Universidad Politécnica de Madrid

Intelligent Virtual Environments

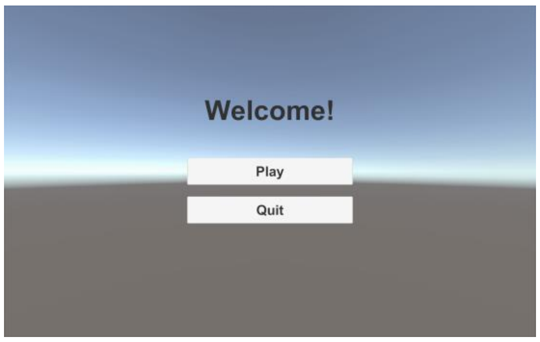
Assignment 1 [Github Link]

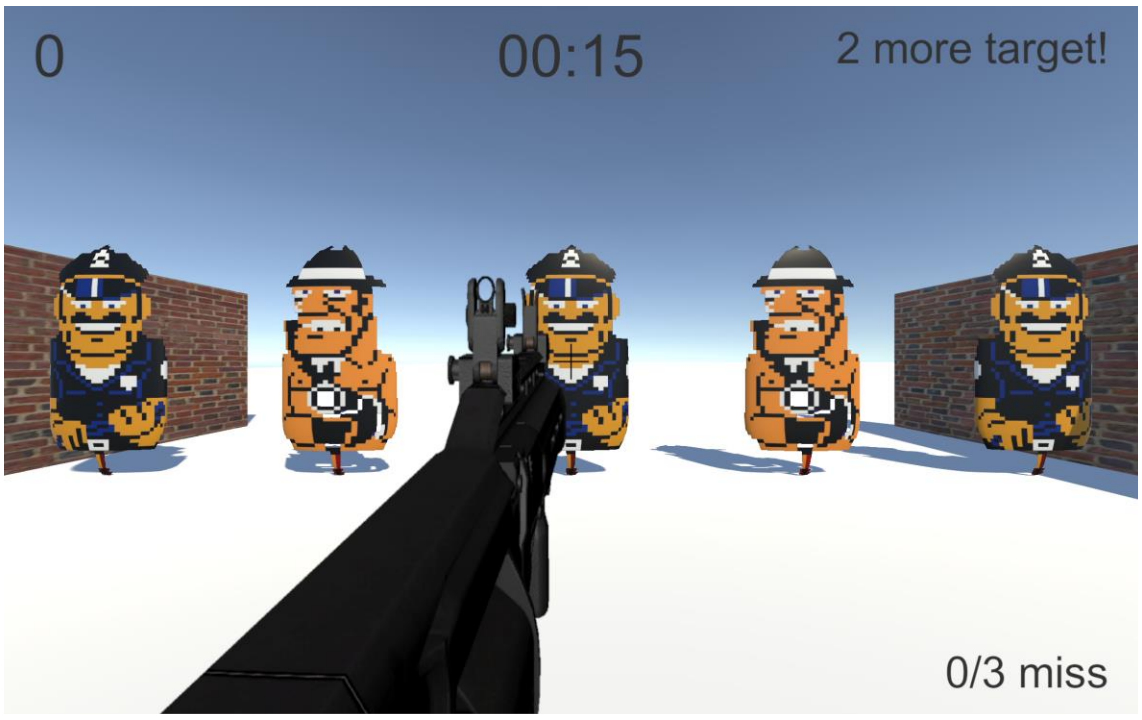
Antonio Franzoso	antonio.franzoso@alumnos.upm.es
Jose Luis Carrillo	joseluis.carrillo @alumnos.upm.es

