                     Fall 2021 Principles of Software Engineering

                                             FAU CEN 4010

**The Vaccine Strikes Back**

Group 23

                                                 Milestone 1

                                                    9/28/21

                                              Revision History

**Executive Summary**

The impact that COVID-19 has brought upon the world is undeniably horrendous. There have been approximately 4.55 million deaths, and the death toll is still rising. Multiple vaccines have been put out, and as of September 1st, 2021 only 39.6% of the world's population has received a vaccination. Currently, 3 major vaccines are available: Pfizer, Moderna, and J&J.

We came up with our idea of our game “The Vaccine Strikes Back” not only to have a bit of fun in these tough times, but to try and help spread awareness of how the vaccine can potentially help stop the spread of the virus. Our game will be similar to how the game “Space Invaders” plays, but with a COVID-19 theme. Basically you will control a vaccine needle(i.e. the spaceship in “Space Invaders”) with the vaccine inside of it. This will be used as ammunition to destroy moving and falling harmful covid cells inside the body. If a harmful covid cell reaches the bottom of the screen or touches the vaccine needle, you will lose points. There will also be some cells that are too strong to destroy, which is where the in-game shop will come in. You will be able to buy power-ups and other items to aid the fight through the in-game shop. As you play the game, you will also be able to replenish ammo from vaccine droplets that will also fall from the top of the screen periodically. Our plan for the in-game shop is for the power-up items to be things that are known to be effective in fighting covid . There will be in-game descriptions for each item to help educate people about what they actually do. In conclusion, we came up with the design of “The Vaccine Strikes Back” to not only have a fun experience, but to also learn more about COVID-19 as well.

**Competitive Analysis**

There are a few different competitor games which have been released with similar gameplay to the one in which we are creating. One of these is Space Invaders, a top down arcade classic released in 1978 where you played as a spaceship defending from an alien attack. Another example is Galaga which was released a few years after Space Invaders in 1981 with added features such as being able to take over an enemy spaceship and play as them. Another game like these which introduced a new mechanic was Asteroids. In Asteroids, you controlled a spaceship which had free movement across the screen instead of a fixed position and shot asteroids out of the sky. The table below summarizes the features included in other games and the differences between those and the one we are creating:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Space Invaders | Asteroids | Galaga | The Vaccine Strikes Back |
| Fixed Spaceship | X |  | X | X |
| Currency/Shop |  |  |  | X |
| Endless | X | X | X | X |
| Varying enemies | X |  | X | X |
| Powerups |  |  | X | X |
| UI/Menu |  |  |  | X |

**Space Invaders**: Space Invaders involves controlling a spaceship located at the bottom of the screen. The spaceship can only move horizontally and shoots vertically based on its position at the bottom. A key difference between “Space Invaders” and “The Vaccine Strikes Back” is the ability to access a shop and UI to buy power ups within the game.

**Asteroids:** Asteroids use a unique control mechanism to control the spaceship to blow up asteroids. Unlike our game, Asteroids allows control of the spaceship on any part of the screen, losing the restriction of staying at the bottom of the screen. This opens up the field that the player interacts with.

**Galaga:** Galaga offers many of the same features found in our game including a powerup and different types of enemies. In Galaga, if you destroy a boss ship you can gain the boss ship as an upgrade to be attached to your ship, doubling your damage against enemies. A key difference here is instead of dropping upgrades from enemies, enemies will drop ammunition for our ship to use. The shop will be the primary place to find upgrades for our ship.

**The Vaccine Strikes Back:** The Vaccine Strikes Back is meant to fulfill all of the features of the previous competitors and add even more to this classic game. It is also meant to be a fun and educational game which can provide a learning experience unlike the previous games. The item descriptions will allow people to learn about real life parts in the creation of the vaccine and the menu and shop added into the game allows more control and freedom over the functions of the ship. We choose to keep a fixed spaceship design to allow a finer area where we can control what happens in the player space. This allows for a higher quality game which requires more skill than luck and controls which are simple for anyone to learn.

