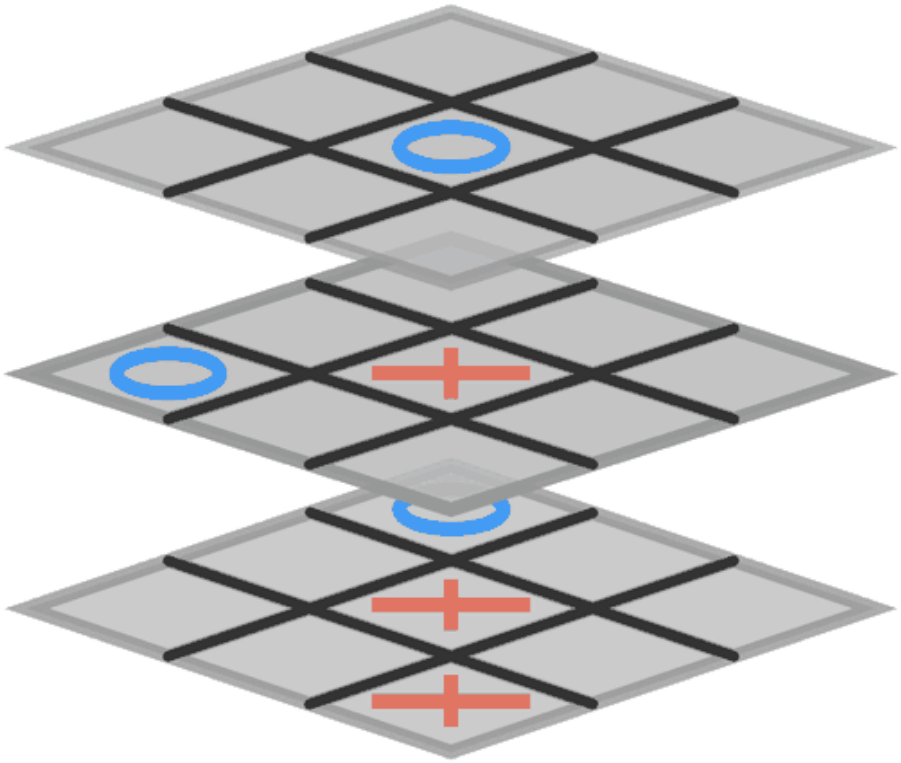
Score: 96/100
Good job!



3D-Tic-Tac-Toe

Group 30
Prepared by
Hasan Ali, Umer Qazi, Syed Raza and Claudio Jimenez
***for use in* CS 440**
at the
University of Illinois Chicago

Fall 2020

Table of Contents

Table of figures

I. Project Description

1. Project Overview

3D Tic-Tac-Toe is an extension of the century old game of Tic-Tac-Toe. This game will be available on both mobile software (iOS and Android) and as a website for PC users. Each player will create an account with a unique user ID. This will track their progress in the game and their achievements. The new game will have all the rules from the original game and some more. 3D Tic-Tac-Toe will consist of up-to 4 players with every player trying to win the game before others. The game surface will be in a cube shape resulting in 6 total surfaces where the game can be played. A player needs to win on 2 total surfaces to be considered a winner. The game will have powerups/charms to make the game more exciting. It will be a free-to-play game with players earning or purchasing different power ups and avatars for their gameplay.

2. The Purpose of the Project

2.1 The User Business or Background of the Project Effort

There are experienced fans of the original Tic-Tac-Toe game worldwide. Many adapt to the simple rules of the game and have an interest in gaming at a competitive level to where the stakes are higher and more than just 2 players can be involved. The online gaming business is a very large market that modern gamers all have become a part of. Allowing gamers to play with each other via multiplayer online gaming opens up the potential of the game to reach great heights. There is always a demand for new games and 'fan favorite' games that have a modern twist to it. We intend to sell this directly to gamers as the gaming industry is a large and growing one.

2.2 Goals of the Project

Our 2 primary goals are challenge fans of the original Tic-Tac-Toe game and allow multiple players to be involved. We want to bring a fresh new look to all the fans of the original game with new features, powerups, and perks as you continue your journey on this game. This will create an enjoyable multiplayer atmosphere that is popular in today's modern games as well as including perks that allow your player to improve and be rewarded. Allowing players to have these new features will allow the players to play the refreshed 3D version for a much longer time span than the original 2d tic-tac-toe. Another goal of ours is to increase game platform subscribers which will ultimately increase game revenues. This is vital to having 3D tic-tac-toe being a success as our game design should retain the platform subscribers and increase over time.

2.3 Measurement

We will measure the use of our product via feedback and reviews given by our customers. As we plan to have our game implemented in the apple store, google play store, and on our website, users will be able to provide real time feedback on the game to express how they are feeling. These ratings out of 5 stars will be a key indicator if our users are truly enjoying the game and all of its features. This data will allow us to determine what features of the game the players like and don't like, as well as informing

us on any missing content they would like to have included in the game. After completing a game, the users will see a small pop up on the side that will ask if the wish to provide feedback/review on the game. We will know when these metrics will be achieved when reviews increase, and player subscribers increase by 10%.

3. The Scope of the Work

The "work" is the communication and interaction between the game and various services related to the user.

3.1 The Current Situation

The client, Board Games 2.0 INC, would like to explore this specific field as they see it having potential. This would build upon their series of n-D games such as 4-DChess and 4-DCheckers with 3-D Tic Tac Toe.

As for the current situation for the game itself, it is simply the basic game that can be played anywhere with pencil and paper. Its origination can be dated back to 1300 BC and yet it has stood against the test of time with children today playing the same game. While there are a multitude of individuals that have created this game, it cannot be attributed to any one single entity.