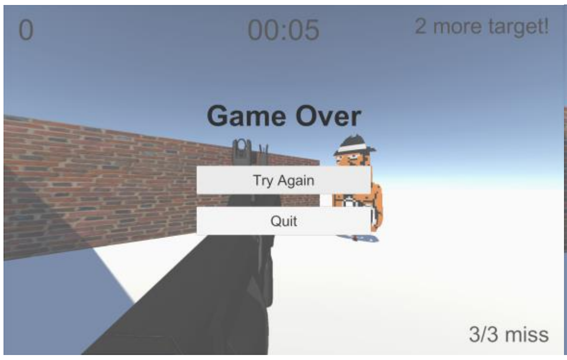
Content

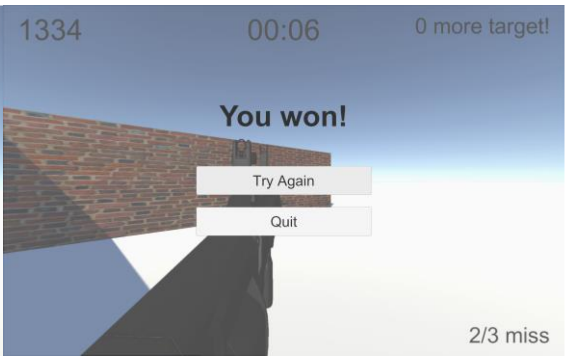
1. Objective

Develop a basic version of the classic video game *Hogan's Alley* in Unity3D, allowing the player to shoot enemy and ally NPCs. The main objective is to maximize the score by hitting enemies while avoiding shooting allies. The project includes basic functionalities such as a scoring system, a timer, pause, reset, and game-ending menus, as well as a real-time event logging system.









2. Game Features

a) Game Mechanics

- **Single Round Gameplay:** The game is structured around one decisive round where 5 characters (randomly selected between enemies and allies) spawn on-screen at fixed locations. using the mouse or spacebar.
- **Objective:** Hit every enemy while avoiding allies within the time limit to win the game.

b) User Interface (UI)

- Timer showing the remaining time.
- Score counter that updates dynamically based on performance.
- Pause, reset, and game-ending menus.

c) Logging System

- Key events such as NPC appearances and disappearances, shots fired, their results, scores, and game-ending scenarios are logged in a CSV file with timestamps.

d) Game Dynamics

- Randomly selected characters (at least one enemy per game) spawn together and remain visible until hit or until the timer ends.
- A scoring system rewards players for quick reactions, with higher scores for faster completion of the round.

3. How to play

When the game starts, 5 characters will spawn randomly on the screen, including at least one enemy.

- Use the cursor to aim and left-click the mouse or press the Spacebar to shoot.

- Hit all the enemies without exceeding the time limit to win the game. Your score increases based on how quickly you complete it.
- Avoid hitting allies - doing so three times ends the game in failure.
- The game also ends if you fail to hit all enemies before time runs out.
- Access the pause or reset options from the menu at any time.

4. Project Structure

4.1. Game Scenes

- **Main Menu:** Initial game scene with options to start or exit the game.
- **My Scene:** Main scene where gameplay takes place

4.2. Folder and files

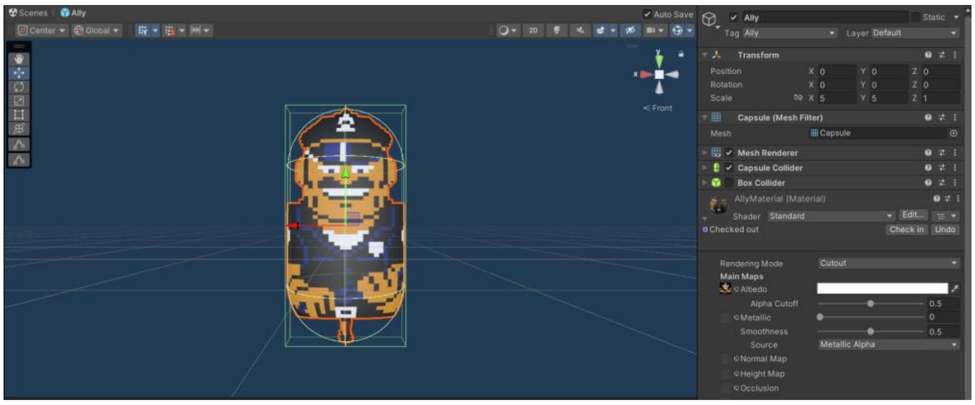
4.2.1. Assets Folder

a) Logs

- Contains a CSV file that logs game events, including NPC spawns, player actions, and scores.

