TEAM ZOMBIE: UNREAL ENGINE 4 ZOMBIE SURVIVAL

Blake Lapum & Ashley Hendrickon

Grand Valley State University – CIS 467 Capstone – Winter 2020

Overview

Goal:
The main goal of the project was to create a FPS (first person shooter) zombie survival game through the Unreal Engine 4, via the Unreal Editor. Because we are both brand new to the engine, the first steps and most basic goals for "Zombie Survival" were to create an extensive landscape, be able to walk around, and kill zombies.

As the project came together and more advanced features were feasible, our advanced goals were to implement a health system for both the player and the zombies, an ammo system, and smoother zombie AI animations and logic.

Figure 1: Unreal Engine logo, Copyright Unreal Engine 2020

Landscape

Problem
Most modern games have extensive maps with realistic landscape and architecture. There are trees where there should be trees, and there are chairs where there should be chairs.

Our map aims to keep a realistic feel, while also keeping the uneasy, dark vibe of a zombie game.

Solution
- Imported asset packs
  - Warehouses, nature, and miscellaneous foliage
- Adjusted lighting to make the game have a dusk-type feel while still being able to see and navigate the landscape
- Created river border (with physics)
- Surrounded map edges with mountains and plugged any escapable spots so nothing falls off the map
- Added rain falling (rendered only around player character)
- Added starry sky and moon
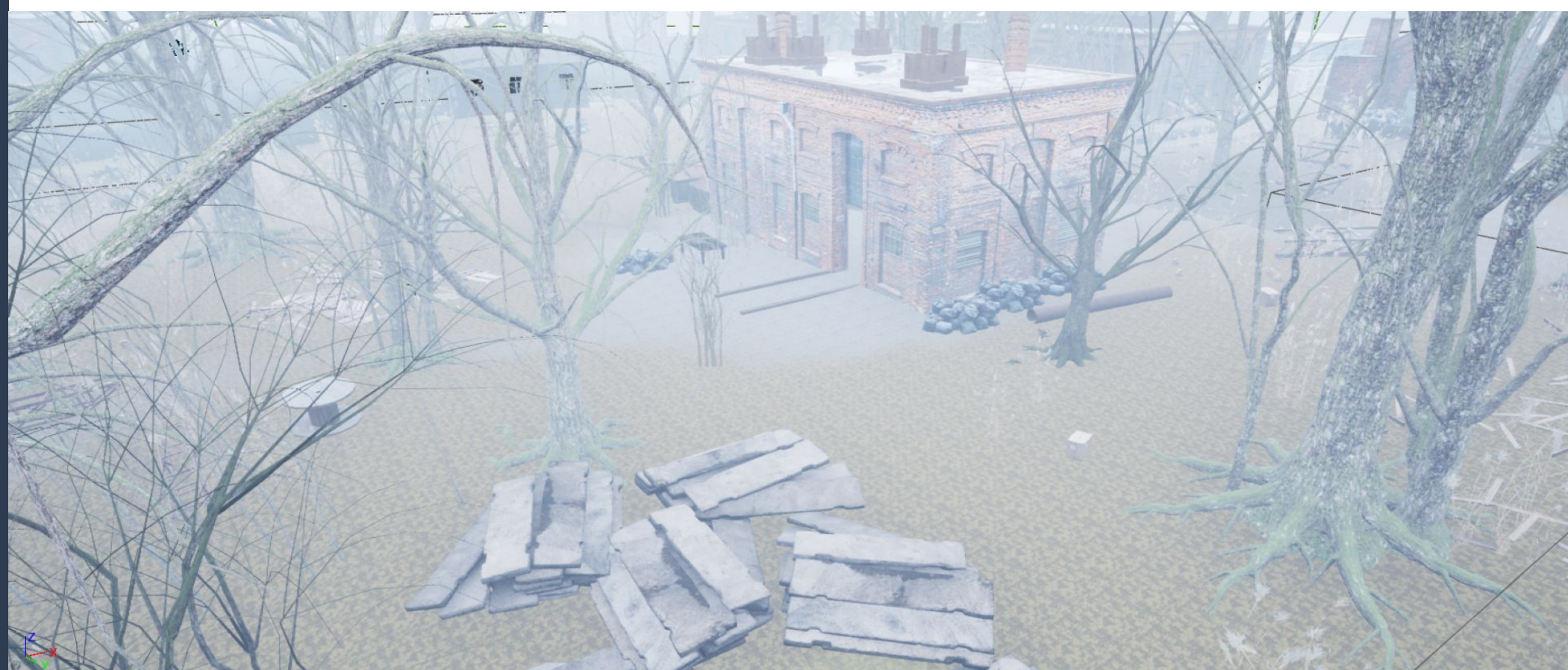- Added navigation mesh so player & zombies can only roam intended area

