UNITY MINOR BOSS SCREENCAST PLANNING

#Note

Introduction:

Introduce myself

Introduce my minor boss and chosen learn project

Walk through the different steps of the project

Explain why I didn’t do multiplayer – Unity phasing out old system and developing new system as I say this.

Explain that what I saw that this project really needs was POWERUPS

Walkthrough of the powerup system – Scriptable objects and EXPANDABILITY

Explain Main Menu and options menu – including playerprefs and audiomixing and toggling pickups

TIME TO TALK PERFORMANCE :D

One major thing that people said while playtesting this game was that ‘Lag spikes’ and input latency was a problem that impaired the performance of the game significantly. So I set out to fix this the best I could.

As I’m sure you know, because c# does not have the pointers system that C++ has to explicitly state if an object is a reference, the compiler needs to figure that out for itself. Which results in this thing called GARBAGE COLLECTION :D

Essentially it’s a bad thing (like memory leaks in c++) and what you need to do to avoid it is to never EVER use the “new” statement when assigning to any local variable in a method that is called frequently (Update and FixedUpdate) which solves most of the problem.

Before I made this screencast I went on a bit of a crusade to eliminate any garbage collection. But that still didn’t solve the problem entirely. Which I figured out had something to do with the rapid creation and destruction of the player’s shells and shell explosions alongside other objects.

WHICH BROUGHT ME TO THE DOORSTEP OF BRACKEYS TO LEARN THE BASICS OF OBJECT POOLING

#save the memory turtles

After implementing this I decided that I was just about done but the last balancing issue I decided to fix was that there was nothing stopping the players from mashing the fire button and using it as a short range assault rifle, essentially I needed to give it a cooldown