# CSE 3541 Lab Report 6

Austin Hendricks

The Ohio State University

CSE 3541: Computer Game and Animation Techniques

Dr. Naeem Shareef

24 April 2023

## Table of Contents

# Requirements and Import Instructions

## Requirements

- New Input System (please install from Unity Registry)
- Cinemachine (please install from Unity Registry)

## Import Instructions

1. Install requirements from Unity Registry
2. Allow restart of editor
3. Navigate to Edit → Project Settings → Player → Configuration and set the field “Active Input Handling” to “Input System Package (New)” and click Apply to restart the editor one more time
4. Import my project via Assets → Import Package
5. Once imported, navigate to Assets → Scenes and open scene titled “ZombieArena”
6. On the popup, click to import TMP essentials
7. Navigate to Game view and make the aspect 16:9
8. TMP essentials will not work properly until the game is played at least once. This is due to the nature of the TMP fonts. Please click Play to start the game, and immediately click Play again or hit the Escape key to stop the game
9. Now, the game is ready to play. Enjoy!

## Introduction

### Brief Description of Implementation

In this lab, I designed a first-person-shooter game called Zombie Arena where the player must fend off hoards of zombies using a variety of realistic weapons without dying. In this game, the player can move and look around, sprint, jump, operate weapons, switch between weapons, and collect pickups.

### Controls

- Move: **WASD**
- Look: **Mouse**
- Sprint: **Shift**
- Jump: **Space**
- Aim: **Right click**
- Shoot: **Left click**
- Reload: **R**
- Cycle Weapons: **Mouse Scroll Wheel**
- Switch to Rifle: **1**
- Switch to Pistol: **2**
- Switch to Rocket Launcher: **3**
- Exit Game: **Escape**

## Functionality

### Overview of Gameplay

The object of the game is to survive as many waves as possible in the arena. The arena itself consists of a 100m x 100m pit, which the player spawns in, and a spawn platform 12m above the surrounding walls of the pit, which is where the zombies spawn. Inside the pit are various obstacles for the player and the zombies to maneuver around. The gameplay of *Zombie Arena* is separated into waves, with each wave spawning more zombies than the previous. A wave is complete when all zombies from that wave are eliminated, at which point a sound is played and a countdown until the next wave begins. The player loses a life each time they are touched by a zombie, and the game ends when the player is out of lives. When the game is over, a Game Over pop-up appears displaying the total waves cleared and total zombies killed, with options to either quit the game or play again.

### Zombies

The zombies are operated via a script that is an extension of my Predator script from Lab 5. They are operated by a finite state machine, with states for wandering around, chasing the player, and avoiding obstacles. I added a health attribute to the script as well as methods for taking damage and dying so that the player can kill the zombies. I also increased the zombies’ field of view to 360 degrees and the view distance to 50 meters, so that all zombies except those far from the player will chase.

## Player / Camera

The player object contains a custom PlayerController script that interfaces with Unity's CharacterController, as well as a custom WeaponController script, in order to move the character around and operate weapons. The camera uses Cinemachine for a first-person camera that looks around based on mouse input delta. The player starts the game with 3 lives, and loses a life every time a zombie touches them. The game ends when the player is out of lives.

## Weapons

The player has three weapons they can use: a fully-automatic rifle, a semi-automatic handgun, and a rocket launcher. Each of these weapons are operated via the WeaponController attached to the player object. With each weapon, the player may shoot, aim, and reload. The handgun fires a 9mm round and ejects a 9mm casing, just like many real-life handguns of this style. Likewise, the rifle fires a 5.56mm round and ejects a 5.56mm casing, just like many real-life rifles of this style. Upon impact, these bullets leave bullet holes on anything they hit. The casings and bullet holes remain in the scene for a while and then automatically self-destruct after a certain amount of time. The rocket launcher fires an RPG-style rocket projectile which emits sound through the air and explodes upon impact with any surface or zombie, killing any zombies within 8m of the explosion. Each weapon has simple manually-implemented animations for recoil, reload, and aim. Additionally, each weapon has sounds for shooting, reloading, and dry firing. An ammo display at the bottom right of the screen is constantly updated.

## Pickups

There are two kinds of pickups that zombies randomly drop during gameplay: life pickups, and ammo pickups. Both types of pickups are animated with the same floating animation and collect animation scripts that I animated my coins with in Lab 3. Life pickups are floating 3D hearts that, upon collection, will give the player another life and update the health HUD at the top right of the screen. Life pickups are not collected if the player already has the maximum of 3 lives. Ammo pickups are floating 3D arrangements of bullets (5.56mm and 9mm) that, upon collection, will max out the ammo of the player's currently equipped weapon and update the ammo HUD at the bottom right of the screen. Ammo pickups are not collected if the player's currently equipped weapon already has maximum ammo. Whenever a zombie dies, there is a 5% chance it will drop an ammo pickup, and a 0.5% chance it will drop a life pickup.

## Optimization

I created my environment using hundreds of elements so that the Gridbox Prototype materials I downloaded would accurately correspond to the measurements it says on the material (i.e. 1 meter on the arena floor is equal to 1 meter in world space). In order to optimize my game, I created a custom MeshCombiner script and downloaded a Mesh Saver script so that I could combine all of the elements of my environment into just a few distinct meshes and save those meshes to my assets. This move brought my frame rate from around 70 fps to over 500 fps on my machine.

## Techniques Used and Authorship

### Techniques

I used the following techniques in my project, and personally authored my implementation of all of them:

- Finite State Machine (zombie behavior)
- Object Pooling (zombie spawning)
- Raycast (calculating where to shoot bullet towards)
- OverlapSphere (zombie vision, rocket explosion radius)
- Events (custom EventManager and Events written by me)

### Authorship

All of the code in my project is personally authored by me except for the following: MeshSaverEditor.cs and CollectableAnimation.cs. All parts of my project not authored by me are detailed below in the Materials From Outside Sources section on the next page. Everything in this lab not listed in that section is, in fact, authored solely by me.