Figure 2: Map showcase

Player Character

Problem:
The Unreal Engine has a project template for a first person shooter, so beginning functionality is achieved out of the box. When you first open the project, the player can move in any logical direction (using the arrow keys), jump (spacebar), and shoot a "ball gun" (left-click).

While the base functionality supplied was a good start, the key bindings were not ideal, there was more functionality to be achieved (sprinting, aiming down sight, melee, etcetera), and there were little to no animations.

Aside from the look and feel of the base character, there was no initial logic for a health, ammo, or damage system.

Figure 3: Screenshot of base Player Character and default AI system

Solution:
Look & Feel
- Remapped the keys for moving to: A (left), S (backwards), D (right), and W (up), while spacebar remained bound to jumping
- FPS Assault Pack happened to be free for a month and with it came:
  - Weapons and animations (reload, melee, sprint)
  - A new character player with more advanced logic

Health
- Implemented a starting health of 100.0 and displayed the health bar and percent on the screen
- Each time the player gets hit, they lose 5.0 health
- The player can have a maximum health of 100.0 and the game ends when the players' health reaches 0.0
- Medical packs randomly spawn throughout the map and can be walked over for a health increase of 25.0

Ammo
- Player starts the game and immediately walks over an ammo pack, which increases ammo supply by 25 (one magazine) when you walk over it
- A player can carry a maximum of five magazines at a time (each has a capacity of 25 rounds)
- Each time a reload occurs, the whole magazine is dispensed and the player loses any extra rounds that remained in the magazine

Damage
- Head shots deal 10.0 damage
- Body shots deal 5.0 Damage
- Melee deals 3.0 damage

AI & Spawn System, Zombies

Problem:
The Unreal Engine does not have a template for zombie characters or the AI said characters would use to know the player character is nearby and should attack. The Unreal Engine also does not have a template for spawning random objects (zombies) throughout the map.

Solution:
Zombie Character
- Imported free zombie characters and animations from mixamo
- Implemented and smoothed zombie animations via the Blender editor within the Unreal Editor
- Implemented health system; zombie maximum health is 15.0

Figure 4: Zombie Closeup Mid-Animation

AI & Spawn System
Free for a month was the Open World AI Spawn System, with it came:
- Basic AI with built-in zombie functionality (attack the player)
- AI Spawning (random or static)
- Added functionality for zombies to deal damage
- Extended spawn system to respawn zombies when the count got low enough

Figure 5: Final Implementation of Zombies and Player Character

End Result

We are Happy About:
- Learning a new technology and gaining an understanding of how game development works
- Learning the Blueprint "coding" system
- Adding complex logic to existing systems and Blueprints for customized behavior
- Creating Blueprint classes from the ground up with responsive functionality
- Building a landscape from the ground up and importing necessary assets to create the creepy, dark feel of the game
- Planning a game with a goal in mind and reaching that goal

