

Assignment Report

Mingyuan Ba SID: 530157791

Introduction

Within the `WizardTD` package:

1. **Manager Class:** The `Manager` class maintains three lists to store fireballs, towers, and monsters. It is responsible for both the battle logic and the logic for generating monsters. Furthermore, the `Manager` class holds references to other classes, ensuring that nearly all functionalities can be achieved through simple method calls to `Manager` from within the `App`.
2. **ConfigLoader Class:** Responsible for reading information from the `config.json` file.
3. **GameBoard Class:** Takes charge of rendering the map and finding paths.
4. **Wizard_house Class:** Manages the `ManaPool`.
5. **InputHandler Class:** Handles user inputs.
6. **App Class:** Responsible for running the game and connecting all the above classes. The classes in the packages `WizardTD.Monsters`, `WizardTD.Towers`, and `WizardTD.Tiles` are respectively responsible for the implementation of monsters (including movement), towers, and tiles (including paths).

Object-Oriented Design Decisions

Class Hierarchy

Monsters

Within the `WizardTD.Monsters` package, the `Monster` class provides foundational functionalities for various monsters. These functionalities include: - Moving (`move`) - Death - Playing death animations - Taking damage

The subclasses, namely `Gremlin`, `Beetle`, and `Worm`, primarily focus on customizing the images and death animations without needing to reimplement the basic functionalities mentioned above.

Tiles

The `Tile` class within the `WizardTD.Tiles` package provides foundational functionalities for various tile types. This includes attributes such as coordinates, index in a two-dimensional map array, ability to place towers, and the corresponding getter methods.

Extension

What I implemented in the Extension section is the “Infinite Mode”. This means that as the waves progress, the number of monsters will increase. Additionally, the monsters’ HP and speed will also escalate until the player’s mana is depleted.

Implementation Details:

1. **Mode Switching:** By setting the `Mode` field at the beginning of the `config.json` file, you can toggle between the Endless mode and the Normal mode (`Infinite/Normal`). Both modes share the same battle logic.
2. **Endless Mode Monster Strength:** In the Endless mode, the increase in monster strength (in terms of speed, quantity, and HP) with each wave is defined by nine keys at the end of the `config.json` file. The values for the monsters will vary within a range specified by `config.json` .
3. **Types of Monsters in Endless Mode:** - i) **Gremlin:** This is the standard monster, similar to what is present in the Normal mode. - ii) **Void Worm:** In the Endless mode, the Worm may randomly appear in the first half of the path to simulate the effect of a “void creature”.