## Material from Outside Sources

### Code

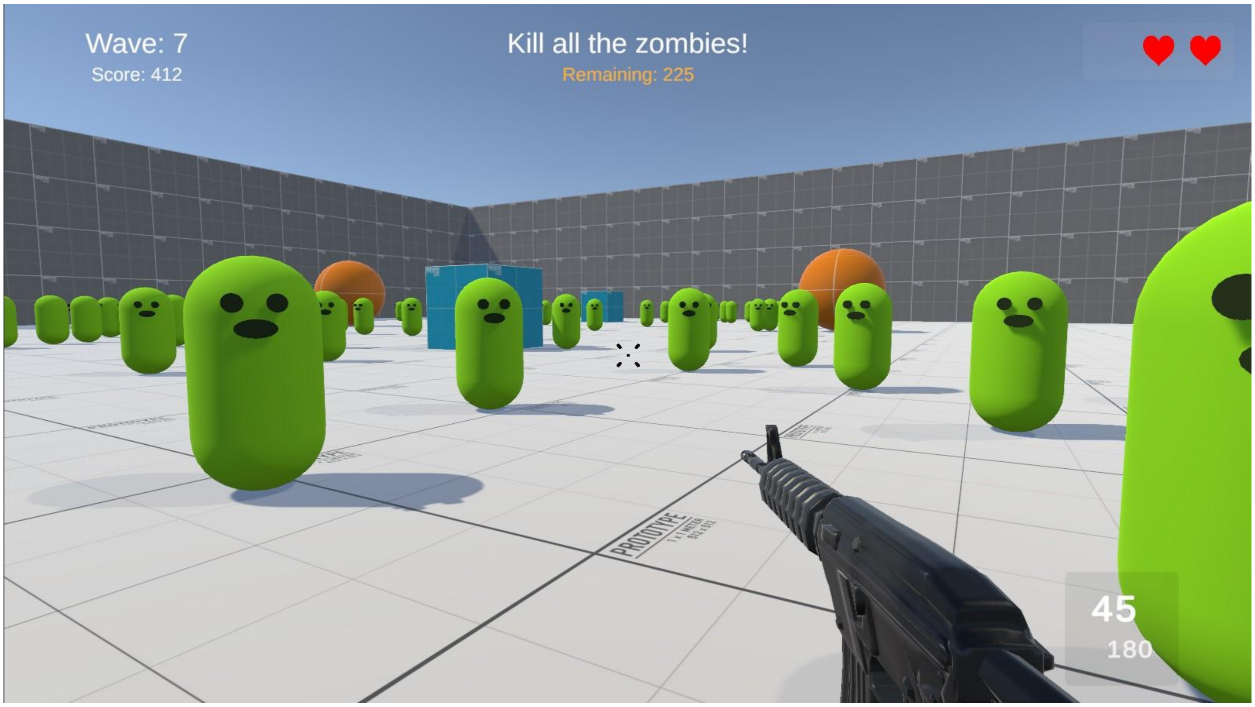
- MeshSaverEditor.cs – GitHub
- CollectableAnimation.cs – Simple Gems Ultimate Pack
- EventManager.cs – Myself, Lab 3
- Events.cs – Myself, Lab 3 (heavily edited for Lab 6)
- LoadSceneButton.cs – Myself, Lab 3 (heavily edited for Lab 6)
- MenuNavigation.cs – Myself, Lab 3
- QuitButton.cs – Myself, Lab 3
- QuitHandler.cs – Myself, Lab 1

### Meshes and Materials

- Floor and Wall materials – Gridbox Prototype Materials
- Pistol and Rifle meshes, materials, and sounds – Guns Pack: Low Poly Guns Collection
- Rocket Launcher and Rocket Projectile meshes and materials, sounds, and visual effects – Stylized Rocket Launcher Complete Kit with Visual Effects and Sound
- Muzzle Flash – Particle Pack
- Bullets (9mm and 5.56mm) – Ammunition Pack (Demo)
- Bullet Casings (9mm and 5.56mm) – VFX Bullet Casing
- 3D Low-Poly Heart (Life pickup) – Simple Gems Ultimate Pack
- All other sound effects – Pixabay