3.2 The Context of the Work

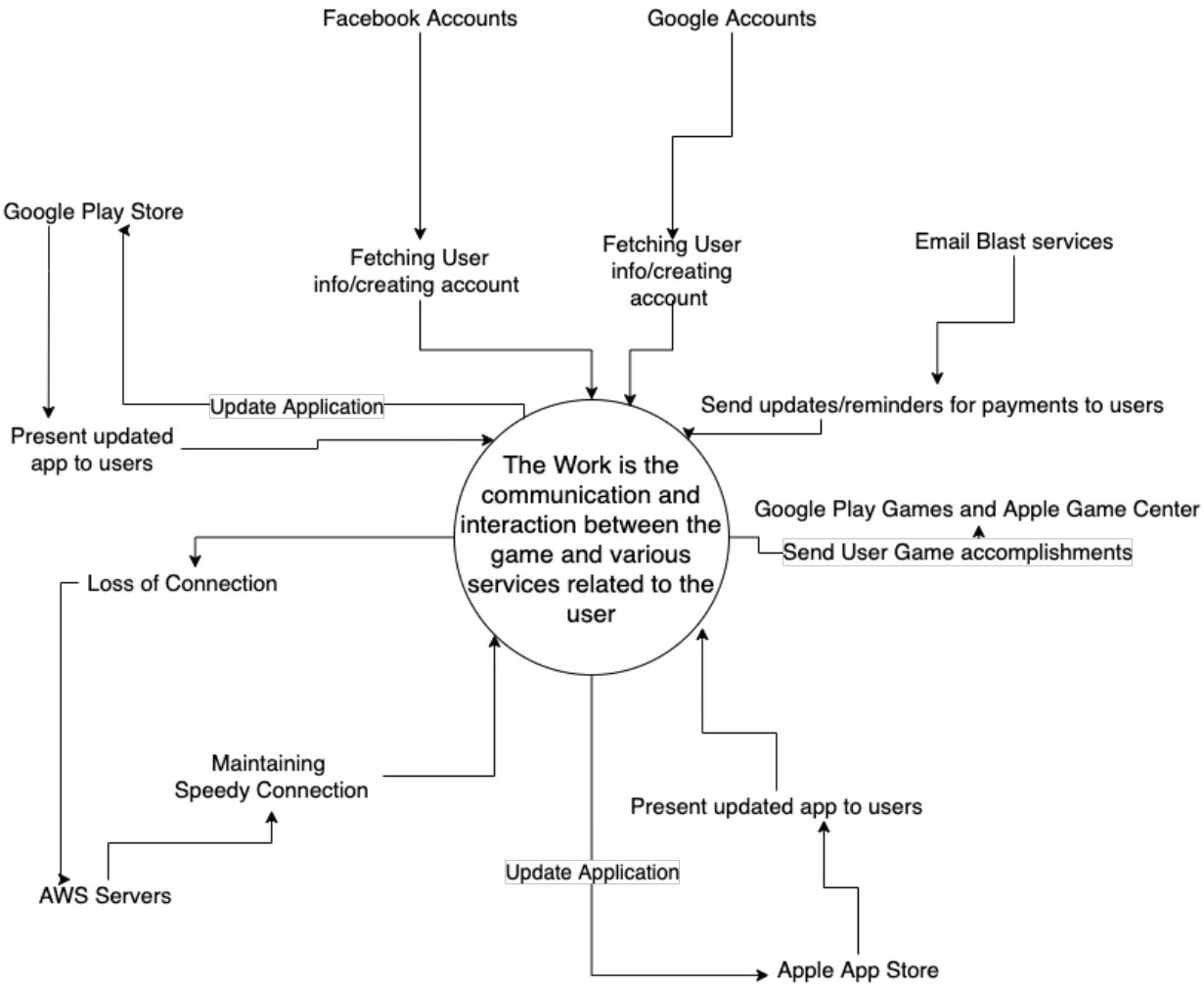


Figure 1 THE CONTEXT OF THE WORK DIAGRAM

3.3 Work Partitioning

(a) Business Event List

Event Name	***Input and Output***	***Summary***
1. Update application for Android	*Publish updated app to Google Developer Console (out)* *Allow users to update app on their phone(in)*	*Update the code with new features, to account for policy change etc. Need to tell Google to update for all*
2. Update application for Apple	*Publish updated app to Apple Developer (out)* *Allow users to update app on their phone(in)*	*Update the code with new features, to account for policy change etc. Need to tell Apple to update for all users*
3. Facebook integration with user's Accounts	*User profile info (in)*	*Whenever a user signs in with a Facebook account, will have to re-direct to their website*
4. Email Service blast periodically	*Send reminder emails for payments/new features (in)*	*Keep users updated with new features for the game and entice them back in*
5. Problem with AWS servers	*Report problem of servers (out)* *Fix problem/find out who's problem it is*	*Servers needs to stay up 24/7 so problem needs to be fixed quickly*
6. Google integration with user's Accounts	*User profile info (in))*	*Whenever a user signs in with a Google account, will have to re-direct to their website.*
7. Update game info to Facebook account for user to share	*Send game accomplishments to Facebook (out)*	*Give the user the ability to share their accomplishments through Facebook*
8. Google Play Games and Apple Game Center update to new rankings	*Send game accomplishments to Google/Apple (out)*	*Using Google/Apple services, users can easily compare their accomplishments with their friends*

3.4 Competing Products

There are different attempts at a game of 3-D Tic Tac Toe but none as elaborate as this one. One simple search comes with multiple examples of these attempts.

However, they are played on various websites with multiple other games and no features for collaboration/competition. It is not a complete experience for the user, rather they are all very basic in nature with subpar graphics. Through various communication features, customization features, and Artificial Intelligence we will be revamping the whole game with a new set of rules while retaining the simplistic foundational nature which made it so enjoyable in the first place. It will give users the experience of a full-fledged game while also challenging themselves to get better at it.

4. The Scope of the Product

3D Tic-Tac-Toe is a modified version of the traditional 2D game, the proposed product addressed the need of the online game business to be constantly innovating and developing new products, of providing entertainment, bringing the player's community together and encouraging interaction and competition among gamers. 3D Tic-tac-toe would achieve this work by creating a game platform (online, android and iOS Apps) with a front-end interface where user interact with the system and a back-end server that connects the players, runs the games logic, provides AI players, keeps gamers scores, their account info, accomplishments, provides messenger services and system/game help. System maintenance and updates to the game will performed on a schedule by the system managers.

