CEN 4010 Principles of Software Engineering, Fall 2019
Milestone 3: More Detailed Requirements, Architecture, and a Vertical Software Prototype

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 high score of the user will be stored on their personal profile page and are automatically updated on the top 10 high scores(if applicable) on the scores page, that can be viewed by any user on the platform. 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 on the scores page if high enough to be apart of the top 10 achievers on the site. A separate tab will be dedicated to user interactions with others on the platform. A feature to post statuses 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 a 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	No user to user interaction
Friends	Single-player use only
User account	No storage of user information
Top 10 High Score Achievements	Arcade use only
Themes	Singular theme
Multiple difficulties	Fixed difficulty

The game will have the same features as the competitor as well as the features of the team as listed in the table above. Captain Defender will have all the features listed above. There are multiple features that differentiate Space Invader and Captain Defender. Captain Defender is designed to be the definitive edition of both games. Learning from the archaic features of Space Invader, Captain Defender will improve on the shortcomings of the original game as well as staying true to what made the original so unique. 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 view a friend's activity through statuses, have a profile and possibly showcase their high score on the top 10 high scores page to every user on the platform. 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

Attack - The amount of damage the player or enemy does to their opposition.

Attack Multiplier (AM) - The number that amplifies the enemy's attack.

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

Captain Defender – a game character played by the user. Captain Defender will have to defend his turf.

CSS- Cascading style sheets is a stylesheet language, that describes documents written in HTML or XML. It also describes 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.

Defense - The amount of damage that is reduced from the player every time the enemy hits the player.

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.

Heads Up Display (HUD) – displays the source and will display a progress bar in future updates

Health Points - The number of hits the player can receive before the game ends.

High Score - The highest amount of points the player has received while playing the game.

Home Base – The player has to protect his turf from the Invaders. A place the player can be in without the threat of the invaders.

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.

JavaScript – is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions.

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.

Score – points earned from performing well in the game

Session – a session is a temporary and interactive information interchange between two or more communicating devices, or between a computer and user.

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

Shields – Placed in front of the player to block some of the damage the invaders' shoot.

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.

Swarm – A large number of enemies or invaders.

The player – controls the Captain Defender using the keyboard and tries to get the highest score possible.

Themer – The subject matter the game is built around.

UML – Unified Modeling Language

UML Diagram – is a diagram that visually represents a system along with its components.

World Invaders – enemies to be destroyed in the game. These will try to take over Captain Defender's turf.

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 a score. If the score is high enough, it will be displayed as the top 10 scores amongst everyone on the website. They can also post a status about it which others can see. If the user does not want to play the game, they can go to the status page and see what other users are up to. Here they can read other statuses and feel like they are connected to the other users on the page.

<u>Use Case Scenarios:</u>

1. John goes to the webpage and is greeted by a login page. He does not have an account so he must use the 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. He also has the right to enter a short bio that appears on his profile card. 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.

2. 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 compare it with her friends on the website. She goes to the achievements page and sees she ranks #2 amongst all the users on the website. She can post a status letting her friends know of her score.
3. 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 of all his friends. He then posts a status of his own to let his friends know how he's doing. He checks the scores page and tracks his friend's progression. He sees how everyone ranks on the website. He has checked up on statuses and scores and logs out.

High-Level Functional Requirements

1. Login Page - The Owl Gamers website shall have a login page to protect against unwanted users.
   1.1 The login page shall have users enter their information to access the game.
   1.2 After the user has created their account, they shall log in to the website using their newly created account.

2. Top 10 Scores Page - The Owl Gamers website shall have a Top 10 Scores page where players can view the Top 10 Scores of the game.

   2.1 This page shall display the scores of the Top 10 scores by the players of Owl Gamers.

3. The Game - The Owl Gaming website shall have the Captain Defender game that allows players to gain points.

   3.1 The game shall be playable on the website and shall allow the player to gain points.

   3.2 The points scored shall be shown to the player.

3.3 The player shall need to log in using their Owl Gamers account credentials in order to play the game.

4. The Database - The Owl Gaming website shall have a database that will hold all of the player's information.

   4.1 The players shall be able to see the Top 10 scores of the game by all players of Captain Defender.

   4.2 All of the information the player enters or achieves shall be stored in the database.

5. User Profile - The Owl Gaming website shall have each user's high score posted on their profile.

   5.1 Each player shall be able to view their own high score.

List of Non-Functional Requirements

1. Usability - The Owl Gaming website shall be a gaming platform that welcomes new users and shall be easy to navigate through.