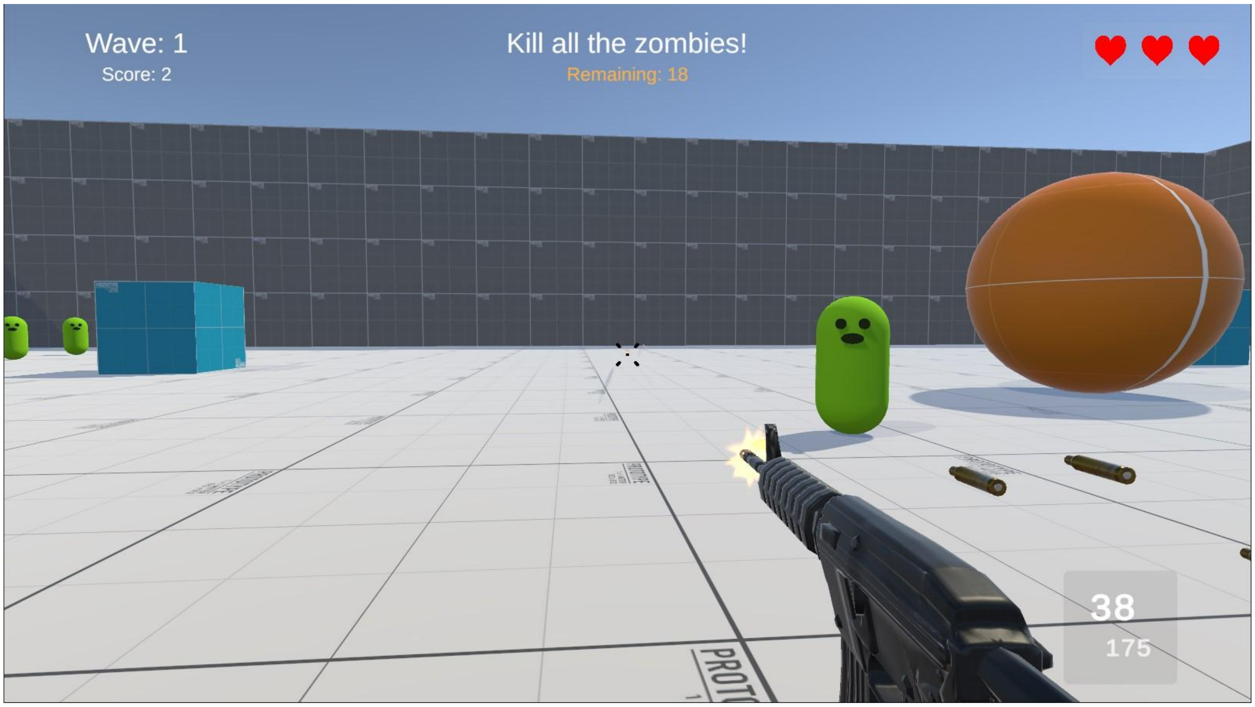
## Screenshots of Final Product

Zombie Hoard



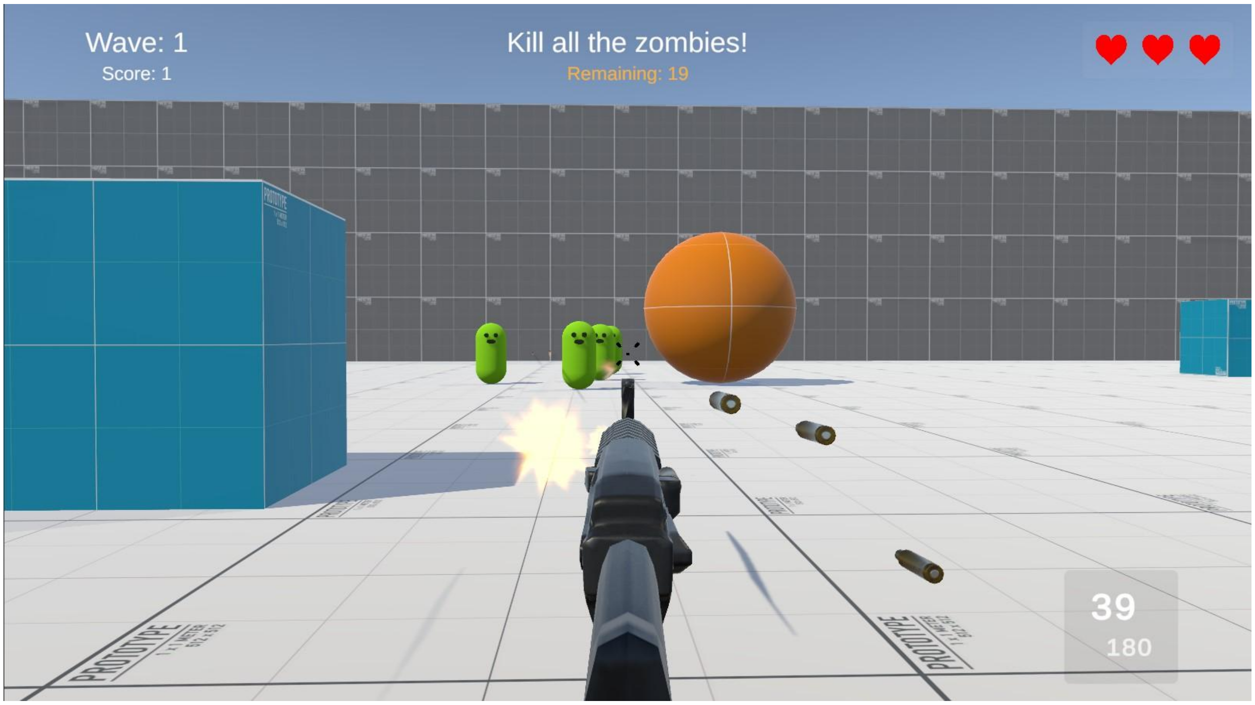
As you can see in this photo, there are several zombies (225 remaining) towards the start of Wave 7. I have two lives remaining and have 45 loaded rounds in my rifle with 180 more rounds in extra magazines.

Rifle Hip Fire



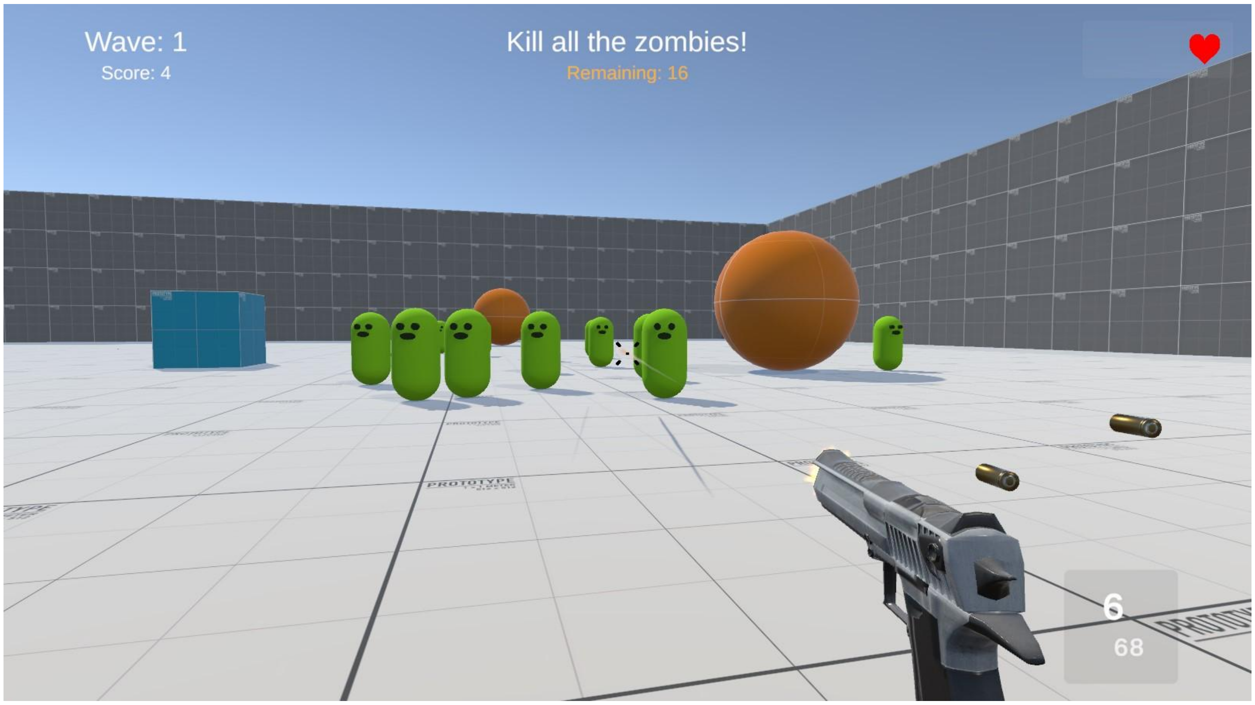
This photo displays the rifle firing. Notice the empty 5.56mm shells being ejected with each round shot, as well as the muzzle flash. These casings will land on the ground and stay there for a short time (around 30 seconds) before disappearing. The muzzle flash automatically disappears quickly after firing.