We are generally happy with how the game turned out. The landscape has the creepy look and feel we were going for, the player character has an array of functionality and animations, and the zombie AI look like zombies and attack as zombies should.
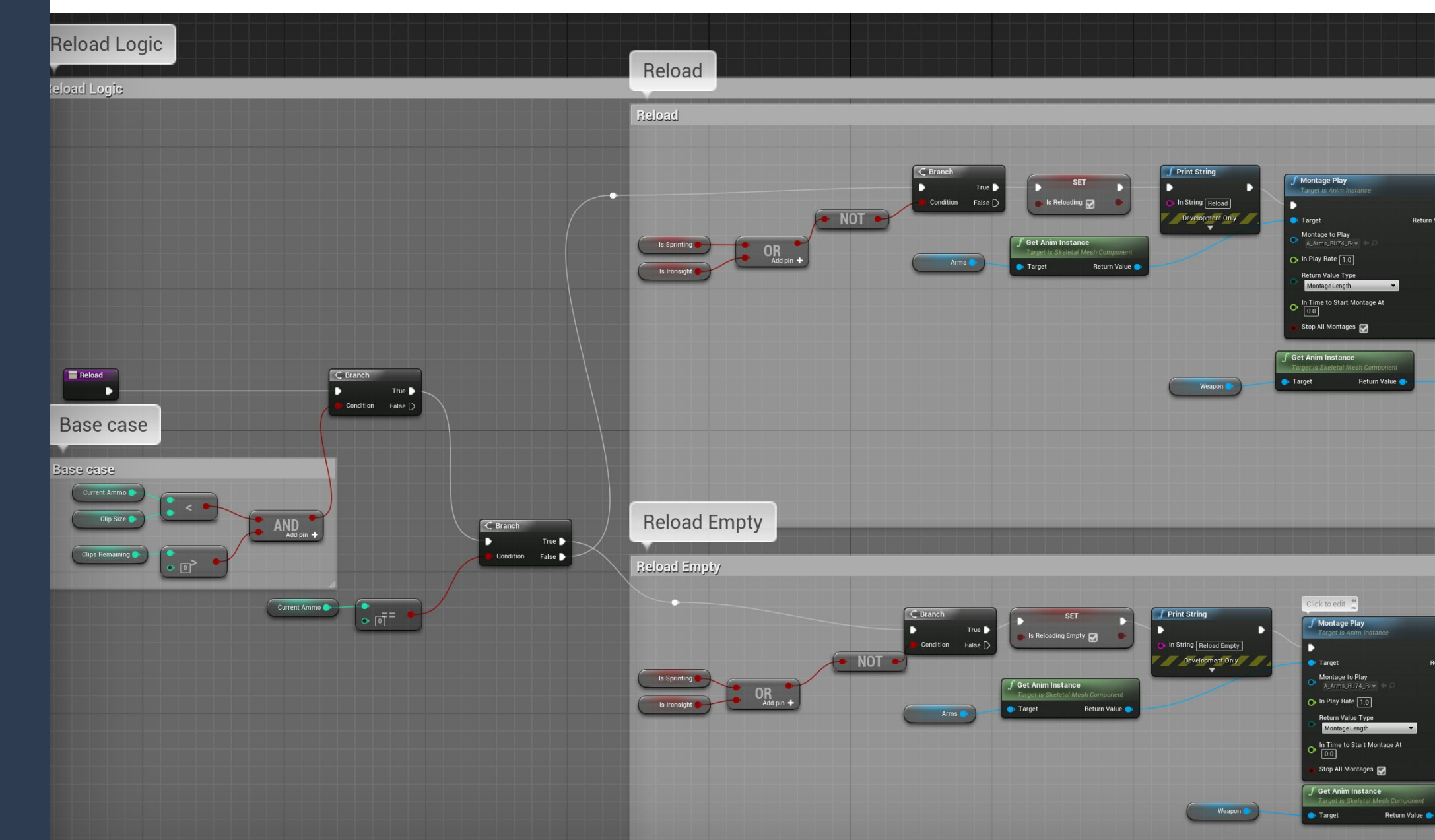

Figure 6: Blueprint Code example encapsulating reloading logic

What Comes Next
- Optimize performance for PC's with lesser specs
- Create a backend database to store user scores
- Add cinematic scenes to build the storyline
- Different difficulties
- Utilize more weapons

Acknowledgements

Support
The creator support from both the FPS assault pack and AI Spawn system were unmatched. The discord channels are both community and creator supported, which made questions dissolve quickly.

Tutorials
The tutorials by Unreal Engine experts and fans on YouTube, and other various websites, made implementing complex features as simple as could be.