FINAL PROJECT REPORT

“Plants vs. Zombies – Battleships Ed.”

Developed By:

Ivan Torrez

Hayley Ngyuen

Nathen Priyonggo

Mohammed Shoeb Daniyal

Under The Guidance Of:

Evan McCarty

Instructor, CS 342: Software Design

Department of Computer Science

April 2024

COLLEGE OF ENGINEERING

UNIVERSITY OF ILLINOIS AT CHICAGO

1200 W HARRISON ST, CHICAGO, IL 60607

Abstract

Plants vs. Zombies has been the fastest selling game since its release in May 2009. For the final project, i.e., to develop a Battleships game, we decided to impart a theme to it. The theme would be Plants vs. Zombies, but their fight is through the mechanics of Battleships.

The originality of our application comes from the theme. At its core, it functions according to the rules for battleships; liking it to one of the best games ever, is what makes our game, “rad.” We have also implemented an unbeatable bot, we intend to make our demographic, test their strategies or luck against something which reads them like a book.

The chief demographics for this game would be the nostalgic Plants vs. Zombies players as they would be familiar with the game of Battleships as well. Another qualifiable demographic would be anyone who’s looking for a short competitive game. Furthermore, the selling point would appeal to kids around the ages of 8-13 who are attracted to Zombie visuals.

Throughout this document we intend to discuss the design, achievements, and failures of our project. We would also be discussing our teamwork and future expansions and usage for our game.

Game Design

The design is that of Plants vs Zombies meets Battleships. Implementing the rule book of the Battleships game, to make the game further enticing we decided to add a theme to it. The game is designed with the intention of being mobile-friendly, which is why we have decided to shrink the grid, which is the playground for ships.

Walkthrough:

To best explain the design, a walkthrough for a user is the best way to understand it. Primarily, once the game has started the players are met by the username page, which styles the logo we designed by integrating our two main themes and some of the characters that we use throughout the game. The user is asked to enter a username. To further meet our design and aesthetic requirements, we have created a few rules on the username requirements. We added a text field, and our game accepts keyboard input and we have implemented a “Confirm” button. Once a valid username is selected and confirmed, the user is met by the home page:

To implement a leaderboard, we have introduced a Sunlight system. Winning would give a plant, i.e., what the players are, one hundred “Sun Points” while losing would make them lose fifty of the same. This would be in-line with the theme of the game, and it would add more desire to win each game.

Figure 1 Home Page

To make the rules page more interesting, we designed a cartoonish design where each chat describes the rules of the game.

To further the improvement of the design, we decided to implement a blurred background for the rules page. A simple, blurred background where the rules pop up completements the theme of the game and keeps the user interested.

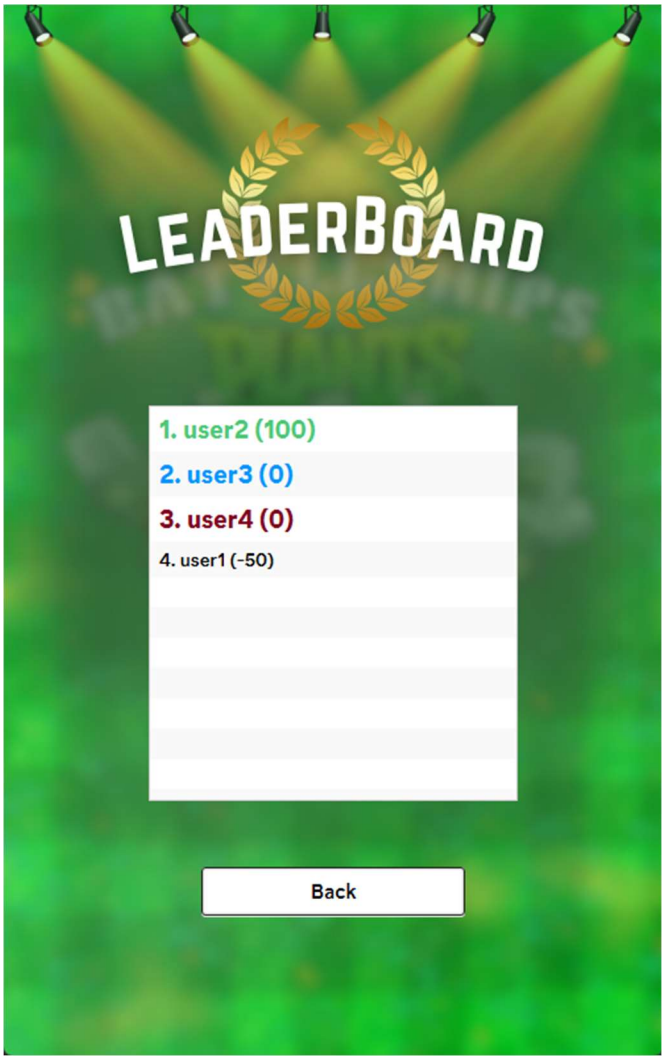


Figure 3 Leaderboard Page

Figure 2 Rules Page

Gameplay:

We have two modes to select from, i.e., Battle online or battle offline. On selecting either, the user is taken to a Placement page where they can place their ships. We have designed and added an instruction for so that the users can easily understand how the placement would work if they were “confused.”

