Plants vs Zombies

Atishay Jain | 2018026
Setu Gupta | 2018190

Overview Working

Backend

Contains all the essential classes and backend of the game like Plants, Zombies, Grid, Levels, User, etc.

The game's main computation runs on the backend itself

Frontend

All screens have a dedicated class in frontend classes.

The frontend classes are a means to connect to the backend where the game is playing. Frontend classes with the help of timertask, continuously update themselves according to backends current state

Key concepts used

Singleton

Most frontend classes just need one object as only one played would be playing the game at a given instance. Thus singleton design pattern is used.

Polymorphism

Due to the vast implementation of inheritance, the code is able to exploit polymorphism at multiple places which making method calls.

Multi-Threading

Multiple threads and tasks are created to ensure smooth functioning of the code. We are primarily using javafx.application.Platform for this purpose.

Inner Class and Lambda functions

Inner Classes lambda functions are being used in event handlers of the GUI components depending on the need.

Facade

Using switch cases in Shop to realize relationship between entities in the shop, i.e different Image Views and Plant Objects in this case.

Inheritance

A lot of items share common functionalities, like all the plants and zombies do. We are using different subtypes extend the common subtype. As a result the code is highly modular and around 150 (java + css + xml) files run together to make it work.

Contribution

- Atishay Jain
  - Graphics Design: Designed and collected all the required assets for the game. Created a number of assets himself using illustrator.
  - Frontend Screens: Designed flow of frontend screens and what can be accessed from where..
  - Testing and Tweaking: Testing and tweaking values of various objects to ensure that the game works as it should along making different levels and waves in the game.
- Setu Gupta
  - Backend Design: Designed the main flow of the backend and how relevant information is shared between different classes.
  - Game Logic: Was primarily responsible for coding the main logic of the game window as how frames and refreshed and how different views should be handled.
  - Serialization: Insured that only relevant components are serialized and proper state is resumed at desiralization.
- While both of us had individual tasks defined, none of the task were achieved by either one of us alone. A lot of code was written together so won't be able to distinguish certain parts of the game in terms of who did what. Both of us were involved with most segments of the project and ensured that we were able to debug the code properly and deliver the results in time. It's safe to say there was equal work distribution.

Bonus Components

- Almanac
  - Contains list of all available plants and zombies along with their short description.
- Shop
  - Contains a list of special plants that can't be unlocked during gameplay.
  - Players can use the coins collected during gameplay to buy these plants.
- Shovel
  - Contains shovel to uproot plants if needed.
- Regex
  - Regex is used when a new username is taken to ensure only valid usernames are accepted.