**Data Definition**

**Use Cases**

**Use Case - Start**

User wants to start a game. User will come to a home page where he would use a “Play” button in order to be taken to a game screen to begin a game.

1. **Description:**

Use case will describe a process of how the user will start a game.

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

4. **Primary flow of events:**

4.1 User opens the application and arrives on a home page

4.2 User selects a play button

4.3 System displays a quick tutorial

4.4 User closes a tutorial and starts a game

4.5 Terminate Use Case: Start

**Use Case - Settings**

User wants to adjust game settings such as volume and account settings. User will come to a home page where he would use a “Settings” button to get to a settings screen.

1. **Description:**

Use Case will describe a process of how the user gets to the settings screen to adjust settings.

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

4. **Primary flow of events:**

4.1 User arrives at a home page

4.2 User selects “Settings” button

4.3 System takes user to a Settings page

4.4 User adjusts Settings according to their needs:

4.4.1 User may adjust their volume

4.4.2 User may login or log out of their account (perform Use Case: Login)

4.5 System saves users adjustments

4.5 User presses “close” button

4.6 System takes user back to home page

4.7 Terminate Use Case: Settings

**Use Case - Login**

The user wants to login into their account to save progress and view the leaderboard. User would come to the “settings” menu and click the “Login” button.

1. **Description:**

Use Case will describe a process of user logging into existing account

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 User has an internet connection

3.2 System is available

3.3 User has an account

3.4 User is in settings menu

4. **Primary flow of events:**

4.1 User arrives to “Settings” menu

4.2 User clicks on a log in button

4.3 System prompts user to enter their Login and Password information

4.4 User enters their information and clicks “Login”

4.4.1 If user doesn’t have existing account, go to alternate flow 5.1

4.5 System logged user into a system

4.6 Terminate Use Case: Login

5. **Alternate Flows:**

5.1 User Does not have an existing account:

5.1.1 Perform Use Case: Create Account

5.1.2 Move to step 4.3

**Use Case - Create Account**

The user wants to create their account to save progress and view the leaderboard. User will go to the settings menu and click on “create account”.

1. **Description:**

Use Case will describe a process of a user creating a new account.

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 User has an internet connection

3.2 System is available

3.3 User is in settings menu

4. **Primary flow of events:**

4.1 User arrives to “Settings” menu

4.2 User clicks on Create Account button

4.3 System Prompts user to Enter a new Login and Password

4.4 User enters login and password information and clicks create button

4.5 System Checks for the availability of the Login

4.5.1 If login isn’t available go to alternate flow 5.1

4.6 System saves Account information

4.7 Terminate Use Case: Create Account

5. **Alternate Flows:**

5.1 If Login is take by another user

5.1.1 System will notify a user that login isn’t available

5.1.2 Go back to 4.3

**Use Case - Leaderboard**

The user wants to view their score on a leaderboard and compare it to other players.

1. **Description:**

Use case will describe a process of user opening and tracking a leaderboard

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 User has an internet connection

3.2 System is available

3.3 User has an active account

3.4 User is logged into their account

4. **Primary flow of events:**

4.1 User arrives to a home menu

4.2 User clicks on a leaderboard button

4.3 System takes user to a Leaderboard menu

4.3.1 If user isn’t logged in, go to alternate flow 5.1

4.4 User can scroll and see their position on a leaderboard

4.5 User clicks on “X” to go back to home menu

4.6 Terminate Use Case: Leaderboard

5. **Alternate Flows:**

5.1 User isn’t logged in:

5.1.1 Perform Use Case: Login

5.1.2 Go back to 4.3

**Use Case - Shop**

The user wants to purchase improvements with an ingame currency to better progress in the game.

