

Final Post

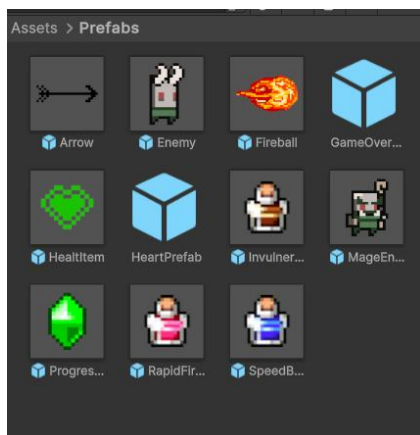
This is the final version of our game. The player spawns with three hearts of health and can move using smooth platformer mechanics that include double jumping and wall jumping for agile traversal.

There are two types of enemies: the normal enemy, which follows and attacks the player with melee strikes, and the Sage, a ranged enemy that throws fireballs. Each enemy presents a different kind of threat, making the gameplay more varied and challenging.

When enemies are defeated, they have a chance to drop several different items:

- **XP gems** that fill the progress bar
- **Hearts** to refill health
- **Speed Potion** (temporary movement boost)
- **Rapid Fire Potion** (increases firing rate)
- **Invincibility Potion** (brief damage immunity)

Once the **XP bar is full**, the player can hold down a button to **progress to the next level**, continuing the run.



We focused on creating a fast-paced, responsive platformer with RPG-like elements and meaningful combat. The combination of enemy types, power-ups, and skill-based movement gives players multiple ways to approach each level.

```
if (validPositionFound)
{
    ObjectType objectType = RandomObjectType();
    GameObject spawnedObject = Instantiate(objectPrefabs[(int)objectType], spawnPosition, Quaternion.identity);
    spawnObjects.Add(spawnedObject);
}
```

This line instantiates a new enemy at a random valid spawn position on the tile map. The objectType is randomly selected, cast to an integer to index into the objectPrefabs array. The enemy is then spawned using Instantiate(), placed at spawnPosition with no rotation (Quaternion.identity), and added to the spawnObjects list for tracking and future cleanup.

While the game currently doesn't have a final boss or end screen, the loop of combat, progression, and movement feels satisfying—and leaves room for future development if we decide to expand the project.

Authors: Antonio and Nicklas