4.1 Scenario Diagram

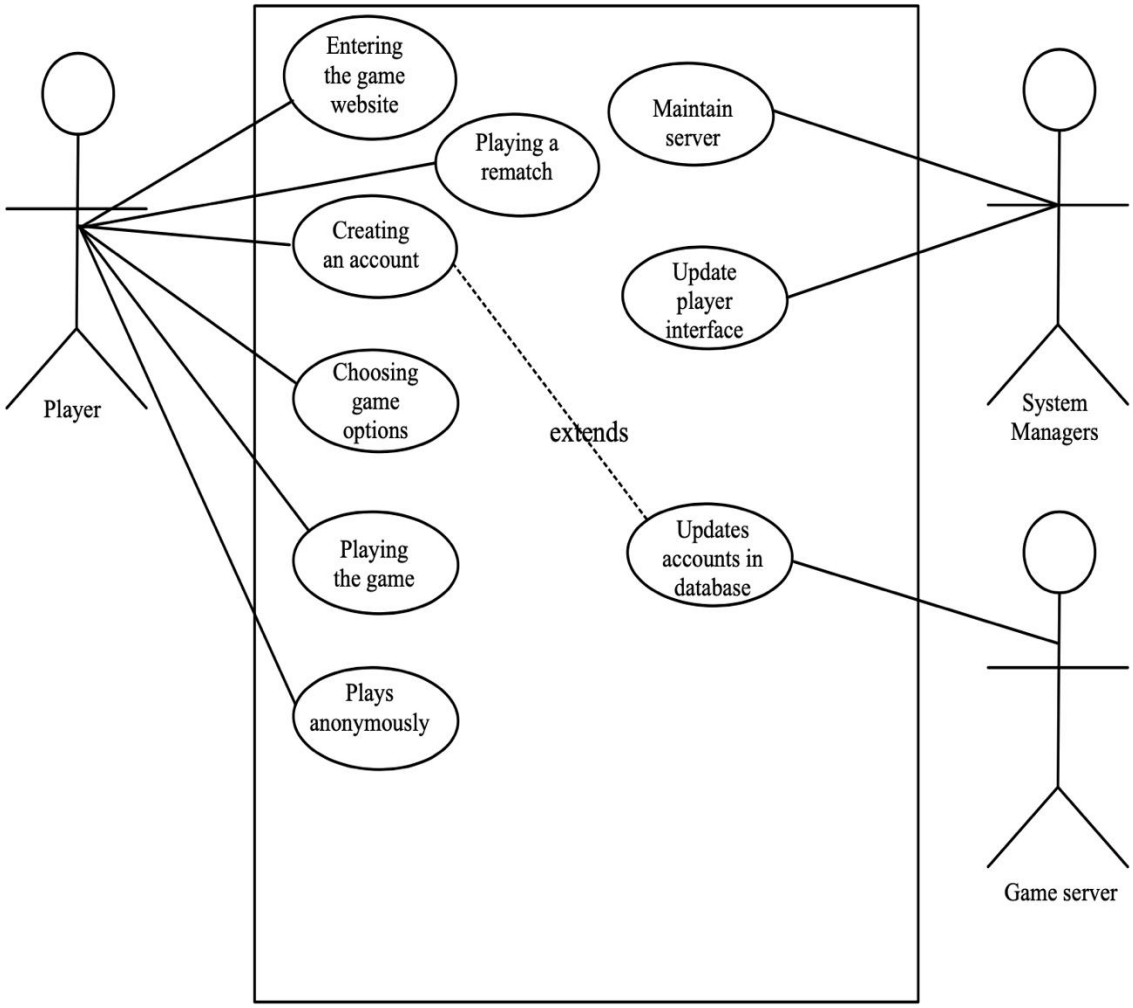


Figure 2 SCENARIO DIAGRAM

4.2 Product Scenario List

Scenarios	Users
Pre-Game Scenarios	
1. Entering the game website	Players
2. Creating an account	Players
3. Playing anonymously (without account)	Players
4. Choosing game options	Players
Gameplay Scenarios	
5. Playing the game	Players
6. Playing a rematch	Players
Post-game Scenarios	
7. View profile and accomplishments 8. Have ability to share on social media	Players Players, System managers
System Maintenance Scenarios	
9. Maintain server	System managers
10. Update player interface	System managers

4.3 Individual Product Scenarios

1. **Entering the game website:** If Joe want to play online, he will open a web browser and enter the game website address via email invitation or advertised link. The system will serve a welcoming page to Joe with the options to create an account, login or play anonymously. Joe will know that he is in the right page by seeing the company logo and game name.

2. **Creating an account:** On the welcoming screen Mary will click on create account button, she will be redirected to another screen where she would be asked to enter her name, chosen username, phone number for account recovery(optional), password, credit card information and billing address(optional) Once Mary enters her information, she will have the option of creating a game avatar. She would have the choice of customizing the avatar's face, body and attire. The game server will add the account to the database and serve Mary with the game lobby screen.

3. **Playing anonymously:** On the welcoming screen Tom will click on the play anonymously button. The server will redirected Tom to the game lobby screen. The game server will automatically generate a username for Tom that he can only use

once. Tom would not be able to take advantage of any of achievements or point system of the game. Tom is ready to set his desired game options and start playing.

4. **Choosing game options:** Hana would be presented with the game lobby screen, she would have to set the following options,

   i. Type of game: Hana could play against other players, or against AI. She could choose to play a ranked tournament, or a practice run.

   ii. She could play with up to 4 players.

   iii. Hana must choose a difficulty level between easy, medium, hard or expert. Only on ranked games Hana will earn points. The higher the difficulty the more point the Hana earns. The system would place Hana with other players with similar points and ranking.

   Once Hana is finished setting these options, she would be able to start the game by pressing the start game button. The system would start looking for other players depending on her chosen settings and would display to Hana an approximate wait time. Hana would decide if she wants to wait the displayed time or change her options. Once the system finds players with similar settings as Hana, they will be bundle into game group and sent to the gameplay screen.

5. **Playing the game:** Harman is now in the gameplay screen. Harman will see a 3D game board were players can enter their move via the player interface, his score, ranking and power ups. Harman will play according to the game rules. Once Harman wins, he would get his awarded points, badges and avatar options (he would only have these awards if he has a game account) If the system declares a game tie, there would be no awarded points or badges. In both cases Harman would be redirected to the game lobby to start another game or quit.

6. **Playing a rematch:** After the system declares a winner or a tie game Gloria would be redirected to the game lobby screen where she and the other players are bundle as a game group, here Gloria can choose to quit the game group or choose a rematch. If Gloria decides to quit the game but the other players decide to stay the new game would only have the remainder players without Gloria. If she decides to stay in the group and play a rematch, her game group would be redirected to the gameplay screen for another match.

7. **View profile and accomplishments:** On the game lobby webpage Patricia will have a “player profile” button. Once she clicks the button, she would be presented with a pop-up window showing her players statistics composed of the number of games she played, the games she won and lost, her accumulated point, award badges, remaining power charms and game ranking based on her points. Patricia would also see her user id and customized avatar.

8. **Maintain the server:** Hugo a system manager would be required to do a system maintenance once a month. He would schedule the maintenance based in user’s

utilization of the system. After Hugo identifies the times where system utilization is below 3% he will schedule a systemwide maintenance. Tina, a player, would be notified via pop up message next time she logs in about the maintenance. During the maintenance no user would be able to log in or play the game. System downtime should not exceed more than two hours and Hugo should schedule it at least 3 weeks in advance. Hugo would be able to maintain the server through the back-end software that he can accesses through the web portal with his username and password.