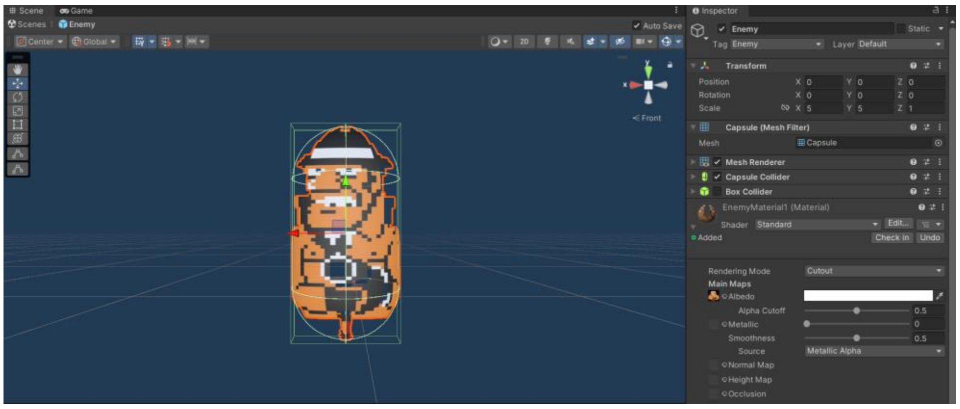
b) Prefabs

- **Ally:** Prefab for ally NPCs.
  - A capsule mesh with an ally texture material using the cutout option.

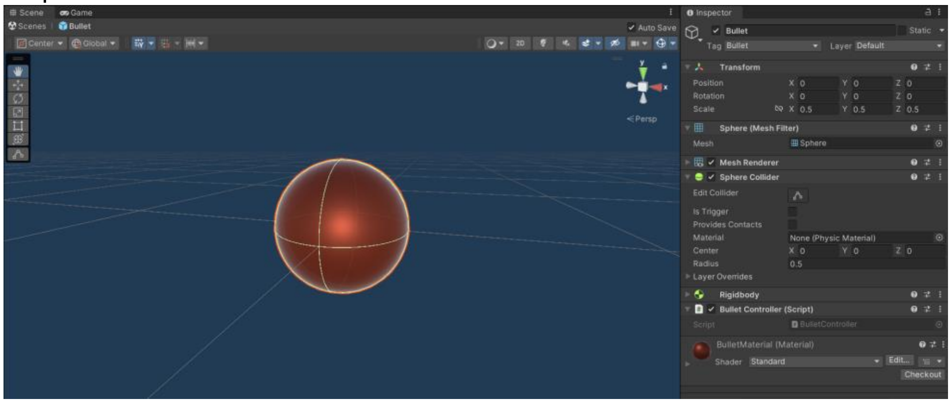


Ally
Tag Ally Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 5 Y 5 Z 1
Capsule (Mesh Filter)
Mesh Capsule
✓ Mesh Renderer
✓ Capsule Collider
Box Collider
AllyMaterial (Material)
Shader Standard
Checked out
Rendering Mode Cutout
Main Maps
Albedo
Alpha Cutoff 0.5
Metallic 0
Smoothness 0.5
Source Metallic Alpha
Normal Map
Height Map
Occlusion

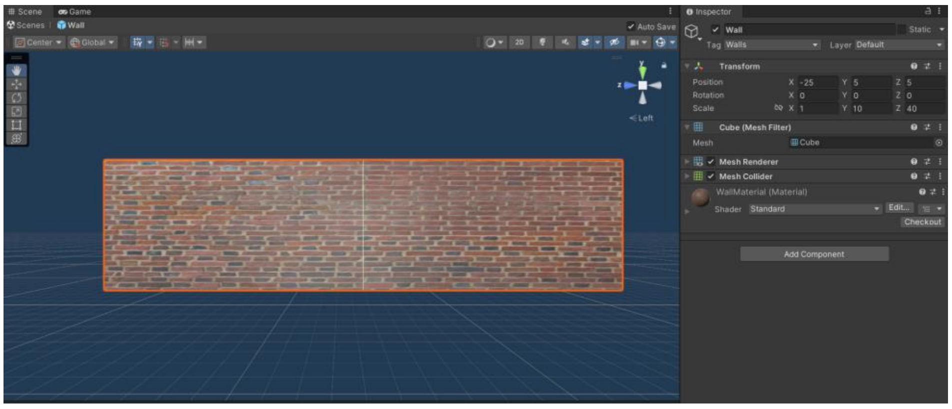
- **Enemy:** Prefab for enemy NPCs.
  - A capsule mesh with an enemy texture material using the cutout option.


```
Inspector
✓ Enemy                          Static
Tag Enemy            Layer Default
Transform
  Position   X 0    Y 0    Z 0
  Rotation   X 0    Y 0    Z 0
  Scale      X 5    Y 5    Z 1
Capsule (Mesh Filter)
  Mesh       Capsule
✓ Mesh Renderer
✓ Capsule Collider
  Box Collider
EnemyMaterial1 (Material)
  Shader     Standard
Added                              Check in   Undo
  Rendering Mode     Cutout
  Main Maps
    Albedo
    Alpha Cutoff                   0.5
    Metallic                       0
    Smoothness                     0.5
    Source           Metallic Alpha
    Normal Map
    Height Map
    Occlusion
```

