CEN 4010 Principles of Software Engineering, Fall 2019

Milestone 1 Project Proposal and High-level description

Owl Gamers

Team number: 5

Monika Spasovska - Product Owner mspasovska2017@fau.edu

Rolando Leiva - Scrum Master rlevia2017@fau.edu

Stephanie Juan - Development Team sjuan2017@fau.edu

Gandhi Lucate - Development Team glucate2015@fau.edu

09/22/2019

**Executive Summary**

The following is a proposal for a Principle of Software Engineering Project consisting of a web-based video game. The game is an arcade shooter game in which the player uses a fixed shooter to defend their base from incoming enemy ships. The name of the game is Captain Defender. The objective of the game is to defeat wave after wave of enemy ships with a vertical-shooting laser weapon, much like Space Invaders.

This project is solely software-based and does not require any hardware. The game will only be accessible by registered users so a log-in and registration page will be provided. The user’s information will be stored in a database. Once a user signs in they will be able to start the game. The game will have multiple levels so the user can progress. The purpose of the game will be to defend a home baseline at the bottom of the screen from invading ships. The game will allow for linear horizontal movement and vertical projectiles. If a projectile hits an invader, the invader will be destroyed and disappear. If the user can defeat all the invaders, they win, and if not, they lose the game. The achievements of the user can be stored and shared. The website will run on a server with limited access by any user. By including this feature the user will be able to add players and track their achievements. A separate tab will be dedicated to user interactions with others on the platform. A feature to post statuses and share achievements will promote communication between others.

Our motivation for developing this project is to create a fun user platform where gamers can gather, play a fun game and be social. The whole team is highly motivated in creating a fun experience for everyone and we believe gaming is a huge industry with lots of opportunities. The average teenager and young adult will most likely delve into some sort of gaming, whether it be console, PC or mobile. We want to give the user a unique PC gaming experience they will be able to share with friends and strangers online, compete, connect and share statuses. This project can eventually lead to a whole online gaming platform with many other games to match everyone’s taste and style.

**Competitive Analysis**

|  |  |
| --- | --- |
| **Captain Defender (Personal)** | **Space Invader (Competitor)** |
| Statuses | Point for shooting alien |
| Friends | Defense bunkers |
| User account | Game music |
| Achievements | Arcade use only |
| Themes | Singular theme |
| Multiple difficulties | Fixed difficulty |

Captain Defender is going to be more interactive than previous versions of similar games like Space invaders. The user is going to be able to communicate with friends, have a profile and receive achievements. Captain Defender’s unique feature is that other games like it do not have a player to player interaction like our game. Our game will also change themes as the user progresses through the levels for a more versatile and entertaining user experience.

**Data Definition**

**Bootstrap** – open-source CSS framework for front end development.

**CSS**- Cascading style sheets is a stylesheet language, that describes documents written in HTML or XML. It also describe how elements should be rendered on screen, on paper, in speech, or on other media.

**Cyberduck** - is an open-source client for FTP and SFTP. Cyberduck is written in Java and C#.

**Database**- A structured set of data held in a computer, especially one that is accessible in various ways.

**Eclipse** – an integrated development environment (IDE) used in computer programming.

**FTP** – The File Transfer Protocol is a standard network protocol used for the transfer of computer files between a client and a server on a computer network.

**Html** – HyperText Markup is a standard markup language for creating web pages.

**Java** – General purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

**Junit** –is a widely used testing framework along with Java Programming Language.

**LAMP Server** – Is an open-source Web development platform that uses Linux as the operating, apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language**.**

**Microsoft Visual Studio** – is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, websites, and web apps.

**SFTP** – the SSH File Transfer Protocol is a network protocol that provides file access, file transfer, and file management over any reliable data stream.

**Space Invaders** – is a 1978 arcade game created by Tomohiro Nishikado. Space Invaders is a fixed shooter and set the template for the shoot ‘em up genre.

