GitHub Repo Link: https://www.github.com/pdobbins23/CMPM121-Project

**NOTE:** Please check the **asgn3** branch for the code specifically for this assignment.

Seeya Pillai, Peter Dobbins, Astra Tsai

- Additionally, include a pdf file with the names of the team members, **a short description of the three new relics you added**, and a short (1-2 paragraph) reflection by **each** team member on what their contributions were and what they learned.

**1. Crimson Rune**

- **Trigger**: When the player casts a spell (on-cast)
- **Effect**: Gain 3 mana
- This relic rewards consistent spell usage by helping players sustain their mana pool during long encounters.

**2. Silver Wind**

- **Trigger**: After moving 50 units (move-distance)
- **Effect**: The next spell cast gains +50 spellpower
- Encourages mobile playstyles by granting temporary spell bonuses for traversing space.

**3. Lucky Charm**

- **Trigger**: At the start of each wave (wave-start)
- **Effect**: Increase max HP by 10

- Helps players scale defensively throughout the game, especially during longer runs.

Seeya:

In this assignment, I focused on implementing the relic system by integrating the relics.json configuration into the game and extending the existing EventBus architecture. I handled relic parsing, managed triggers and effects in a unified Relic.cs file, and implemented both the core relics and the three custom relics. I learned how to structure data-driven gameplay systems in Unity and how to create scalable systems that can react to player actions using programming that was driven to the events. Also, designing new triggers and effects pushed me to think about gameplay balance and player feedback.

Peter:

I focused on implementing the UI for the relic system, showcasing the player's active relics, as well as the UI for choosing relics to activate after each round. I also implemented the class system, which allows the players to choose how their stats will scale each round at the start of the game. In this assignment, I gained more experience working with Unity's UI system and integrating UI with the underlying game logic, where I had to tie the game's UI to the underlying game state.

Astra:

I focused on fixing a number of warnings and errors when I first received this project. This included adding all the actual code referenced for manipulating the player data when under effects applied by the relics, as well as various other minor type error fixes. For relics, I also implemented the code required to trigger the newly added events, as well as the code for activating the Relics. Later on, I also implemented more fixes and workarounds to make the relic and class UI function without any errors or warnings, added the new relics to the JSON files, and selected the relic sprites. Lastly, I ported in missing code from the last assignment for selecting spells (1-4), and I added a secret debugging cheat code (K) to kill all enemies. Over the course of this assignment, I was provided with a lot of experience in fixing Unity bugs. I also learned how to run Unity on Nix OS, so that's fun.