Rifle Aim



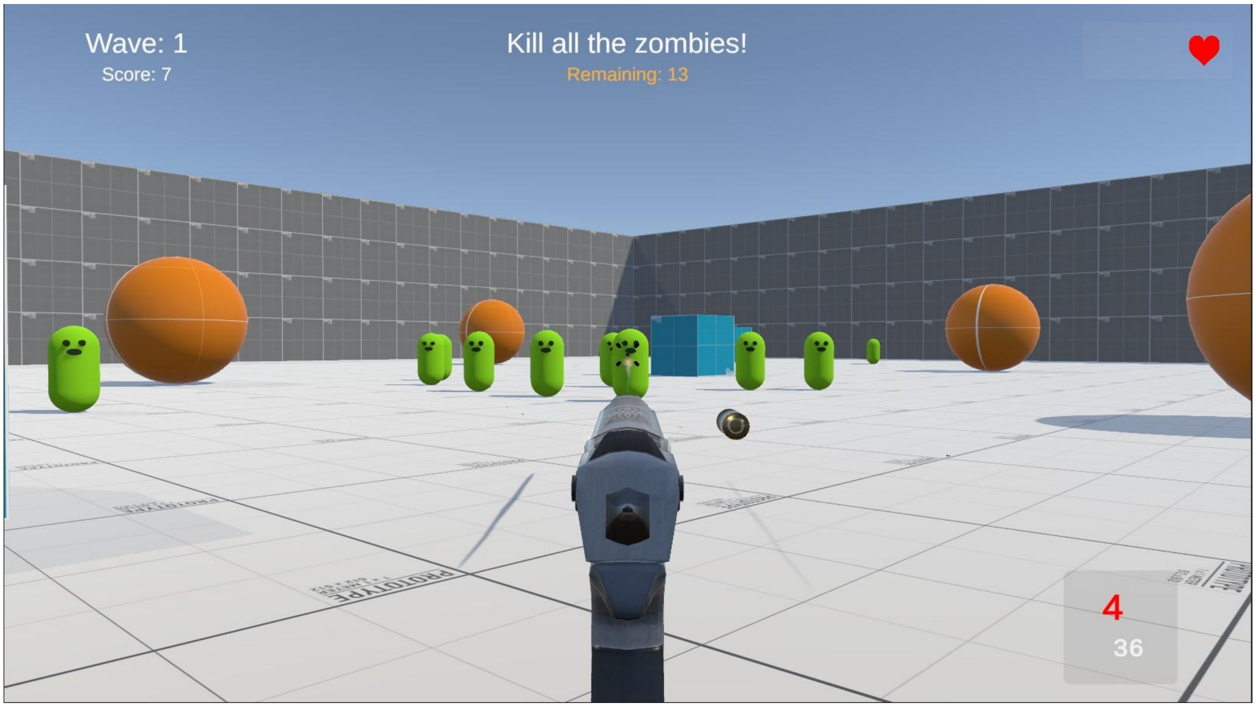
This photo depicts the rifle being shot while aiming at a hoard of zombies.

Pistol Hip Fire



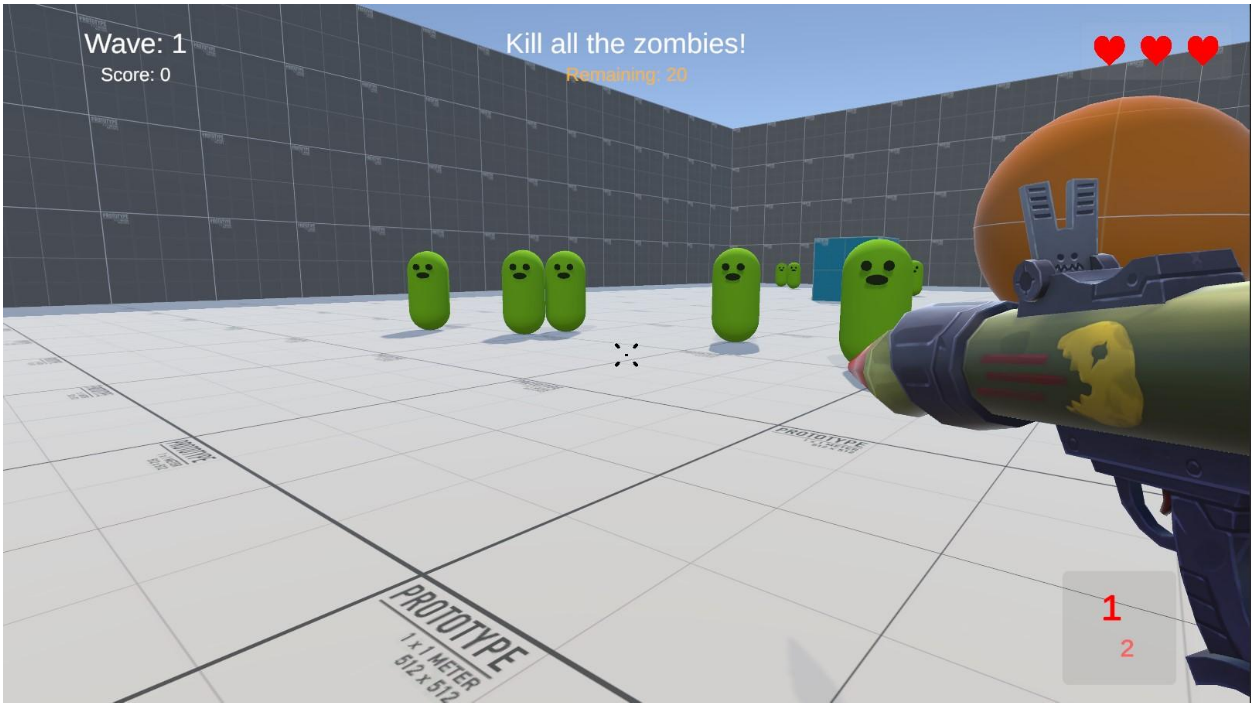
This photo shows the pistol being fired from the hip. Notice the bullet tracer traveling through the crosshair. Also notice the empty 9mm casings being ejected from the gun for each bullet shot, and note that these are different from the 5.56mm casings ejected by the rifle. The pistol also has a muzzle flash which, at the instant this screenshot was taken, was fading out.