After placing the ships as required the players are led to the loading page, with the

iconic peashooter GIF loading screen to give it character. Once the user is matched with either the bot or another player, they play the game.

The Gameplay is quite simple, the grid is a Grid pane of buttons that will send coordinates of each button on the interaction of a mouse click. This will notify the server of the client's hit request. The user will then be notified if the selected button was a zombie or not. If it were a zombie, it would change the button to a gravestone.

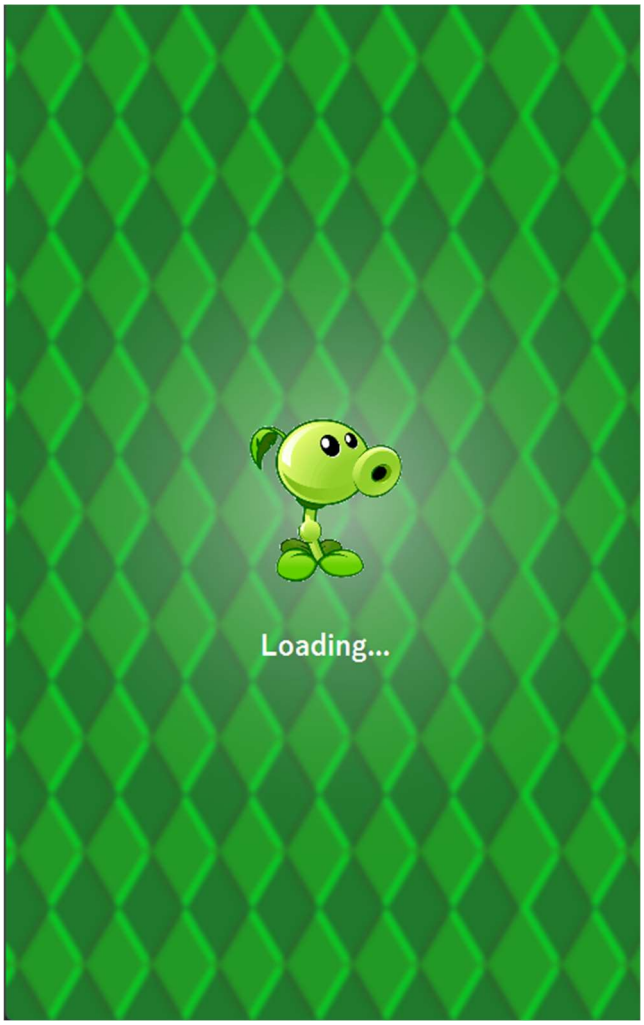


Figure 4 Loading Page

The Game would continue with alternate turns until the end when either player or bot sinks each other's ships completely. The user is then taken to a Win/Lose page depending on their respective result. The home button takes the player to the Home page where they can view their position in the leaderboard after their game to check the positions of other competitors on the server.

Figure 5 The Gameplay

Another way the game would end would be if either of them disconnects, which would make the other player win immediately.

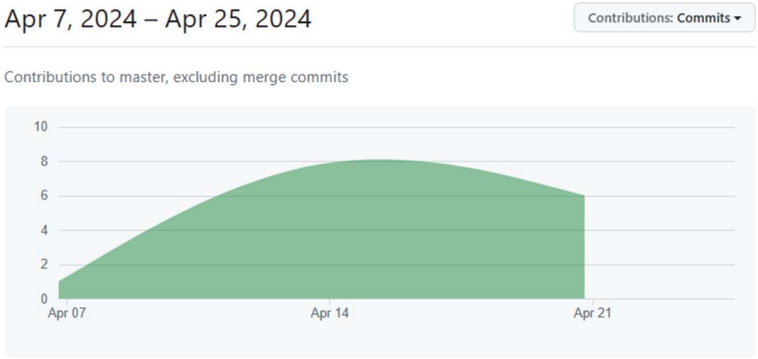
Tools/Technology Used:

We used a variety of tools throughout the development of the game. These would include:

1. IntelliJ Coding Environment & Maven: For its live coding versatility and Maven's project management.
2. Figma: For brainstorming the design and creating UMLs.
3. Java/JavaFX: To build the main structure of the program and the UI.
4. GitHub: For its fast and efficient collaborative medium
5. Canva: To create visually engaging graphics with its vast library of templates and design elements.
6. HTML Color Codes: To match the tone aesthetic of the game's theme.
7. ChatGPT & Oracle: For its vast compilation of Java core libraries.