9. **Update player interface:** Enrique, a system manager, would be able to update the game interface through the back-end software. He will log in the system through the web portal with his username and password. Enrique would then change layouts, colors, graphics and the game UI according to developers' specifications. Once the updates are completed Hugo will click on a button to update the server. There should not be system downtime for these types of updates. After Hugo completes the updates, they are pushed to the front-end user interface. The next time Tina, a player, logs into the system she would be presented with the new updated UI.

5. Stakeholders

5.1 The Client

The client is our parent company Board Games 2.0 INC (refers as the company). The company is privately own with a market capitalization of 10 million dollars and around 250 employees. The company's main goal is to develop new games based on old board games augmenting their capabilities with new rules, graphics, gamer interaction and award systems. The company has had great success with 4D chess and 4D checkers growing their business and market capitalization by 100% within the past 5 years. The company develops and releases new games in a two-year schedule. 3D tic-tac-toe is one of the projects for the 2020-2022 development run.

5.2 The Customer

3D tic-tac-toe is intended for the online gamers market. In 2010 there were around 1 billion gamers worldwide and in 2020 there are around 4.5 billion. This vast market is composed of persons ranging from the ages of 15 to 54. This year revenues projection for the industry are around $196 billion. Demand for online board games has increased 10-fold in the past 5 years. These customers are looking for new and traditional games that have been modified to play online. The most popular and successful games are those that provide the gamers community with competitions and interactions among players as well as rewards and tournaments systems. 3D tic-tac-toe will have all these features and would be updated according to new game trends and technologies.

5.3 Hands-On Users of the Product

The main users of the product are our current game platform subscribers and the online gamer community at large. Their main responsibility is to play the game at least once a week, participate in tournaments and interact with other players. These actors must

be journeyman of online gaming, be in the 10 to 55 years old bracket and be novice of traditional tic-tac-toe.

5.4 Maintenance Users and Service Technicians

System operators and managers will be in charge of deploying the game in the different platforms, of maintaining and updating the system. They are in-house actors with a deep and extensive understanding of the system front and back end (full stack developers) They have worked in close contact with the developing team to understand de game and its functionality. They must have experience in games and product software development.

5.5 Other Stakeholders

Sponsor: The company will be sponsoring and promoting the product on their other game platforms.

Testers: There would be a beta testing phase were current users of our other game products will be invited to try and test the game as well as to give feedback.

Business analysts: Our in-house business analyst and game researchers will be providing support and feedback on our competitors and current products success.

Marketing experts: Our in-house marketing will also use the product to advertise new games and product releases.

Legal experts: Our in-house legal team will provide support on the privacy laws and legal framework in the countries where the product will be released.

5.6 User Participation

Our research and business analyst team will provide the developers with inside on our current product successes and failures, what has work vs. what has not. They will also provide the knowledge of the latest trends in game features and player satisfaction. They will provide answers to our questions, what do players look for during gameplay? what kind of interactions? This input will help the development team in their UI and features design.

Current users of our game platforms will provide through our beta releases knowledge of interface prototyping and as well as usability requirements. They will be required to play the game for at least 20 hours.

5.7 Priorities Assigned to Users

We identified the following priorities in our users,

- Key users: Current company game platform users and subscribers. Currently our platform has 5.1 million players worldwide.
- Secondary users: The online gamers community, system owners and managers.
- Unimportant users: Any user that is not familiar with online gaming. Users that play the game once or have no knowledge of how to play tic-tac-toe.

6. Mandated Constraints

These are the non-negotiable necessities of the project in question.

6.1 Solution Constraints

Description: The product shall be available on the Google Play Store, Apple App Store, as well as a website

Rationale: There are different cohorts of people in all three of those categories and we want this available to as many people as possible so we will target all three audiences.

Fit Rationale: Consistent design between the two applications on the respective mobile devices, and the same theme applied to the Website. Performance should be similar regardless of the device.

Description: The product shall have full cross-platform capabilities. No matter what device the user is using, they should be able to communicate and interact with other players.

Rationale: One of the most enticing features is to play with friends, so different devices should not be the restricting factor for friends to play with each other

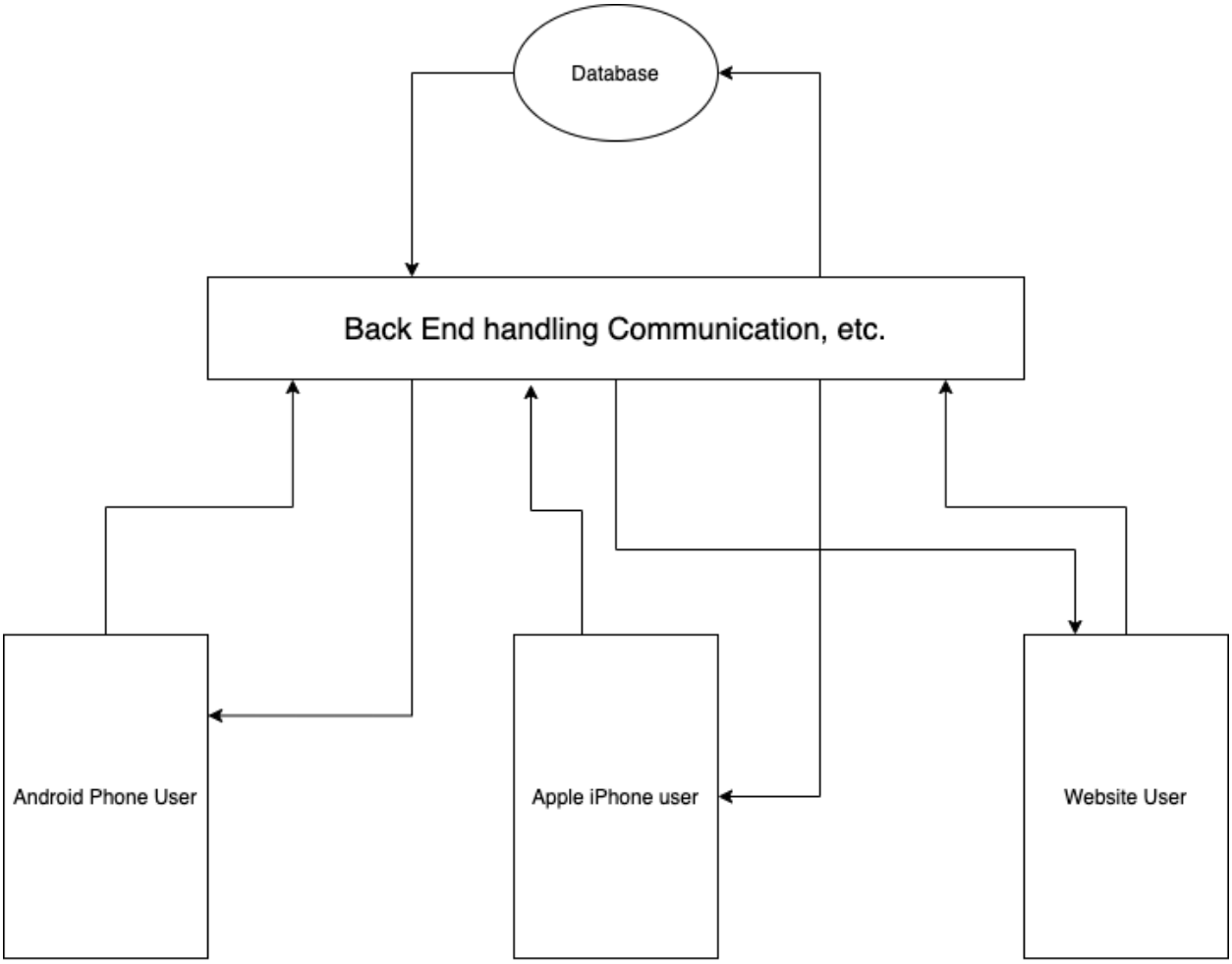
Fit Rationale: Fast connectivity between the three options

