start() to different methods in the Game Manager to make it neater
- Migrated the SpawnEnemy coroutine at the GamaManager by referring the SpawnManager script as a variable.
- Added “difficulty” variable on the GameManager script to set the difficulty int value depending on which difficulty option is chosen (Easy, Medium, Hard). This is done by calculating it alongside the spawnWaitSeconds of the SpawnManager Script (later expanded by my teammate).
- Created the Projectile prefab and added a material into it.
- Helped my teammate setup the game scene by adjusting the position of the Player and Enemy prefab and freezing some of its Rigidbody coordinates.

Note: While I created the coding for basic movement, detect collisions, boundaries and instantiating enemies and projectiles, this was later expanded & edited by my teammate with advanced concepts & coding he has come up with.

Contributions made by B00147637 Adam O’Shea

- Alpha Version
 - Found free assets in the unity store. Added the Earth including its texture and star background to the scene. Added the spacecraft.
 - Modified PlayerController to use AddRelativeForce instead of using transform so that the rigid body physics works.
 - Added a physical material ‘bouncy’ to game objects such as the Earth, the Player, and Enemies to ensure they don’t phase through each other.
 - Added an original prefab called spacecraft to add basic features of all spacecraft like a weapon projectile empty to instantiate projectiles. And I made the Player and Enemy into variant prefabs with features unique to each.

- Made a more general detect collision script for the projectiles that can tell who fired and who can get hit, so only the other side can get hit (no friendly fire).
- Developed an enemy AI script, where the farthest and nearest targets are calculated, and the farthest target chosen. This displayed a better behaviour than choosing the nearest target.
- Added a speed multiplier that accelerates Enemies faster if they are far away from the Earth and slow that down if nearby.
- The enemy spacecraft only fire on targets directly in front of them within a cone of tolerance. Involves the use of vector calculus to calculate the angle between the direction a spacecraft is facing and the direction another spacecraft is located.
- Final Version
 - Created a new class Target that gathers information on Game Objects that the enemies are targeting and helps with calculations of multiple targets later. Using some querying features of C# that allows to group attributes of objects (e.g targets.Sum[x => x.distance]) for summing up or other calculation over a group of objects of the same class.
 - An additional feature of the enemy AI is to calculate the centre of all spacecraft on the same side as the enemy and sometimes go toward that. The enemies hive intelligence was greater after this.
 - Added a UI HUD text element displays the remaining impacts the Earth (as 'Earth lives') can take before game over.
 - Made score and earth lives text elements display a score at the start of a game and not just when an enemy or the Earth was hit.
 - Also add a speed limit factor to the speed multiplier. And made it a public variable to tailor the rate of acceleration manually as desired.
 - Added a particle system explosion effect the Earth and the spacecraft which plays upon destruction. There is a delay of up to 2 seconds before the game object is destroyed.
 - Added a particle system to the projectiles as they are moving.
 - Added sound effects sourced from unity asset store to the projectile when fired, and the spacecraft and the Earth when destroyed.

- Searched the code for errors or mistakes in logic that were overlooked. Such as:
 - Enemy spawning location logic: each coordinate metric like distance and angle of approach are randomly chosen just once to make a valid coordinate in space.
 - Enemy centre of mass calculation in the Target class
 - That each instance of the Game Manager between scenes worked to together with the others and setting some variables as static.
 - Null reference object errors are guarded against with if statements to verify the object still exists and an alternative if not.
 - Any Quaternion or Vector3 zero or null values are handles with alternative neutral values.

X 

Adam O'Shea
B00147637

X 

John Layson
B00148227
