SW Engineering Summer C 2023

[FAU: CEN 4010]

Battle Owls Game

Group 5

Team 🗿(Moai)

Jesse McDonald (Scrum Master)
Nicklaus Becker (Possible Front End)
Alexander Ochoa (Possible Front End)
Noah Legault (Product Owner)
Andrew Donate (Possible Back End)

Milestone 4
7/25/2023
Revision History

Product Summary:

1. Name: Battle Owls
2. All Major Committed Functions:
   The game needs to have a smooth movement and be able to correctly detect collisions with objects within the area. The game needs to be able to log a player's information so that when they return they keep all of their progress. The game needs to keep these players’ information secure and be able to efficiently run the logic behind gameplay features so that it feels intuitive to the players. And finally, the game needs to be a beginner-friendly experience that is fun because, at the end of the day, it's a video game made for entertainment.
3. Unique Features: Snake game that implements online or local multiplayer with powerups and customization of the player to make it look how they want it to.
4. https://test-2.phoenixfl.repl.co/

Usability Test Plan:

Major Function: Game

Test Objectives: To enhance the user interface and edit some mechanics or effects within the game or the website to make it a fun experience that is easy to get into for a newer player from different devices or browsers. And refine core gameplay concepts into something that keeps the user engaged and having fun with the game and get some feedback from outside testers to see if there are things that need to be fixed. And finally to test different abilities and effects that might be added or subtracted from the game based on bug testing.

Test Plan: https://test-2.phoenixfl.repl.co/
We are going to be testing the game as a group and sending it to some friends for them to playtest the game. System setup would be sending out the links for people to use and making sure that they are able to access it on their browser and their computer. The only completion criteria would be when the user has felt like they are done or have tested everything there is to play around with in the game such as abilities and cosmetics. Our intended users are going to be people looking for online entertainment as it is a game so our goal is going to be to take feedback and make the game as fun as we can for our users and reduce the amount of bugs or other problems they might have. Whether that be through some modifications of basic game mechanics or the addition or subtraction of certain abilities within the game or the removal of unneeded mechanics from the game entirely.

Questionnaire Form: https://forms.gle/TZXkY93d3sZuvkFt5

QA Test Plan:

Test objectives: To test the quality of gameplay and mechanics and makes changes based on user feedback in order to make the game more fun. We will also be using this as a way to see how players feel about different abilities and power-ups within the game to make sure that everything we are putting into the game has a purpose within the game that people want to use or are fun to use.

Hardware and Software Setup: Using the link to the website on a computer will bring you to the main page where you will be able to log in with your own information and be able to customize your owl before entering the game.

Feature to be Tested: Gameplay and mechanics in order to make sure that we are getting feedback directly from players on the main component of our software project. This will allow us to use the feedback and change the game where necessary to meet their standards.

Test Title	Test Description	Test Input	Expected Correct Output	Test Results
Google Chrome	Testing the game within the Google Chrome browser.	N/A	The working game is within the window.	PASS
Edge	Testing the game within the Microsoft Edge browser.	N/A	The working game is within the window.	PASS

Code Review:

Coding Style:

```
document.addEventListener('keydown', (event) => {
  const keyPressed = event.key.toLowerCase();
    //checks for capslock
  switch (keyPressed) {
   case 'w':
   case 'arrowup':
    direction = 'up'; //click or tap functions for mobile.
    break;
   case 'a':
   case 'arrowleft':
    direction = 'left';
    break;
   case 's':
   case 'arrowdown':
    direction = 'down';
    break;
   case 'd':
   case 'arrowright':
    direction = 'right';
    break;
   case 'p':
    direction = 'pause';
    speedValue.textContent = `0`; //TODO FIX
    break;
  }
 });
```

Best Practices for Security:

Major Assets We Are Protecting:

1. User Data
2. Game Logic
3. User Interface and Customization

Confirmation of Password Encryption:
No login yet so N/A
Confirmation of Input Data Validation:
No input so N/A

Non-Functional Requirements:

1. Performance
    a. The game shall have responsive controls to ensure smooth and enjoyable gameplay. **DONE**
    b. The game shall support a large number of concurrent players in the multiplayer mode without significant performance degradation. **ON TRACK**
    c. The game shall load quickly, minimizing the time it takes for players to start playing. **DONE**
2. Reliability
    a. The game shall have minimal downtime for maintenance or updates to ensure uninterrupted gameplay. **DONE**
    b. The game shall handle errors and exceptions gracefully, providing appropriate feedback to players. **ON TRACK**
    c. The multiplayer mode shall have mechanisms in place to prevent cheating, such as detecting and penalizing unfair play. (TBA) **ON TRACK**
3. Usability
    a. The game's user interface shall be intuitive and easy to navigate, catering to players of different skill levels. **DONE**
    b. The game shall provide clear instructions and tutorials to help new players understand the gameplay mechanics. **DONE**
    c. The game shall support multiple input methods, such as keyboard and mouse, and touch, to accommodate different devices. **ON TRACK**
4. Security
    a. The game shall protect user accounts and personal information from unauthorized access or data breaches. **ON TRACK**

    b. The multiplayer mode shall implement secure communication protocols to ensure the confidentiality and integrity of player interactions. **ON TRACK**
    c. The game shall have mechanisms to prevent and detect cheating, such as unauthorized modifications or hacks. (TBA) **ON TRACK**

5. Maintainability
    a. The game's codebase shall follow modular and well-documented design principles to facilitate future enhancements and bug fixes. **ON TRACK**
    b. The game shall have a version control system (using github) in place to track and manage code changes. **DONE**
    c. The game's development process shall include proper testing, debugging, and error-logging mechanisms to facilitate maintenance. **ON TRACK**
6. Scalability
    a. The game shall be designed to handle an increasing number of players as its popularity grows. **ON TRACK**
    b. The multiplayer mode shall support matchmaking and balancing algorithms to ensure fair and enjoyable gameplay for all participants. **ON TRACK**
7. Compatibility
    a. The game shall be compatible with major web browsers such as Google Chrome, Safari, Firefox, and Microsoft Edge, ensuring a consistent experience across different platforms. **ON TRACK**
    b. The game shall support various screen sizes and resolutions, adapting to different devices and display configurations. **DONE**

GitHub link:
https://github.com/Phoenixflare1579/Owl