Description: The product shall have full cross-platform capabilities. No matter what device the user is using, they should be able to communicate and interact with other players.

Rationale: One of the most enticing features is to play with friends, so different devices should not be the restricting factor for friends to play with each other

Fit Rationale: Fast connectivity between the three options needs to be the standard, it is unlikely that they will keep on coming back if they have connection issues.

6.2 Implementation Environment of the Current System



Users on the different platforms are allowed to communicate with each other and play games together. This requires all of the respective Hardwares to be able to access a common server and share data between each other.

FIGURE 3 COMPLETE SYSTEM

6.3 Partner or Collaborative Applications

- Option to export game data to a Microsoft Excel sheet which will show improvement by means of a graph. This allows the user to have access to their statistics and do what they like with it

- Option to share information with the user's available applications

    - Share scores on social media

    - Automated text messages to invite other people to play

- Email blast service to keep data organized and send account updates/features update/policy changes
- Work with 4D Chess and 4D Checkers to promote 3D Tic Tac Toe

6.4 Off-the-Shelf Software

Google Play Store – It is a necessity for any business creating an application to publish on the Google Play store to give easy access to all the Android users.

Apple App Store – It is a necessity for any business creating an application to publish on the Apple App store to give easy access to all the iPhone users.

AWS Servers – Maintaining user data in a safe place will require the user of Amazon Web Services. They will be retrieved and kept safe through Amazon's services.

6.5 Anticipated Workplace Environment

- The messaging of other players should not interfere with the gameplay of a user.
- Any reminders by email should not be frequent such that users think they are spam.
- The product must have simple big and easy directions to help the user navigate
- The product must start off easy to encourage the user to get better at the game
- The product must be visually appealing as that is involved in customer retention

6.6 Schedule Constraints

- Window of opportunity: (to take into account)
    - With Corona forcing everybody to spend unprecedented amounts of time in front of a screen, this opportunity needs to be availed. However, the product that is released needs to be top notch and high quality to match the Client's needs
- Deadlines:
    - Initial marketing start: Jan 2021
    - Program sent to testers for functionality: Sept. 2021
    - Hype Buildup/People reviewing the game: Feb 2022
    - Release: Jun 2022

- Notes:
    - Important to release in a Summer since the younger demographic will be off from school and have a lot of free time.
    - Need to create a "Hype Train" with marketing, working with influencers, etc. before final release.

6.7 Budget Constraints

Total Budget: $300,000

- Top notch Artificial Intelligent robot. Will need to create a robot based on various models that understands the game and can play it at varying difficulty levels.
- Graphic Designing for logos, and animations for applications/website
- Android and iOS programmers respectively
- Website programmer and designer
- Maintaining server and databases

7. Naming Conventions and Definitions

7.1 Definitions of Key Terms

Users/players are use interchangeable in use cases and they refer to any person that plays the game.

System owners are persons that maintain or update the system from the back-end server.

All terms are used in this document:

Avatar: A player's customizable character

Rank: The position in hierarchy based on a players wins

Points: Rewarded by completing certain milestones during games

Badges: Rewards granted after reaching a certain number of points

Power-charms: Special abilities a player can use during the game to get an advantage over other competitors

Google Accounts: Any account created through the Gmail service by the user which they have the option to connect to their User account on our platform.

Facebook Accounts: Any account created through the Facebook service by the user which they have the option to connect to their User account on our platform.

Google Play Store: The App store that the application will be published on for users that have Android Phones.

Apple App Store: The App store that the application will be published on for users that have iPhones.

AWS Servers: Servers to hold our data

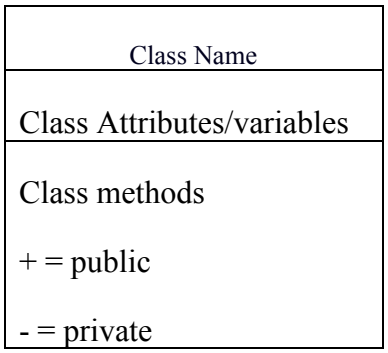
Google Play Games: The social application provided by Google, giving users the ability to compare scores with their friends and get accomplishments.

Apple Game Center: The social application provided by Apple, giving users the ability to compare scores with their friends and get accomplishments.

Gameserve: The management and maintenance software use for the back end of 3D Tic-tac-toe.

7.2 UML and Other Notation Used in This Document

We will be using the standard UML notations as described in UML Distilled by Martin Fowler. An example UML will be shown below via the Medium article where all basic UML class diagrams elements will be present.



