Incremental Movement Shooter
Project Reflections

Thamsen Borges

December 15, 2023

Part I

Overview

0.1 Big Eyes...

From the Player's Mind The opportunity to make a game in the image of my favorite titles was too sweet to ignore. While easier options were on the table for making a game in the time required, I picked something that, over my time in similar games made by large studios, I wanted to see made and be able to play.

0.2 ...Small Stomach

To the Player's Tiny Hands However, my lack of experience in the required software stack led to many hurdles and pitfalls that I could have avoided if I knew what I was getting into. The ideas are simplistic and well-scoped, but the lack of similar work in the field made putting together a project much more difficult than originally assumed.

Part II

How It’s Going

Chapter 1

Project Goals

1.1 World

Visuals A progressively dingier, more corrupted cityscape for the player to explore, with movement full of weight and consistent feedback.

Audio With each level, the intention was to ramp the player up another layer into final frenzy.

1.2 Abilities

Incremental Movement and Power The player, starting with a pistol and a standard walk, obtains a grappling hook to soar through the same level in reverse, and then obtains a weapon to physically destroy the resulting corruption spreading through the city.

1.3 Enemies

Corrupted Pawns The primary threat of the first level.

Corrupting Tendrils The second level's intended threat, which begins to shower the city in pawns while enclosing the player in fleshy chambers.

The Source At the end of the third level, a boss was planned to let the player finish the fight.

Chapter 2

Project Status

2.1 Completed Segments

Level Layout The general idea for all three levels is laid out in the submitted executable.

First Person Character The camera implementation was finally finished when switching to Unreal.

Separate Character Blueprints Each level was modified so that the player could begin each section without worrying about pickups; animations were planned to indicate to the player that a new ability was earned.

2.2 In-Progress Elements

Textures Left for a later date, the polish on the models was not yet developed.

Background World To keep the player focused, the main map is small; finding a good method to contain that player while not breaking immersion was unfruitful, so it was put on the backburner.

Mesh Animations in Levels Two and Three The aspect of a changing world was advanced enough that the original materials reviewed for the engines didn't reveal an easy path forward for these.

Antibody Gun Since mesh animations were a blocking item, this entity that could interact with them was held back.

Enemies Originally, Lyra would have provided this, but Lyra was dumped as back-porting First Person to it was too effortful.

Part III

How It Went

Chapter 3

Early Days

An exciting project With titles like *Titanfall 2* and *Doom (2016)* among my favorite games, I was excited to try and implement mechanics that felt like this, as those simple components seemed to make a smooth and fun experience an afterthought. Project planning was simple, and the game design process was very organic and creative.

Chapter 4

Engine Frustrations

4.1 Unity's Allure

Prior Bias In my time online, I've been part of groups that had previously used Unity in their 3D work and as a general development engine, and it seemed very mature for putting together simple games that worked well. With many VR games I'd played being made in Unity, I had a thought that a first person shooter would be almost equivalent to a "weekend project," and my expansions on that would be quick and easy.

Beginner's Luck Putting together the first couple levels (in their whiteboxed form, at least) was relatively simple after only a couple days of learning, and it seemed that the items I had researched (mostly the grappling hook) were going to be easy to implement from simple tutorials available on YouTube.

4.2 Betrayal

The Devil: Details Unfortunately, when trying to integrate the knowledge from so many different sources and disparate places, Unity developers reliably run into problems with how many separate systems and practices are used across the engine, and end up writing much of their functionality themselves, constantly reimplementing basic systems with little abstractions.

First Person Cameras While there are many of these already available in one way or another, Unity's open source and friendly development community rarely makes these types of games. Any first person games made in Unity were instead closed-source and generally secretive about their practices.

Minimal Documentation While Unity's featureset is documented in principle, the actual usage and integration of these features is very sparse, and despite large public interest, knowledge is spread between many sources and almost entirely unsearchable, since many of the experts only publish YouTube video tutorials to describe projects. Even developing the more complicated meshes for the later levels took more along a week to find the specific features in Unity that I was looking for, and I had to abandon that to get what I wanted by exporting level two to Blender, then re-importing it after modifying the mesh in the more ergonomic editor.

Despair After looking at project deadlines, I realized it was untenable to continue in Unity. I started looking for other engines, including Godot and Unreal, and then I realized the choice was pretty obvious.

4.3 A New Hope

A Small (and Unfruitful) Detour In order to try and avoid wasting further time, I watched a few videos about development in Unreal, and was originally impressed and sucked in by the Lyra Starter Game, which is a third-person-first shooter game that intends to be a jumping-off point for developers making multiplayer, networked games for the future. I erroneously thought that its modularity meant that I could strip out what I needed and make a minimalist game on top of it, but then ran into hiccups when the few people trying to do what I did had given up and instead worked from a better angle: Building their simple games from simple parts using the OTHER tools Unreal already has. I essentially had to resolve to stop trying to save time by overplanning and get into the engine to figure out the game plan.

The FPS Engine After biting the bullet, registering, and downloading Unreal Engine, I opened it and was immediately presented with project templates that aligned exactly with my intentions: Epic Games had already packaged a "beginner FPS" to bound off of, and constantly publishes assets and examples, tutorials and videos, all from the first person source, and the online communities for making games in Unreal are Unparalleled in friendly support and healthy knowledge. Every question I had was much easier to answer in this context.

Migration With all of this new knowledge, porting what little I had over into Unreal to get a working demo for user testing was actually incredibly simple, and learning what I needed to know for further development was actually rather simple.

Chapter 5

More Complaining

5.1 Unreal's Not Perfect

Industry (lack of) Standards However, I learned that development of games remains hard and varied, with many pieces of software still used for almost every aspect of game development. My lack of prior experience in all of these pieces of software (Blender, Unreal, Unity, Substance Painter, ZBrush) that are commonly used by studios IN TANDEM, which all have very similar and overlapping feature sets with different interfaces, meant that learning the same utilities in multiple tools took up a majority of the development time, and added to frustrations about a lack of general industry pedagogy online.

Learning is Still Fun Regardless of the time crunching and various holdovers, each time I learned how a new detail of game development was done, I gained a little more appreciation for how much work has been done (no matter how much still needs to be done), and how much great effort people put into making their ideas come to life in these engines. Constantly seeing the various projects people are working on in all of these game development communities was refreshing and inspiring every time.

Part IV

Now What?

Chapter 6

Further Plans

Continue Development I primarily plan to simply continue on the path, next with in-level mesh animations, so that I can get the chase of level two working, and keep trucking along the path I traced out in my development plan.

Yet Another Engine? I think it would also be interesting to try out this process in an engine like Godot, just to get a feel for what that system's current strengths and limitations are as both an engine and a supporting community.

Chapter 7

Thank You

This project allowed me to express, hands-on, ideas I've had about video games for a very long time, and I already have a better perspective of this thing that takes up a significant part of my life. I didn't mention any difficulties in generation of narrative or focus on process generics because all of those were well-covered in the course material, and that was what was necessary to get all of this planned and designed in the first place. Thanks again for letting me and encouraging me to make this game.