   1.1 Once users get a look and feel for the platform they can easily begin playing the game, seeing their high score on their profiles, and be able to view the Top 10 Scores of the game amongst all the players of Captain Defender.

2. Performance - The Owl Gaming websites response time should be 2 seconds or less starting with the login and applies to viewing each individual updated high score and viewing the Top 10 players scores.

   2.1 The website shall be able to support up to 50 players on the server at any given time to be able to maintain response time.

3. Security - The Owl Gaming website shall not execute commands embedded in data provided by users that forces the application to manipulate the database tables in unintended ways.

3.1 No user shall be able to view any other users' personal information unless posted publicly by the user.

4. Maintainability - The Owl Gaming website and the game shall have a 90% probability that any errors encountered will be repaired within an hour.

   4.1 The coding shall be simple enough that future maintenance can be edited quickly and efficiently, which will also make the coding simple and adaptable to any new requirements.

5. Accessibility - The Owl Gaming website users shall be able to access their profile and update their information and icons at any time.

   5.1 No other users aside from yourself shall be able to change or view any of your personal information that is not public.

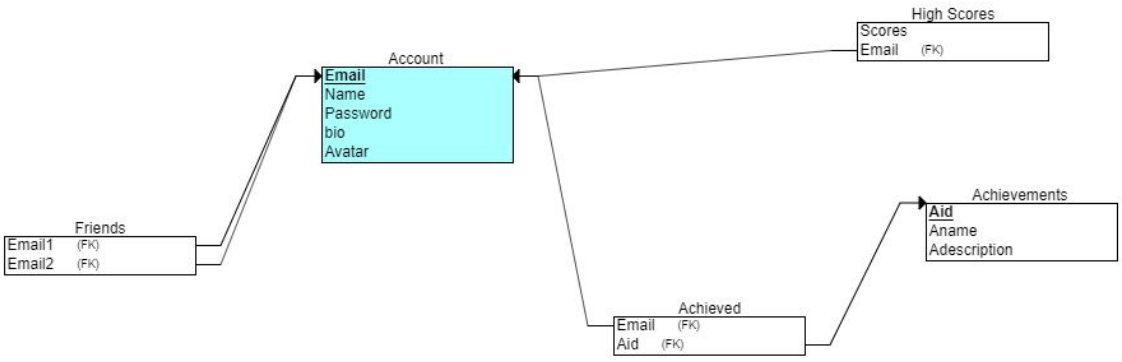
6. Data Storage - The Owl Gaming website shall store any and all personal information input to the website into a database such as Name, and Email.

   6.1 Individual players' high score shall be publicly viewable on their profiles.

7. Availability - The Owl Gaming website and game Captain Defender shall be available to any user who has created an account at any time in the day.

High-level system architecture and database organization

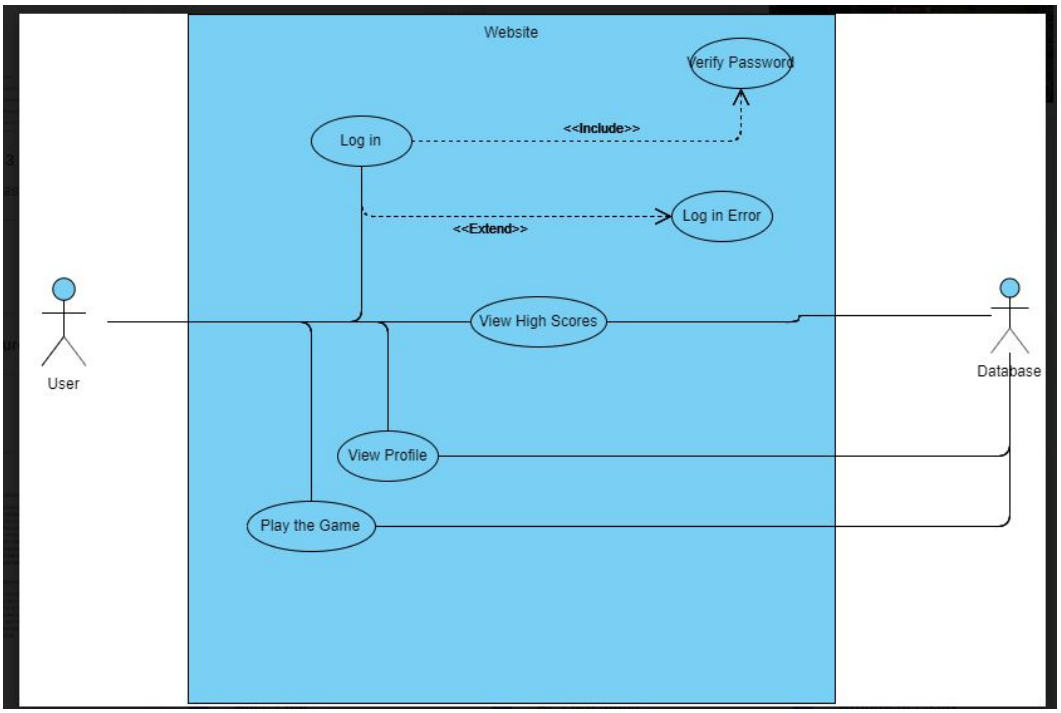
1. Most of the code has been written in PHP or HTML. The game will be coded in Java and will follow the class diagram in the High-Level UML diagrams section.



