Github: https://github.com/pdobbins23/CMPM121-Asgn1.git

Peter Dobbins, Seeya Pillai

Assignment 1: Enemy Spawning

Seeya:

Peter and I worked closely together on the first assignment, collaborating on both the coding and Unity development processes. I primarily focused on reading and storing information from the levels.json file, managing game state transitions between waves, handling player death scenarios, and creating a new enemy class. Since we worked on all of the parts of the project as a team, we were able to quickly solve errors as they occurred by supporting each other through troubleshooting. One of the main challenges we faced was understanding how the various C# scripts interacted within the Unity framework. It took time and careful analysis to figure out how the system was structured. We also explored existing scripts, such as the WaveLabelController, to learn how new features could be implemented effectively and integrated with the existing codebase. This process also taught us the importance of writing clearer, more maintainable code. Throughout the assignment, I significantly improved my debugging skills. Encountering and solving bugs helped me gain a much stronger grasp of Unity's workflows, and I will continue to use those skills in future assignments.

Peter:

I primarily worked on integrating the enemy and level data into the wave system and implementing the RPN (Reverse Polish Notation) interpreter. This involved reading the cached data processed from the JSON files (enemies.json and levels.json), and mapping each of the

fields to attributes of the spawned enemies. I also worked on the implementation of RPN, and incorporated calling and using it when evaluating expressions specified by the JSON files for the various level spawn parameters. I learned how to work with Unity scripts and how they can interact with objects, and how to tie in constructs created in code to the rest of the Unity engine. For example, learning how scripts can expose public variables that can be assigned from within the Unity editor itself, such as a game object reference/handle. I also gained experience working with strings in C# when implementing the parsing for the RPN evaluator. I think this assignment gave me some new insight into how I can structure the content for games outside of source code and into data formats like JSON for better extensibility and scalability.