Pistol Aim Fire



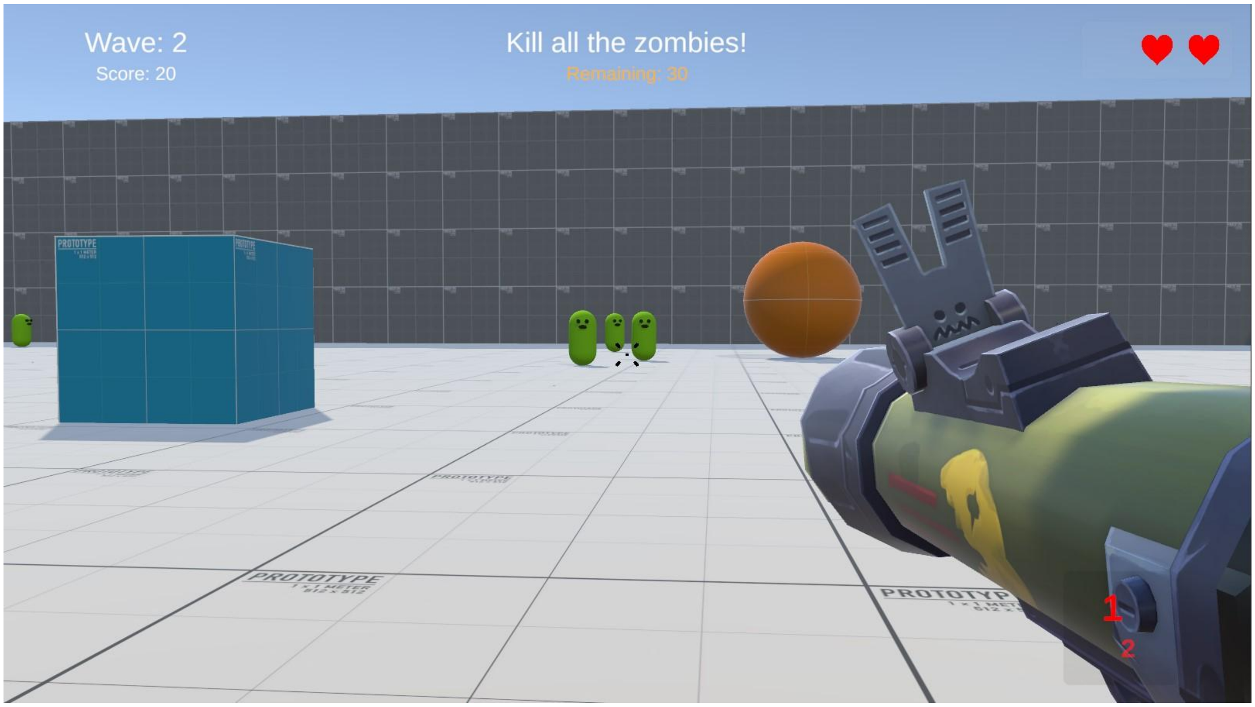
This photo shows the pistol being aimed and shot. Notice the tracer and bullet about to make impact with the zombie being aimed towards. Also note that in the ammo display on the bottom left, the large number (number of loaded bullets) is red. This is because this value is less than 5, so the red color warns the user that the gun is low on ammo. When the ammo reserve (bottom smaller number) is less than 5, it will turn red too.

Rocket Launcher Hip



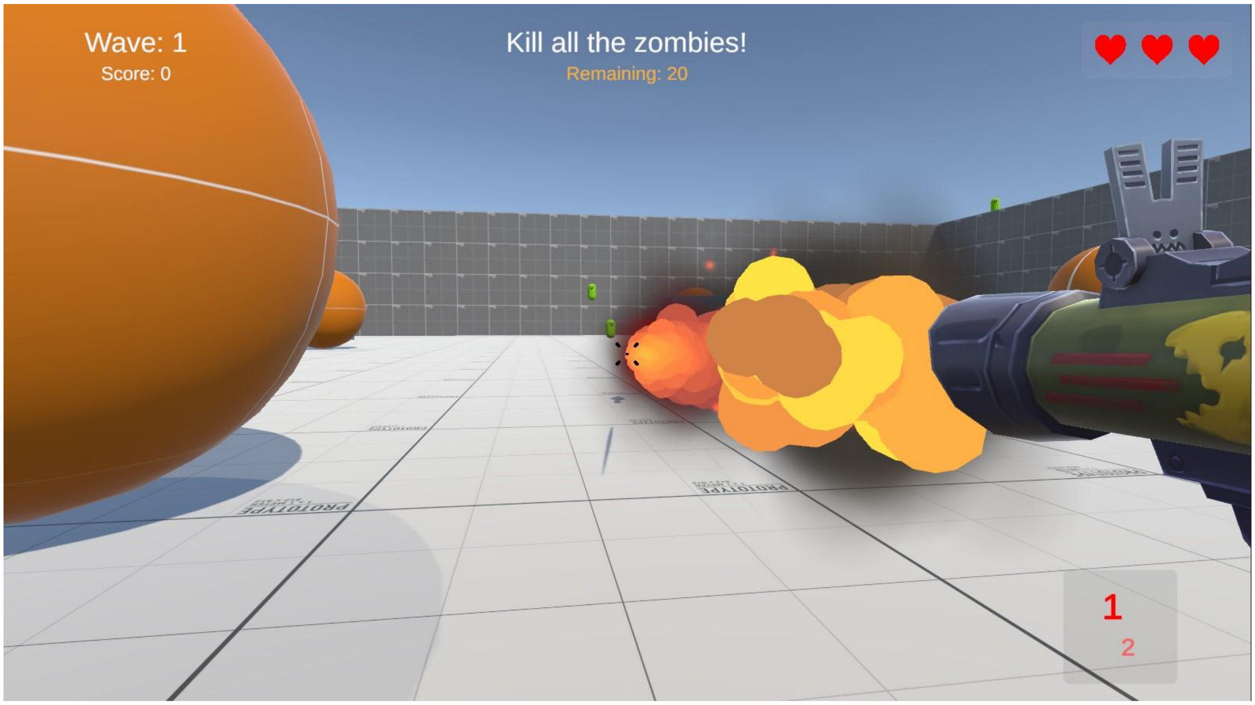
This photo depicts the rocket launcher as the equipped weapon with the projectile loaded in the muzzle, about to be shot towards a hoard of zombies.