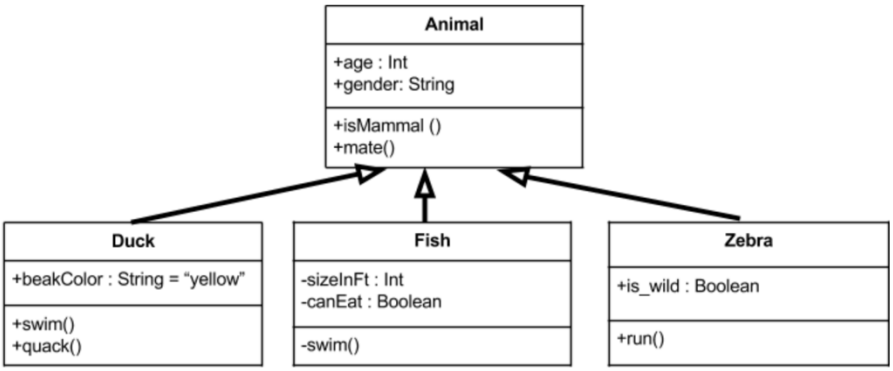


Figure 4 FIGURE UML EXAMPLE

7.3 Data Dictionary for Any Included Models

Database: Table to store the perks and power-charms that are available in the game. The database will also have the user requests when they perform a certain move in the game.

Cloud: The cloud will be where vital information of the user will reside. This includes usernames/email, password, guest account, and in game purchases. We will also store the information of the latest clothes on the avatar and their gaming record of wins, losses, etc.

8. Relevant Facts and Assumptions

8.1 Facts

Requirements:

The user must agree to the terms and services which will clearly let the user know what type of data is collected. The data collected will not have their personal data except their name, email, and password. We will collect the data that are required to run the game. For example, score, wins, in game purchases, and their rank.

Audience:

There are around 4.5 billion gamers in the world with that number steadily increasing at around 5.9% each year.

Game Economy:

The global market of online gaming is more than 165 million dollars and will reach around 300 million dollars by 2025. Globally, 44 percent of gamers reported they would subscribe to a service, with gamers ages 18-45, aspiring professionals and expert gamers the most interested. Too high pricing and poor performance are the top obstacles to subscribing. Our game is catered towards young audience as well so making it free-to-play with buy ins in the game is more beneficial for the earnings and profits our investors will get.

8.2 Assumptions

We are hoping to work on a subscription-based model of the game at the same time which will have everything from the free-to-play model and some more additional features for the enthusiasts. The game will be primarily in English with future works on different languages. We expect the game to be played on Android 5.0 and iOS 13 at the least. 4G, 5G, and Wi-Fi compatibility will be required to connect to our servers to play the game.

II. Requirements

1. Product Use Cases

1.1 Use Case Diagrams

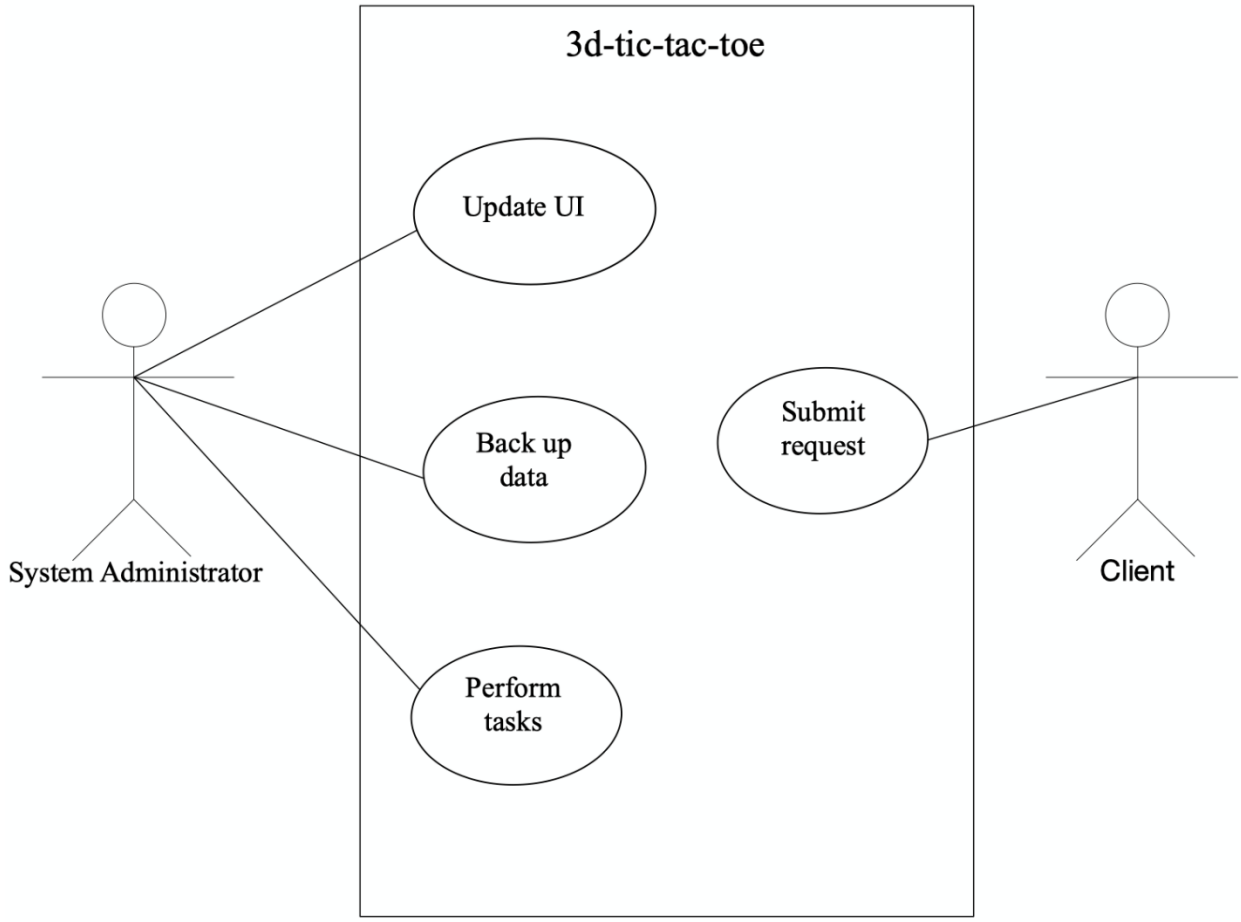


Figure 5 MAINTENANCE USE CASE DIAGRAM

1.2 Product Use Case List

Maintenance
1. Submit maintenance request 2. Backup Data 3. Update UI 4. Perform maintenance tasks

1.3 Individual Product Use Cases

Use case ID: maintenace1 Name: Submit maintenance request

pre-conditions: Actor must be login into the system management software “Gameserve”

post-conditions: Actor will have a maintenance request confirmation number

Initiated by: Client

Triggering Event: Client presses the “request service maintenance button” on Gameserve

Additional Actors: System Administrator

Sequence of Events:

1. Client requests system maintenance on Gameserve
2. System responds by providing the client with a maintenance summary form.
3. Client fills the form with the current maintenance issue and his personal details.
4. Client submits the maintenance form.
5. System conducts data validation.
6. Systems displays submitted form and requests confirmation.
7. Client accepts confirmation
8. System provides client with a confirmation number and email contact.
9. Systems sends submitted request to the system administrators email inbox.