1. **Description:**

Use Case will describe a process of how user will interact with a shop

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

4. **Primary flow of events:**

4.1 User arrives to a home menu

4.2 User clicks on a shop button

4.3 System takes user to a shop menu

4.4 User clicks on the item he wants to buy/upgrade

4.4.1 If funds are insufficient, go to alternate flow 5.1

4.5 System processes users selection

4.6 User can go back to step 4.4 if necessary

4.7 User can click “X” to go back to home menu

4.8 System takes user to home menu

4.9 Terminate Use Case: Shop

5. **Alternate Flow:**

5.1 If funds are insufficient

5.1.1 Notify user that funds are insufficient

5.1.2 Take user back to step 4.4

**Use Case - Shoot**

The user wants to shoot the vaccine in order to destroy Covid particles.

1. **Description:**

Use Case will describe how user will use shooting function of the game

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

3.2 User is in the game menu

3.3 User has ammunition

4. **Primary flow of events:**

4.1 User arrives to a game screen

4.2 Covid Particles appear

4.3 User will press a “shoot” button

4.3.1 If there is no ammo, go to alternate flow 5.1

4.4 System will register the action and displays shooting animation and shows ammo particle

4.5 System will decrease ammo count

4.6 Terminate Use Case: Shoot

5. **Alternate Flow:**

5.1 If there is no ammunition

5.1.1 Make a corresponding sound

5.1.2 Go to step 4.6

**Use Case - Collect Ammunition**

The user wants to collect vaccine particles in order to replenish ammunition.

1. **Description:**

Use case will describe a process of a player collecting the ammunition

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

3.2 User is in the game menu

3.3 Ammunition is on the game canvas

4. **Primary flow of events:**

4.1 User arrives on a game screen

4.2 System creates and displays ammunition particle on the canvas

4.3 User performs Use Case: Move

4.4 User hovers over the ammunition particle

4.5 System registers the interaction of the ammunition and a player

4.6 System increases amount of ammunition

4.7 Terminate Use Case: Collect ammunition

**Use Case - Destroy**

The user wants to destroy Covid Particles in order to get points.

1. **Description:**

Use Case will describe the process of destruction of covid particles

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 System is available

3.2 User is in the game screen

3.3 System has created Covid particles

3.4 User has ammunition

4. **Primary flow of events:**

4.1 User arrives to a game screen

4.2 System created Covid particles

4.3 User performs Use Case: Move to line up with Covid Particle

4.4 User performs Use Case: Shoot

4.5 If ammunition particle touched Covid particle

4.5.1 System registers a hit

4.5.2 Covid particle life points will decrease

4.5.2.1 If life points reach 0 go to step 4.7

4.6 If life points are above 0, go back to step 4.3

4.7 System registers a destruction of a particle and hides it from the game screen

4.8 System registers points for destroying a Covid Particle

4.9 Terminate Use Case: Destroy

**Use Case - Move**

The user wants to move around the canvas in order to shoot and destroy covid particles, as well as collect ammunition.

1. **Description:**

Use case will describe all cases in which the user can move across the board to complete certain actions.