Rocket Launcher Aim



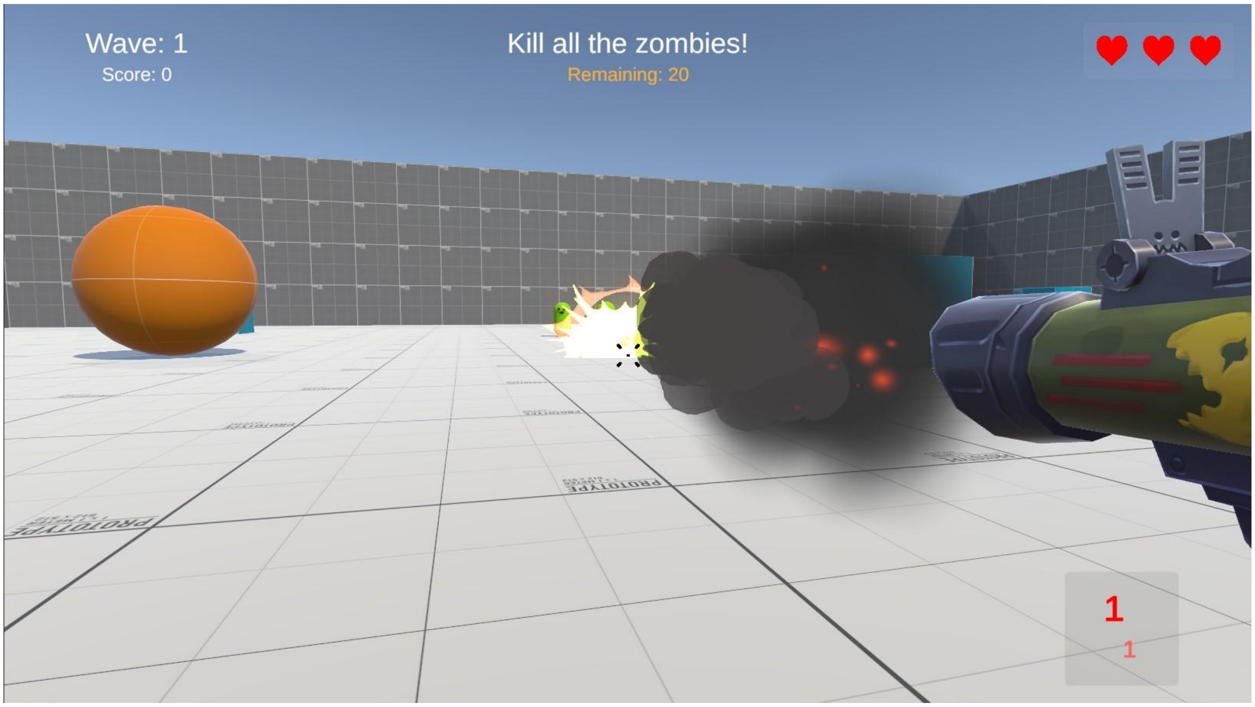
This photo depicts the rocket launcher being aimed towards a group of zombies, about to fire a rocket at them.

Rocket Launcher Fire



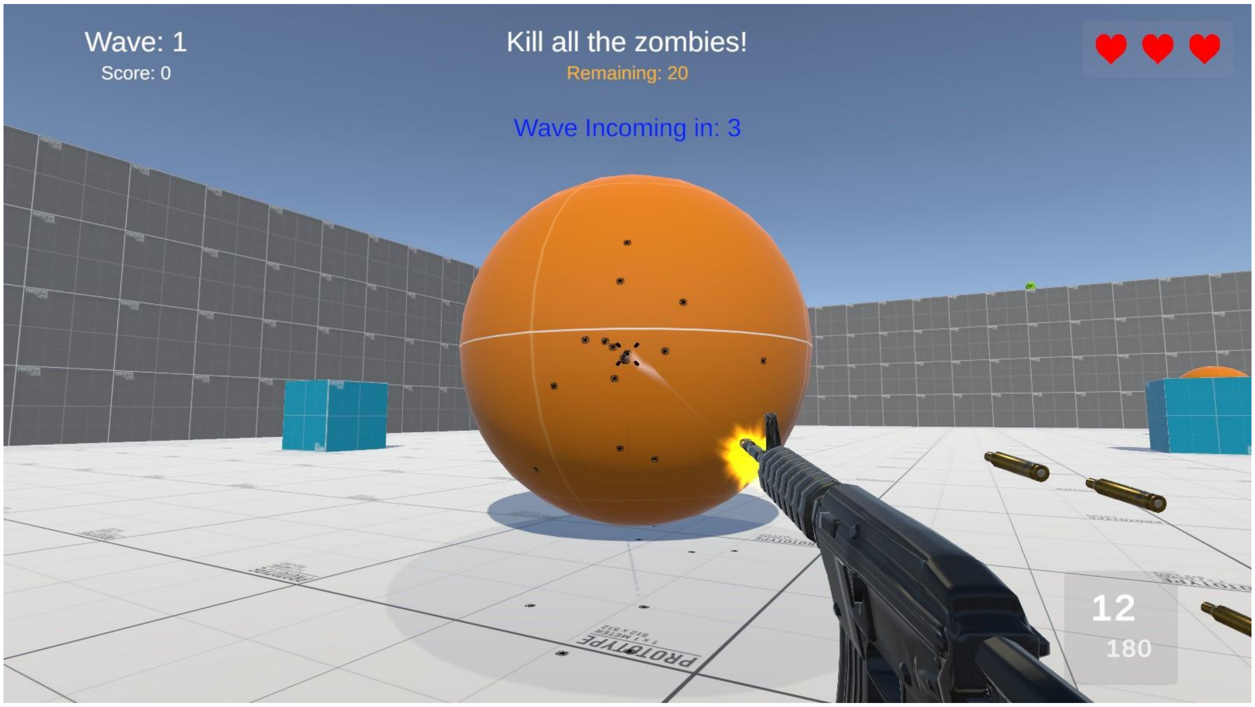
This photo shows the moment after the rocket launcher fires a projectile. Notice the flames that are shot out of the muzzle of the launcher. These flames will automatically disappear a moment after firing. Also notice the trail of the rocket projectile being the muzzle flames (darker orange color). The projectile is followed by this trail of fire and also makes a whirring noise as it flies by an audio listener. Upon impact with an object, the rocket will explode.