Alternatives: Request can be done on paper if system fails.

Exceptions: Data validation fails

Use case ID: maintenenace2 Name: Backup data

pre-conditions: System must be in maintenance mode. Data must be refreshed. Actor must be login into Gameserve and have administrator privileges

post-conditions: latest user accounts data would be stored in the backup server. Actor will have a confirmation number.

Initiated by: System administrator

Triggering Event: administrator presses the Backup option on Gameserve main menu.

Additional Actors: None

Sequence of Events:

1. Actor chooses backup option.
2. System displays menu with backup options.
3. Actor chooses backup data option from menu.
4. System performs a backup configuration test.
5. System displays tests performed and passed.
6. System requests confirmation from actor.
7. Actor provides backup confirmation.
8. Systems performs backup and displays progress.
9. Systems provides backup confirmation and an associated number.

Alternatives: Perform on schedule

Exceptions: At least one backup configuration test fails

Use case ID: maintenace3 Name: Update UI

pre-conditions: Actor must have new UI layout file. Actor must be login into Gameserve and have administrator privileges. System must be in maintenance mode. Actor has “update game configurations menu” open.

post-conditions: Game UI has changed and is updated with new file. Actor will have confirmation number.

Initiated by: System administrator

Triggering Event: Administrator chooses update game layout from menu.

Additional Actors: None

Sequence of Events:

10. Administrator chooses update UI from menu.

    11. System performs update system check.

    12. System responds by asking the actor to choose the location of the file.

13. Administrator provides location of file.

    14. Systems validates file.

    15. System performs update and displays its progress.

    16. System provides update confirmation and corresponding number

Alternatives: Perform on schedule

Exceptions: Update or system check fails. Wrong file.

Use case ID: maintenance4 Name: Perform maintenance tasks

pre-conditions: Maintenance must be scheduled for current period. System must be in maintenance mode. Actor must be login into Gameserve and have administrator privileges. Actor must be in the “perform system maintenance” menu option.

post-conditions: The maintenance record file will be updated. Actor will have confirmation number.

Initiated by: System Administrator

Triggering Event: Actor selects perform system maintenance in Gameserve main menu.

Additional Actors: None

Sequence of Events:

17. Actor selects perform system maintenance in main menu.

    18. System responds with a menu with all the available maintenance tasks.

19. Actors select maintenance task/s to perform.

    20. System responds with a list of maintenance task/s to perform.

    21. System displays confirmation message.

22. Actors provides confirmation.

    23. Systems responds by performing requested task/s.

    24. System displays progress.

    25. System updates the maintenance record file.

    26. System provides confirmation and associated number.

Alternatives: Perform on schedule.

Exceptions: At least one maintenance task fails.

1.4 Product Use Case List

Pre-Game
1. User Changes Avatars
2. User Chooses Game Mode
3. User Invites friends to a personal lobby
4. User queues with friends
5. User reports an unfriendly user

Use case ID: PreGame1 Name: User Changes Avatars

pre-conditions: The game should open up to show the players' character and should have an option to customize it on the main screen before jumping into a game.

post-conditions: Game UI has changed and is updated with the new avatar. The player will now see the new avatar on the screen.

Initiated by: user

Triggering Event: Administrator chooses update avatar from menu.

Additional Actors: None

Sequence of Events:

1. User chooses to update the avatar from the menu.
2. System performs update avatar check.
3. System responds by asking the actor to choose the location of the file.
4. Administrator provides location of file.
5. Systems validates files.
6. System performs updates and displays its progress.
7. System provides update confirmation and corresponding number.

Alternatives: N/A

Exceptions: Update or system check fails. Wrong file.

Use case ID: PreGame2 Name: User Chooses Game Mode

pre-conditions: The game should show game options on the game screen for the player to decide the preferred game mode to play.

post-conditions: The player is then connected to the appropriate server for that chosen game mode.

Initiated by: user

Triggering Event: Administrator chooses game mode from game screen menu.

Additional Actors: None

Sequence of Events:

1. User chooses to select the game mode from the menu.
2. System performs update game mode check.
3. System responds by asking the actor to choose the location of the file.
4. Administrator provides location of file.
5. Systems validates files.
6. System performs updates and displays its progress.
7. System provides update confirmation and connects the player to the corresponding active server for that game mode.

Alternatives: There is no active server for that game mode and a new server needs to be activated.

Exceptions: Update or system check fails. Connection to the server times out.

Use case ID: PreGame3 Name: User Invites friends to a personal lobby

pre-conditions: The game should allow each player to invite their friends to their server to play among themselves.

post-conditions: A personal game between the friends is set up and all of the users are connected to the same server.

Initiated by: user

Triggering Event: Administrator connects all the users to the same server.

Additional Actors: None

Sequence of Events:

1. Users choose to invite their friends.

2. System responds by asking the actor to retrieve the username for the player(s) that are invited.

3. Administrator provides username(s).

4. Systems validates username(s).

5. System connects the invited player to the player who invited..

6. System then connects the players to an active server.

Alternatives: The invited player declines the request.

Exceptions: No Active servers. The connection times out.

Use case ID: PreGame4 Name: User queues with friends

pre-conditions: The users queue as a party for the game.

post-conditions: The party finds an active server and connects to it.

Initiated by: users

Triggering Event: Administrator looks for an open server with space for players to connect them to it.

Additional Actors: None

Sequence of Events:

1. Administrator queues the players as a party.
2. System responds by requesting the usernames of the players queueing.
3. Administrator provides usernames.
4. Systems validates usernames.
5. System searches for an open server with space for players.
6. System connects the party to an open server with other players.

Alternatives: No open server so the players are connected to a personal server with no extra players.

Exceptions: No Active servers. The connection times out.

Use case ID: PreGame5 Name: User reports an unfriendly user

pre-conditions: The game should have an option for users to report an unfriendly user for toxicity or harassment.

post-conditions: The reported player is then investigated for their behavior.

Initiated by: user

Triggering Event: Administrator writes a report for the reported player.

Additional Actors: None

Sequence of Events:

1. User chooses a player for their behavior.
2. System responds by asking the username of the user that reported and the user who is reported.
3. Administrator provides the usernames.
4. Systems validates usernames.
5. System writes a report for the incident case.
6. System provides the report to the legal admin in charge of harassment control.

Alternatives: N/A

Exceptions: Update or system check fails. Wrong username.

