Learning and Methods

My work in Unity has been primarily focused on object scripting and properties, and I've learned a number of things from looking up specific questions and properties online. I've focused more on code samples and documentation than full tutorials.

My first concern was figuring out how Unity handles assets and their associated rigidbodies. Notably, the distinction between the transform and rigidbody of an object, and the physics connotations of each, was interesting and important for implementing enemies that move believably and respond to collisions.

In a similar vein, movement options were important to understand. I initially tried adding forces, but found response time to be too slow; setting velocity directly was a far better solution. Of course, this also required understanding how Unity handles vectors and coordinates, and how those worldspace coordinates lined up with the dimensions of the screen given our current camera position.

Naturally, all of this also required understanding how Unity handles scripting and attaching scripts to objects. It also required understanding the Start() and Update() routines, as well as object variables and how to access them.

I also learned how Unity handles prefabs, and how to create and modify new prefabs with rigidbodies, scripts, and variables already created and attached. This enabled me to then create and adjust new objects with scripting, which was a vital part of spawning enemies throughout the level. I also learned that Unity doesn't handle freefloating scripts well, and thus the necessity of attaching scripts to a dummy object when global actions are required.

Further refinements to this behavior also necessitated understanding coroutines and how Unity handles timing. Adding organic behavior also required looking up usage of the Random function, and the Destroy function was necessary to ensure created objects did not accumulate indefinitely.

Design and Contributions

Our team has been in agreement regarding most things, so it'd difficult to pin down specific decisions or contributions I've been entirely responsible for. I was involved with the basic premise of a spaceship shooter game and many of the core concepts, but so was everyone else.

I've probably been more active in deciding enemy qualities than anyone else, but that's primarily because it's been my main area of expertise.

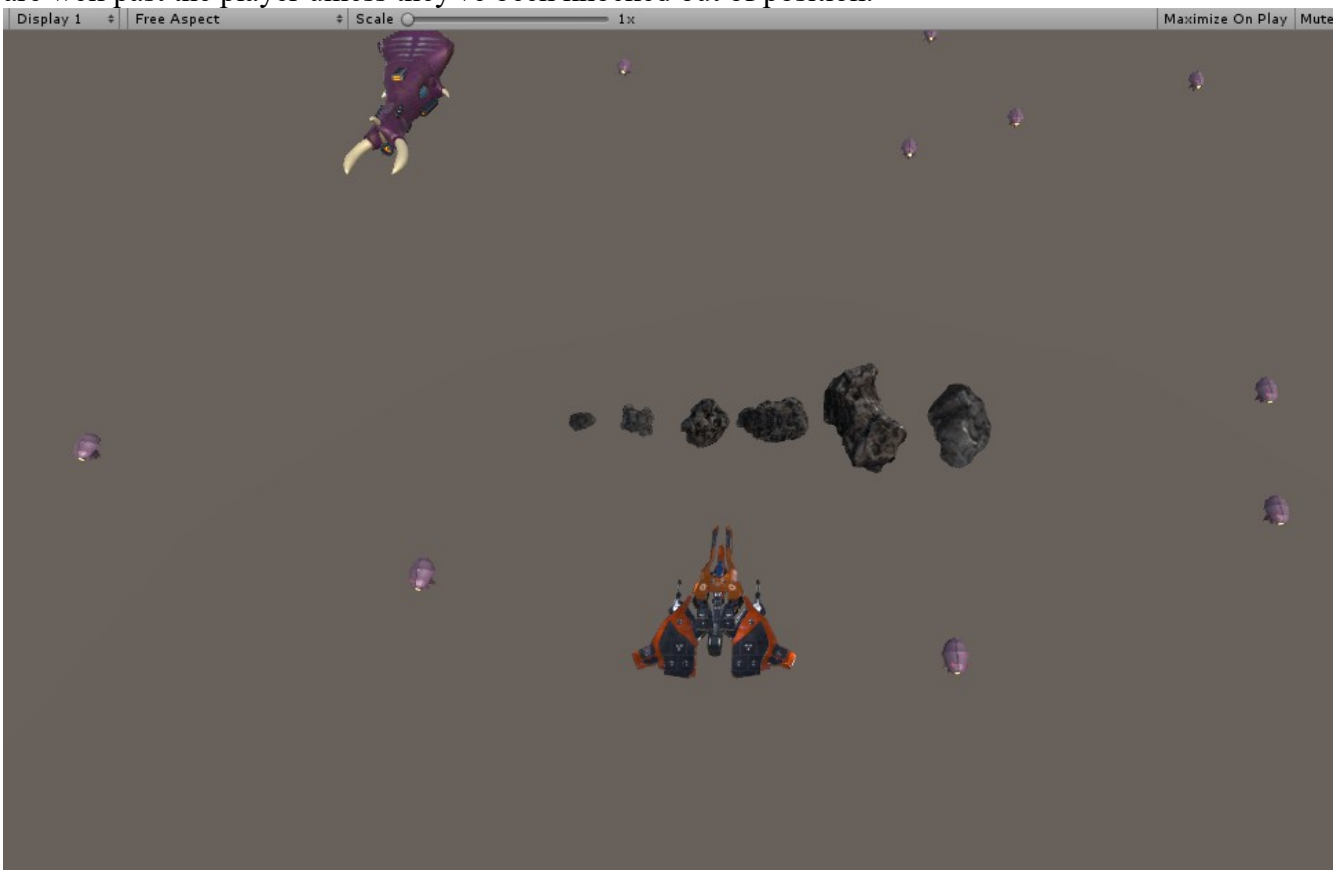
Implementation and Code

I've worked primarily on getting enemies up and running. Currently there are two enemy types, neither of which fires upon the player or takes damage upon being shot.

The larger type spawns once through scripting and patrols back and forth across the screen. It has no other behavior, though if jostled with physics will continue patrolling back and forth as it rotates.

The smaller type spawns in repeated and randomly positioned waves, then flies straight forward. After

ten seconds they are destroyed via scripting to avoid infinite accumulation, though by this point they are well past the player unless they've been knocked out of position.



Further Plans

My primary goals for the future are to refine enemy design and behavior. A few of the major ones are:

-Shooting Back. Currently enemies don't actually engage the player, which obviously hampers their ability to do their jobs. Ideally the screen should be filled with obstacles to avoid and foes to destroy.

-Shooting Pretty: I'd like to experiment with particle effects to produce shinier bullets than we currently have. Notably, the type of ammunition fired gives excellent visual cues about the nature of the enemy you're fighting, so the color and feel of the enemy's weapons has a great deal of thematic function.

-Blowing Up: Currently enemies don't respond to player fire either. Well, more specifically they don't explode; they still respond to the physics of being knocked around by player bullets. While I'd like to maintain this behavior, they should also explode so they player can clear them out and feel good about doing so.

-Multiple Types: Currently there are two rough placeholder enemy types. I'd like to work on implementing different enemy types that behave differently and provide different challenges and contributions to the player's experience.

-Refined Spawning: Current spawning patterns are crude and preset. I'd like to work on refining how the game decides which enemies to spawn. This may be as simple as developing one or more “scripts” (in the movie sense) for it to follow, or it might involve somewhat more organic procedural construction.