

Github Link: <https://github.com/pdobbins23/CMPM121-Asgn2.git>

## Seeya Pillai and Peter Dobbins

New Spells Added:

1. **Knockback Blast** (Base Spell)

A high-impact projectile spell that travels in a straight trajectory at increased speed. It deals significantly more damage than standard spells, and its higher mana cost balances the increased speed.

2. **Chaos Amplifier** (Modifier Spell)

A modifier that substantially increases both the damage and speed of the modified spell. It also changes the projectile's trajectory to a spiraling pattern, introducing an element of unpredictability. This modifier also increases the mana cost accordingly.

3. **Rapid Fire** (Modifier Spell)

This modifier reduces the cooldown duration of the modified spell, allowing it to be cast more frequently. To balance this advantage, the modifier also slightly reduces the spell's damage and increases the mana cost, enabling more strategic use.

Seeya:

This assignment provided experience in designing and implementing a flexible, data-driven spell system within our game framework. I developed the logic for reading and parsing spells.json, ensuring that all spell definitions could be dynamically loaded, and created the SpellBuilder class by creating additional files. I also created the SpellBuilder class to generate randomized spells by combining base and modifier spells. I helped integrate the reward system that presents players with new spells after each wave. I was able to learn more about the Unity functionality and working with the GameManager. I designed and implemented one new base spell (Knockback Blast) and two new modifier spells (Chaos Amplifier and Rapid Fire), which created new gameplay effects. Through this work, I gained more insight into composition-based design, runtime configurability using JSON, and the interaction between gameplay systems and user interface in Unity.

Peter:

I focused on integrating the data read from spells.json into the logic for spell casting. I helped implement the reading of the spells.json and converting into a type that can be worked with from the scripts. I implemented the logic for the different kinds of modifiers (i.e. doubler & splitter),

as well as applying the other modifier effects to the projectiles. While working on this assignment, I learned more about working with complex schemas for game data (spells in this case), and how to map the external data into something you can work with more easily from code. I also gained more experience connecting the external game data to actual behaviors in the game, and I find it to be a very useful pattern for games to encode their content outside of code and in data files with something like JSON.