2. **Actors:**

2.1 User

2.2 System

3. **Preconditions:**

3.1 User is stationary on the canvas.

3.2 User wants to move to shoot the vaccine at a covid cell.

3.3 User wants to move to collect a piece of the vaccine to replenish ammunition.

4. **Primary flow of events:**

4.1 Covid cell appears on the user's screen.

4.2 User uses arrow keys on keyboard to move the vaccine to shoot said covid cell mentioned in 4.1.

4.3 User uses arrow keys on keyboard to collect a potential spawn of the vaccine to replenish ammunition.

**Initial list of High-Level Functional Requirements**

* **Authentication** - The user will have to create a username / account in order to use this product. The user will be able to log in and log out whenever he or she pleases. The current plan is to only have the user create an account with only their first and last name as well as the username they would like. Later on we may switch to an email type of authentication or a username and password one.
* **Reporting** - There will be a report ticket/system implemented in the game, this will most likely be located in the login page or somewhere where the user can easily notice it. The report system will help us developers address issues or bugs that the user(s) happens to come across. Furthermore we will have some contact emails as well in case there is a fault with the report ticket system, that way the customer can address one of us developers directly about the issue with the product.
* **Historical Data -** The user’s progress within the product such as (High Score, Enemies destroyed, ect) will be stored and updated whenever the user manages to surpass their previous score.
* **Recovery** - The user will be able to use a type of function similar to the report ticket system, in case they happen to forget their login/authentication info for the product. This function will likely work almost exactly a “forgot username or password” function that is commonly used for other applications. The recovery system is on our backlog of things to develop however it is the most urgent thing that we are currently working on. Once our team has managed to get the product working , this function will be implemented.

**List of Non-Functional Requirements**

* **Performance -** The performance of this product will not be demanding on hardware as it will be used in the web itself. There should also be no delay in the response time in regards to inputs for the controls.
* **Reliability -** Since the product is web based, the main thing that is reliant on is that the user has access to the internet/wifi/service, and of course a computer or some other electronic device that can access it. We also plan to keep the product running non-stop unless there is an issue that needs to be addressed.
* **Security -** For this product we will be using Firebase , which is a Google created software that helps with security and creating web applications such as this. Furthermore the security regarding the user data (passwords, ect) will be handled by Google through Firebase.
* **Availability -** The product will be available in a short matter of weeks once we are able to launch it. Once it is released on the web , we will closely manage how it progresses and will decide if we will work on a non-web version through an app store. The product should be available 24/7 on the web once released unless it requires some maintenance.
* **Usability -** For this web product the user is required to have some sort of internet access as well as a phone/computer/android device. Furthermore if the user plans to use it via a computer it is also required to have access to a keyboard , as the product will not be usable with a mouse alone.
* **Services and Manageability -** For this product we plan on using a MySQL Database in order to store the user’s login information as well as other data through actions on the website. This data will be managed by us the developers, as well as other services in case the user may have found a bug/exploit that they would like to report.

**High-Level System Architecture**

1. **mySQL Database** - mySQL is the database we will be using to handle the data in our project. Users will be adding to the database through actions on the website and we, the developers, will be managing the data on our side.
2. **Browser Compatibility** - Our game will be a web-based application that will work on all major browsers.
3. **Visual Studio Code(IDE)** - Visual Studio Code is a code editor that we will be using to create the code for our web game application. The following languages will be used in our project:

                    a.) **Javascript** - Javascript will be the language used for the backend of the game to make it very interactive and enjoyable for the user.

                    b.) **HTML(HyperText Mark-Up Language)** - HTML will be the language used for the frontend to ensure that the UI(User Interface) is visually appealing and enjoyable for the user. HTML will allow the browser to display the game.

1. **fau.edu Lamp Server** -The FAU lamp server is what we will be using to host our project.

1. **P5.Play** - P5.Play is a Javascript library for the creation of games. This is what we will be using for all of our animations, interactions, collision detections, etc.

1. **Google Firebase** - We will be using Firebase to handle user authentication for our project. We will not be able to see user passwords, and encryption is handled on Google’s end. This seemed like the best option for our type of project.

**Team Members and Roles**

Robert Johnson(Product Owner) - [rjohnson2018@fau.edu](mailto:rjohnson2018@fau.edu)

Tony Rojas(Scrum Master) - [trojas2019@fau.edu](mailto:trojas2019@fau.edu)

Renan Quiroz(Backend) - [rqueiroz2019@fau.edu](mailto:rqueiroz2019@fau.edu)

Ajay Seedoo(Backend and Frontend) - [aseedoo2020@fau.edu](mailto:aseedoo2020@fau.edu)

Mykyta Palamarchuk(Frontend) - [mpalamarchuk2019@fau.edu](mailto:mpalamarchuk2019@fau.edu)

**Checklist**