**Overview Scenarios and Use Cases**

Our website will greet the user with a login page. If the user does not have an account, they will have to sign in and create one by inputting their full name, email address and creating a password. Once they are signed in or logged in they can proceed to the game. They can start the game and if they win they move on to the next level. If they lose, they fail and the game restarts. Once they pass a level they will get an achievement. They can either stop the game and share their achievements with their friends or continue to the next level. If they choose to share their achievements all their other friends will be able to see it. They can also post a status about it which others can see and interact with. If the user does not feel like playing the game, they can go to the status page and see what other users are up to. Here they can reply to or read other statuses and feel like they are connected to the other users on the page.

John goes to the webpage and is greeted by a sign-up page. He must input his full name, username, email address and password. Once he’s signed in with the correct information, he has access to the website. He can start by selecting an avatar for his icon. Then he enters the gaming portion of the website. He starts playing the game. He loses the first time, so the game resets and he tries again. He wins and reaches the next level. He keeps playing until he reaches the next level. He finishes the game and signs out.

Jenna has created her account on our website. She logs on and plays the game. She feels as if she got an amazing score and wants to share it with her friends on the website. She shares her achievements and posts a status. Her friends comment and her status and share their scores with her.

Jim has created his account on our website. He logs in but does not want to play a game, instead wants to connect with his friends. He goes on the status page, reads their updates and comments on all of them. He gets replies from his friends and he replies back. He has checked in with all his friends and logs out.

**Initial List of Functional Requirements**

1. Login page

* The login page will have the user enter their information and the information that is entered by the user will be stored in a database. After the user has created their account, they can log into the website using their newly created account.

1. Score page

* This page will display the scores of the players of the game. The score page will display the highest score. Also well as high scores, the list will show the users who have played the game. This version of the scoreboard will show which user has played the most games, their highest score, and the average score for each user. The game will be a

1. Game

* The game will be playable on the website and allows the user to gain points. The points gained by the user will be shown to the player and then stored into the database. The player will need to log in using their login credentials in order to play the game.

1. Database

* The database will hold all of the information from the website and the game. The database will have a table that will hold the login information as well as the player’s first name, last name, email, username, and password in order to log into the website. Each username will be unique, making the username a primary key in the information table.

1. Achievements

* The player will receive achievements based on their performance in the game. The achievements will be unlocked when the player meets the requirements listed by the achievements. Once the achievement is unlocked, the player can put the achievements on his or her profile. Other players can view anyone’s achievements.

**List of Non-Functional Requirements**

1. Usability

* The website will be a gaming platform to welcome new users and will be easy to navigate through. Once users get a look and feel for the platform they can easily begin playing the game, posting status updates and achievements, and connecting with other friends on the website.

1. Performance

* The response time should be 2 seconds or less starting with the login and applies to posting statuses and achievements. The website will be able to support up to 50 people on the server at any given time to keep the response time to 2 seconds or less.

1. Security

* Owl Gamers website shall not execute commands embedded in data provided by users that forces the application to manipulate the database tables in unintended ways. No user will be able to see your personal information unless posted on your profile, and vice-versa.

1. Maintainability

* The website and game will have a 90% probability that any errors will be repaired within an hour. The coding will be simple enough that future maintenance will be quick and efficient, which should also be adaptable to any new requirements.

1. Accessibility

* The users will be able to access their profile and update their personal information and icons at any time. One user will not be able to go onto another user's profile and change anything or view any of their personal information that is not public.

1. Data Storage

* Any personal information that is given and input to the website and profiles will be stored into the database such as: Username, First Name, Last Name, and Email to name a few. Any achievements gained will have the option to be posted publicly and will be included in a list of user achievements.

1. Availability

* Any user who has created an account will have the website and game available at any time in the day.

**High-Level System Architecture**

Java will be the primary language used to manage the database and create a game that will be input data into the database. Having both the database and the game be managed by the same language will help the merge of code from different team members. Eclipse will be used to code the game. Some team members have coded in Eclipse before. Eclipse allows the team members to view any mistakes in my code in real-time and can use Junit to create test cases. Being able to create test cases will make the testing process more efficient since all the test cases will be consistent unlike entering data manually. The website will be edited using HTML. Visual Studio will be used to access and edit the HTML and CSS files. Cyberduck will be used to upload the files to the lamp server.

**Team**

Monika Spasovska - Product Owner

Rolando Leiva - Scrum Master

Stephanie Juan - Development Team

Gandhi Lucate - Development Team