Rocket Launcher Impact Explosion



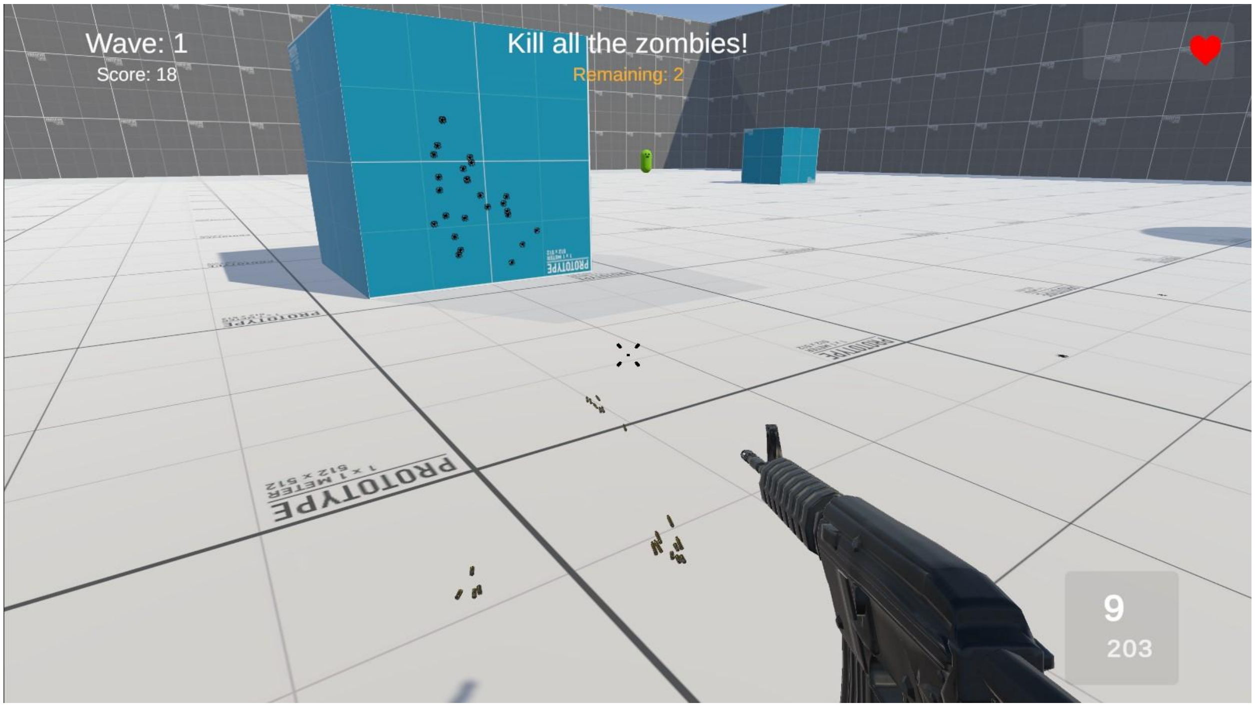
This photo depicts the smoke leaving the barrel of the rocket launcher right behind the muzzle flames, which have faded out at the moment of this screenshot. At the impact point, the rocket projectile is exploding and about to kill the zombies closeby behind it.

## Bullet Holes



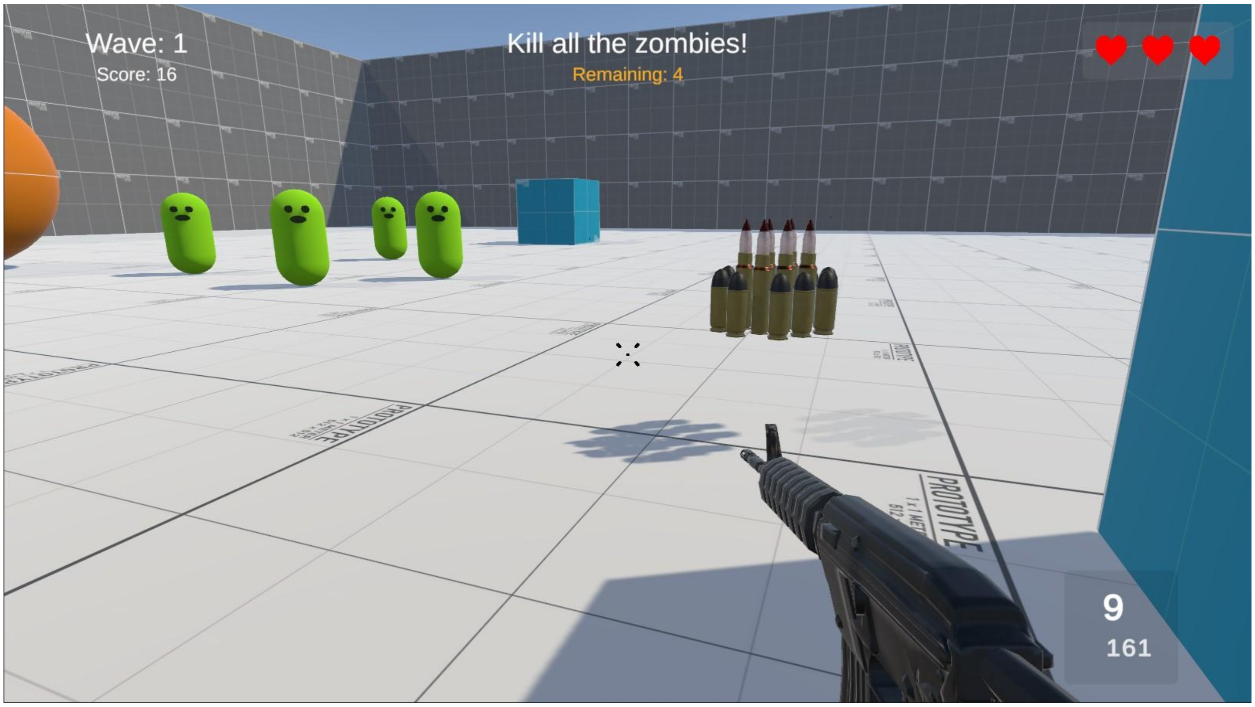
This photo shows the bullet holes created wherever a bullet strikes an object. Notice that this sphere has a curved surface, yet the bullet holes are still rendering with a proper orientation to the impact point. This is achieved by rotating the bullet hole's transform.forward vector to align with the surface normal of the point of impact.