- **Bullet:** Prefab for the bullets fired by the player.
  - A sphere mesh with a bronze metallic material


```
Inspector
✓ Bullet                         Static
Tag Bullet           Layer Default
Transform
  Position   X 0     Y 0     Z 0
  Rotation   X 0     Y 0     Z 0
  Scale      X 0.5   Y 0.5   Z 0.5
Sphere (Mesh Filter)
  Mesh       Sphere
✓ Mesh Renderer
✓ Sphere Collider
  Edit Collider
  Is Trigger
  Provides Contacts
  Material           None (Physic Material)
  Center     X 0     Y 0     Z 0
  Radius             0.5
  Layer Overrides
Rigidbody
✓ Bullet Controller (Script)
  Script             BulletController
BulletMaterial (Material)
  Shader     Standard                Checkout
```

- **Wall:** Prefab for walls around the field.
  - A cube mesh with a wall texture material.


```
Inspector
✓ Wall                           Static
Tag Walls            Layer Default
Transform
  Position   X -25   Y 5     Z 5
  Rotation   X 0     Y 0     Z 0
  Scale      X 1     Y 10    Z 40
Cube (Mesh Filter)
  Mesh       Cube
✓ Mesh Renderer
✓ Mesh Collider
WallMaterial (Material)
  Shader     Standard                Checkout
Add Component
```

c) Scenes

- **MainMenu:** The main menu of the game.
- **MyScene:** The gameplay scene containing all game objects.

d) Scripts

- Contains all the scripts for handling all the game events and updates.

e) Sprites

- Contains png files to be used as sprites in the game. These are:
  - Ally sprite
  - Enemy sprite
  - Wall texture
  - Gun sprite
  - Crosshair

4.2.2. Game Objects in MyScene

- **Canvas:** Contains UI elements (timer, score, pause menu, etc.).
- **Main Camera:** Configured with player perspective and rotation controls. Contains the script for handling player's shooting (bullet-objects collision)
- **Directional Light**: Provides lighting for the scene.
- **EventSystem**: Handles input events for UI and interactions.
- **Terrain**: Represents the ground plane of the gameplay area.
- **Walls (Left, Right, Back)**: Boundary walls with the wall texture material.
- **NPCController:** Controls the spawning of NPCs (enemies and allies).
- **LogicScript:** Manages the game logic, including win/lose conditions and event logging.

4.3. Interactions, Mechanics, and Developed Scripts

1. **LogicScript (Game Logic Management)**
    - Controls pause, reset, and game-ending scenarios, updating the "Game Over" screen.
    - Logs key events in the log file.
2. **MainMenu**
    - Manages transitions from the main menu to the game scene or exits.
3. **NPCController**
    - Spawns enemies and allies at random positions, logging each appearance.
    - Removes NPCs after a specified time.
4. **TimerController**
    - Controls the game timer and updates the timer display.
5. **BulletController**
    - Manages the behavior of fired bullets.
    - Calculates and updates the score based on the hit object (enemy or ally) and logs the results.
6. **GunController**
    - Controls the player’s weapon and manages the creation of projectiles (bullets).
    - Logs the number of shots fired.
7. **CameraController**
    - Manages the camera rotation based on mouse movement.