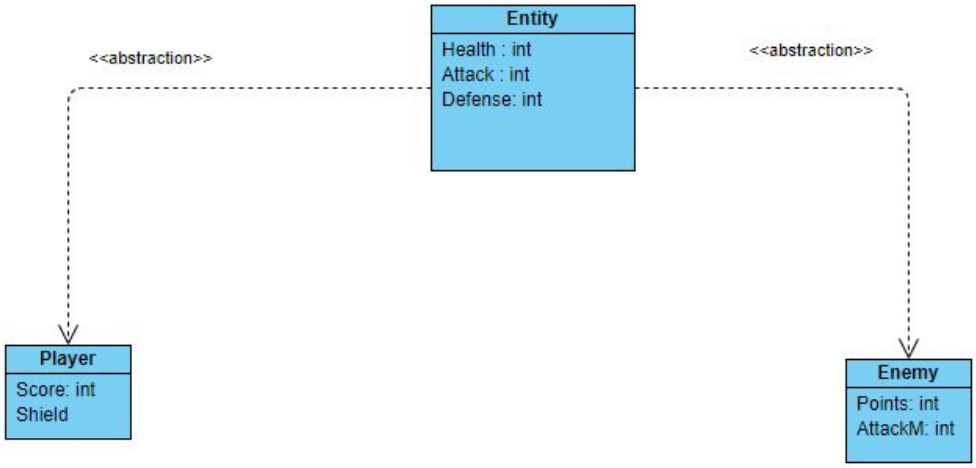
2.
3. Media Storage: The database will hold images and an audio file to play in the background for the game.
4. The terms that are searched for the database are the email, name, and scores of each user. The email is used to identify each user from each other, the name is displayed at the home page and the high scores will be filtered and sorted. The high scores page will display the highest scores and can be filtered based on the user.
5. The team can create a game that will be played through the website. This game will allow players to earn a score that will be stored in the database. The highest score will be on the user's profile page
6. The rating of the game will depend on the user's score. The higher the score, the higher the user will be placed on the scores page.

High-Level UML diagrams

Use Case



Class Diagram



Current Key Risks

1. Skill Risks: Every team member has their strengths and weaknesses. Some prefer front end development and some prefer back end. The skills of every team member have been communicated internally, so work is split fairly. For example, if a team member is good at database system management, they will focus their time and effort on that rather than designing the aesthetics of the website.
2. Schedule Risks: Three out of the four members on the team are distance learners and prefer online classes due to the distance from our living situation to campus. For this reason, it is quite difficult to gather everyone physically in one location to work on the project. A group chat and Discord have been set up for team communication. The group chat and Discord are sufficient methods for communication as a substitute for meeting face-to-face. Other obstacles that stand in the way of meeting regularly are external

individual commitments such as jobs, work trips, and other timely engagements. The team members have agreed to communicate their time conflicts in advance, so the work can be done in a proper time manner.

3. Teamwork risks: Some of the members were not meeting their potential when it came to submitting parts of the project they were assigned. The teacher was notified in an attempt to steer the team in the right direction. The team was given a second chance and motivation to be more involved. The issue should be resolved.

Peer Evaluation

Point Distribution:
Monika Spasovska- 37.5/25
Rolando Levia- 37.5/25
Stephanie Juan- 25/25
Gandhi Lucate- 0/25