



PLANTS VS ZOMBIES

Aditya Singh Rathore (2018007)
Divyam Gupta (2018032)

IMPLEMENTATION

- The implementation of the project began with a brainstorming session on the various classes and their relationships, following which we formulated the UML diagrams for our project.
- After deciding upon the classes, interfaces and the class relationships, we moved to the next major component of the project, i.e., designing the static GUI for our game. The starting point of course, was SceneBuilder, which eased a major part of the process.
- We decided to use ImageView as our component of choice for the various icons, created the whole layout grid from scratch, designed multi-shaded tiles to achieve a near one hundred percent clone of the original game. From the menu to each button on the screen, everything was brought into existence from scratch.
- Once we were done with the static GUI, it is at this point we realized the actual complexity of achieving a fully functional game. The largest hurdle in the whole process was figuring out a way to work with multithreading in JavaFX to achieve smooth animations as witnessed in the original game.
- With the aim of a smooth gaming experience in mind, we set out by curating design patterns suited for the project. These are covered in the next slide.
- The animations in the project were achieved by using Timeline and KeyFrame classes in JavaFX. To ensure fluid motion, we associated each lane of the game with its own Timeline, The most used transition is the TranslateTransition to translate both zombies and peas from Peashooters.
- The game has a total of 5 levels, which vary in difficulty and user ability by increase in the number of zombies and the number of plants available. Level 4 in specific is exciting as it provides the user with a fixed amount of sun tokens, forcing the user to optimize on the resources he has at hand. Level 5 becomes even more challenging with the existence of a moving belt of plants from which the user can pick any one of them.

DESIGN PATTERNS AND MAJOR ASPECTS

- **Observer Design Pattern** - the Observer design pattern was the core of the whole collision handling system. Whenever the collision handler detects a collision in our project, it checks the type of collision and communicates this information between the objects that are currently under collision. For example, when a pea collides with a zombie, the pea's visibility is set to false and the zombie's health is decreased accordingly. Similarly, these are checked for plant and zombie collisions, mine and zombie collision, etc.
- **Strategy Design Pattern** - the Strategy design pattern comes into use when we consider the movements of the various components on the screen. Each object that can move implements a movable interface and defines its movement discretely.
- **Producer-Consumer Design** - the Producer-Consumer design can be seen in the actions of the Peashooter. Each pea shot by the Peashooter can either go out of the screen or hit a zombie. Our Peashooter always shoots its next pea whenever it encounters any of the two scenarios. It would never shoot a pea arbitrarily.

CONTRIBUTION OF EACH MEMBER

- **Aditya Singh Rathore**

- Designed the static class UML diagram
- Designed the whole layout of the main grid, from tiles to the carousel with the available plant options.
- Designed the Zombie and the Lawnmower classes, developed relationships among various types of Zombies and their attributes.
- Added audio and hover effects on the tiles to provide an engaging user experience.
- Developed the collision handler, detecting collisions between every possible pair of objects.

- **Divyam Gupta**

- Designed the use case diagram
- Designed the menu layouts, buttons, in-game menu for save, resume and exit options.
- Designed the level selection menu.
- Developed the logic for the Plant abstract class, which other types of movable and non-movable plants extend.
- Developed the serialisation and deserialisation blocks by saving the positions of the components and JavaFX does not provide native serialisation ability.

Lastly, both of us figured out the working of the various transitions and animations using Timeline and KeyFrame classes, designed the levels on the basis of an initial test of the game,



For Bonus

Level Variety

- We have added a variety of levels. They are of three types :
 - Normal: Earn sun tokens and place plants. Forst three levels are based on this type.
 - Fixed Sun: This level tests the behaviour of the player when with limited resources. We offer only 750 suns and Player has to manage these resources accordingly.
 - Huge Waves: This level has unlimited currency. However, the number of zombies is huge. Thus, this level tests how good the player is with strategy.

Aesthetics

- The overall game has beautiful effects and graphics, many of which are drawn specially for the game.