Bullet Casings On Ground



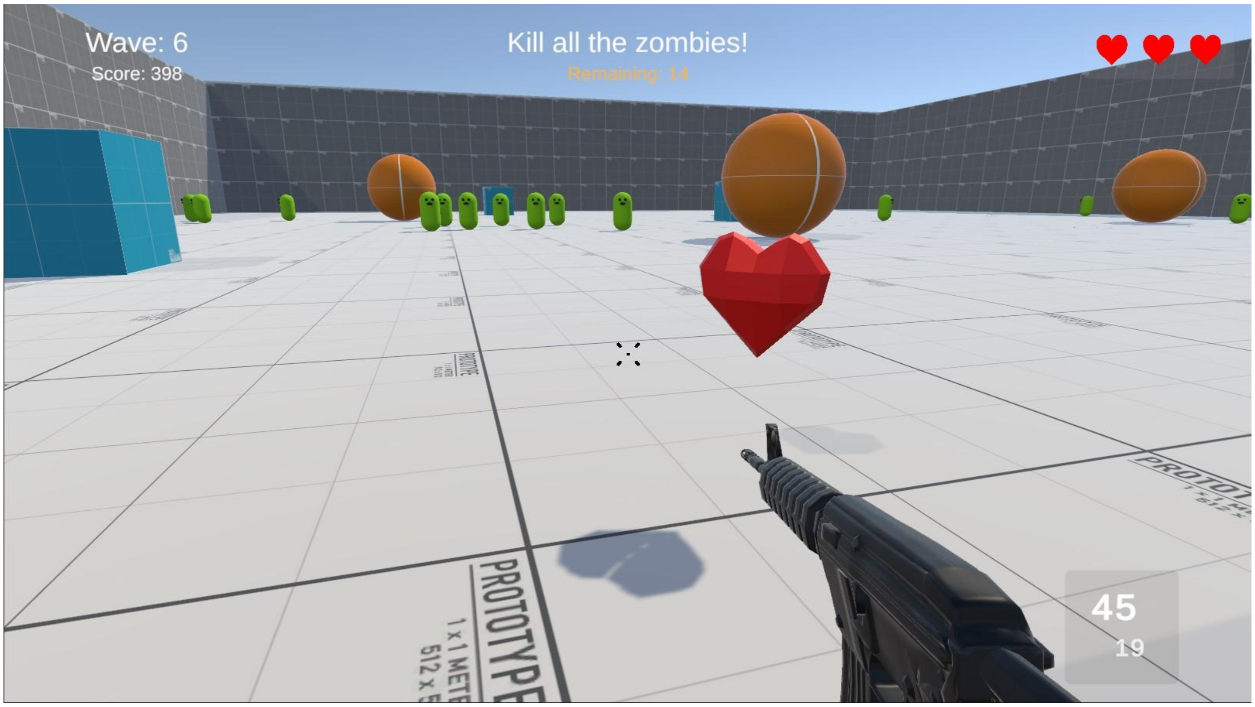
This photo shows some bullet casings lying on the ground. The casings on the right are 5.56mm casings from the rifle, and the casings on the left are 9mm casings from the pistol. These casings, as well as the bullet holes on the blue cube, will disappear automatically after 30 seconds of spawning.

## Ammo Pickup



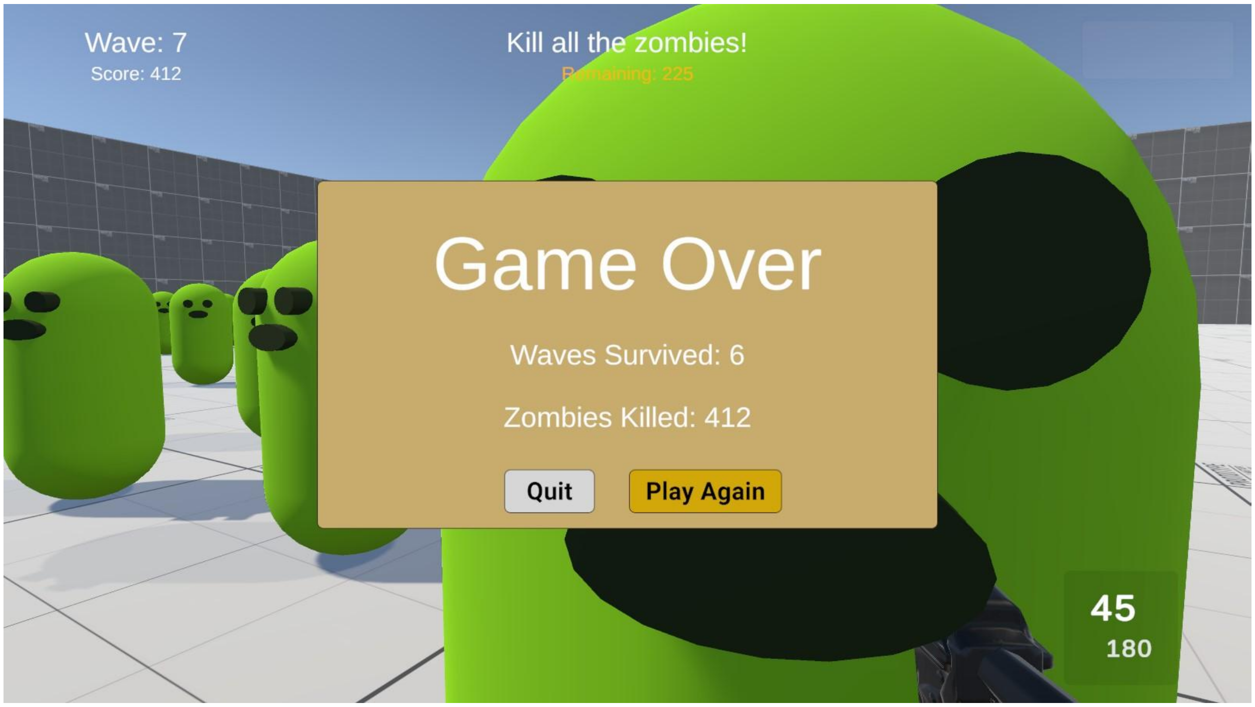
This photo shows an ammo pickup floating in front of the player. In real time, this pickup bobs up and down, scales smaller and larger, and rotates. Upon collection, the player’s rifle will max out its ammo since that is the currently equipped weapon. Also upon collection, a coroutine causes the pickup to play a noise and float upwards for a brief time before disappearing.

## Life Pickup



This photo shows a life pickup floating in front of the player. Just like the ammo pickup, this life pickup is animated and will behave the same way when collected. The pickup cannot be collected at the moment captured in the screenshot, since the player already has all three lives as seen in the top right corner of the screen. However, if the player loses a life, they may then collect this life pickup which will gain the player one life back.

## Game Over



Wave: 7
Score: 412

Kill all the zombies!
Remaining: 225

Game Over

Waves Survived: 6

Zombies Killed: 412

This photo depicts the game over screen. The waves survived as well as the zombies killed totals are shown. From here, the user may select to Play Again, which reloads the scene and begins a new game, or Quit, which quits the game. This menu is traversable via WASD keys, arrow keys, and mouse point. An option may be selected via clicking it or selecting it with WASD or arrow keys and pressing enter. In the screenshot, the Play Again option is preselected which is why it is yellow. Play again is automatically preselected every time the game over menu is activated. Thanks for playing!