Timeline & Teamwork:

Graph 1 Client Code commits.



Graph 2 Server Code commits.

The timeline of the project can be dictated by the graphs of commits made. We all used discord as a means of communication for our thoughts and ideas during the initial week, where we had already produced the design using Figma.

We stuck to our roles in the initial project document where our defined responsibilities were executed throughout the month. The work was divided in the following manner:

1. Nathen: Nathen was responsible for the management of the team and creating a designated workspace for each team member. He worked to implement the client-side GUI. He used Canva to design all graphic aspects of the game including the buttons, the background, the logo, etc. Nathen also helped in

implementing the AI and the client-to-client connections, and the classes to manage ships and each grid element. He also completed the initial design documents.

2. Hayley: Hayley worked on the UI of the game. She implemented the initial designs for the Home, Rules, and Placement Pages. And with Nathen's aid, the game was made aesthetically pleasing. She also helped complete the initial design documents.
3. Ivan: Ivan worked on the server's logical flow and put together ideas that are used. He helped in the implementation of the leaderboards and worked mostly on testing the game extensively. He helped in the development of the serializable classes and experimented around the logic of client-server communication. He also worked on the UMLs and initial project report.
4. Mohammed: Mohammed worked on the AI class and helped Nathen implement the AI and client communication. He also assisted in the Ship class to make the AI communicate with the server, extending to the client. He also worked on the initial and final project reports.

Accomplishments & Failures:

We experimented with different ideas, and this section seeks to discuss the overall improvement during the project's timeline. Firstly, with the design, the initial designs

did not stand out, so we added our theme to it making it more pleasing in terms of graphics and design.

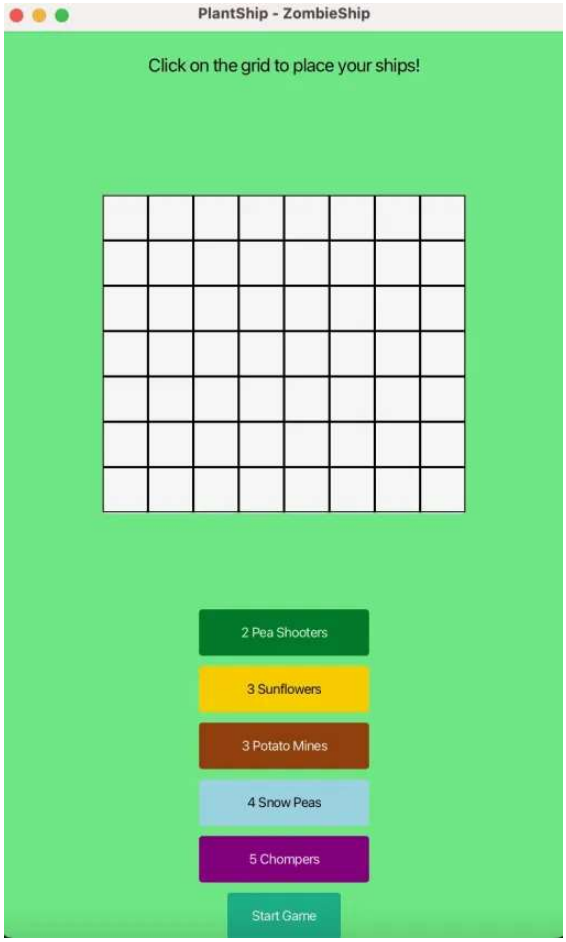


Figure 6 Initial Design

For the backend code, we initially wanted to use a "Grid" class, which would contain a 2D array, however, to save space and decrease the time complexity we use an Element class that would contain the coordinates of each ship. This class would also have the size of the ship and the state of each ship, i.e., if it has been hit or not.

For the AI class, we tried using the Monte-Carlo Tree Simulation (MCTS) algorithm, however, the algorithm was too complex to implement in a game where once the user sets the ships it cannot be changed. We also tried to implement ChatGPT APIs to work the game, but the financial and deadline constraints were a speed breaker. The current AI uses a strategy where it starts at the center of the grid and proxy's through, recording moves that had been hit in a strategic manner.

Feedback:

We presented our design to a few people and asked them to evaluate it, highlighting one thing they liked and disliked:

"Having the Sun points as rank points is a cool detail and brings nostalgia. One criticism is that I wish on the username page, the text would disappear when I type something into it…"

- *Theo Florian Armin Klaus Müller, Student of Philosophy*

"The online battle is fun and, I liked the attention to detail in sticking to the theme. One criticism would be that I wish there was an easy AI, so that players can experience different levels of difficulty."

- *Trung Ton Tran, Student of Computer Science*