1.5 Product Use Case List

Gameplay
1. User starts game 2. User achieves milestone

3. User sends message
4. User Uses powerup
5. User wins

1.6 Individual Product Use Cases

Use case ID: gameplay1 Name: User starts game

pre-conditions: Must be connected to the internet and have correct game mode chosen

post-conditions: The user has started the game with friends

Initiated by: The user

Triggering Event: The user queues up game mode and presses 'Play'

Additional Actors: The system and the database where the user accounts will be stored.

Sequence of Events:

1. The user opens the game.
2. The system asks if the user wants to invite friends.
    - The user could play in public lobby as well
3. The user chooses to queue up game and press 'play'.
4. The user enters a loading screen where the system sets up game mode with friends or with other players in a public lobby.
5. The system checks the database to check that the credentials are correct.
6. The user has entered the game.

Alternatives: N/A

Exceptions: The user has internet issues and cannot start game

Use case ID: gameplay2 Name: User Achieves Milestone

pre-conditions: User is in the game and has met the requirement for a certain milestone.

post-conditions: Milestone award is bestowed to user.

Initiated by: The system

Triggering Event: The user completes a designated task that is pre-set by the game system

Additional Actors: The database where the milestone requirements will be stored.

Sequence of Events:

1. The user is playing the game.
2. The user has completed a certain challenging task.
3. The system runs a test case to ensure the milestone task is met.
4. If yes, the system displays a textbox with ‘Congratulations, you’ve achieved a milestone!’.

Alternatives: N/A

Exceptions: N/A

Use case ID: gameplay3 Name: User Sends Message

pre-conditions: User is in a game mode and has selected the chat box.

post-conditions: The user sent message/emoji to other players

Initiated by: The user

Triggering Event: The user clicks on chat box during a game

Additional Actors: The system and the database where the emoji options will be stored

Sequence of Events:

1. The user clicks on the chat box.
2. The system opens chat box and provides an option for text message or emoji.
3. The user selects chat option
    - Or user selects emoji option
4. The system sends message from server to client for other players to receive the message.

Alternatives: N/A

Exceptions: N/A

Use case ID: gameplay4 Name: User uses power charm

pre-conditions: User has enough points to initiate power charm ability

post-conditions: The user uses power charm ability

Initiated by: The user

Triggering Event: Earning points and/or use case, gameplay2

Additional Actors: N/A

Sequence of Events:

1. The system displays all the information about the points the player has and when they can use a certain power-charm move.
2. The user plays game and earns points as they progress through the game.
3. The user earns enough points to use a power charm.
4. The use selects the Power charm option and selects the power charm they would like to use.
5. System communicates with database to initiate the power charm.
6. The power charm animation takes over the display of the users in the game.
7. The power charm is complete, and the users continue to play the game.

Alternatives: User chooses alternate power-charm

Exceptions: A separate user initiates a power-charm at the same time that blocks out the first user's power charm

Use case ID: gameplay5 Name: User wins game

pre-conditions: User has connected 3 X/O's on all 3 dimensions on the cube game layout

post-conditions: The user wins and game ends

Initiated by: The user

Triggering Event: Connecting 3 X/O's on all 3 dimensions on the cube game layout

Additional Actors: N/A

Sequence of Events:

1. The user connects 3 X/O's on all 3 dimensions on the cube game layout.
2. The system checks to make sure the player has met the winning criteria.
3. The system displays a winning animation for the users in the game lobby.
4. The system awards points to each player based on their game performance.
5. All users exit the game mode and return to the main lobby.

Alternatives: N/A

Exceptions: A power charm is used prior to user winning.

1.7 Product Use Case List

Postgame
1. Update scoreboard 2. Share game summary 3. Chat with other players 4. Request for playing again 5. Quit the game

Use Case ID: PostGame1 Name: Update Scoreboard

Pre-conditions: Completion of Game

Post-conditions: Display of Scoreboard

Initiated by: Player

Triggering Event: Last move completed by a specific player

Additional Actors: Other Players

Sequence of Events:

1. Player makes the last move
    1. System concludes that the game has been completed and there is a clear winner based on the points earned and the current board
    2. System posts board for all the players to see
2. Players see complete game statistics regarding the game played
    1. System presents options to play again, quit, or share
3. Players make their respective decision.

Alternatives: Potential tie, System needs to be prepared for that possibility

Exceptions: Player leaving last second, System needs to ensure that that does not interfere with the display for other players

Use Case ID: PostGame2 **Name:** Share game summary

Pre-conditions: Game must be completed

Post-conditions: Game summary shared on the social media platform of the players choice

Initiated by: Player

Triggering Event: Player selecting option to share game summary

Additional Actors: System, Usage of API from Social Media sites

Sequence of Events:

1. Player's game end
    1. System presents the various next actions along with the scoreboard
2. Player selects share option for specific social media
        1. System opens up pop-up for the respective social media
        2. If not logged in, allows the user to login. Otherwise:
3. User types message and subsequently shares on aforementioned social media

1. System returns user back to game with the options to play again, quit, or share

Alternatives: N/A

Exceptions: Connection between System and Social Media fails, System needs to inform user and return to menu

Use Case ID: PostGame3 **Name:** Chatting with other players

Pre-conditions: Game ended already

Post-conditions: Ideas communicated between the players

Initiated by: Various players

Triggering Event: Players communicating via voice or text using the System's features

Additional Actors: System

Sequence of Events:

1. Game is completed
    1. System presents scoreboard
    2. System ensures that the players are still able to communicate with each other since they are in the same lobby.
2. Players can discuss how the game went
    1. System allows for communication to continue
    2. System also has the various options available for the players

Alternatives: Only text or only voice communication for a player that requested

Exceptions: Microphone of a user not working when they are attempting to communicate, mention that to them.

Use Case ID: PostGame4 **Name:** Request for playing again

Pre-conditions: Game is completed

Post-conditions: Another game starts

Initiated by: Player

Triggering Event: Player selects option to play again

Additional Actors: Other players, System

Sequence of Events:

1. Game is completed
    1. System presents scoreboard
2. Player selects option to replay
    1. System enters Player's avatar in a lobby, waiting for other players
    2. Once System has enough players to start a game
3. Main Host has option to start game or wait for more players
    1. System starts game once told to

Alternatives: Player does not choose to play again, Player choose to play then changes mind

Exceptions: If player starts another game but leaves, an AI will play in his stead.

Use Case ID: PostGame5 **Name:** Quit the game

Pre-conditions: Game ends

Post-conditions: Player has left this specific game

Initiated by: Player

Triggering Event: Player selects option to quit

Additional Actors: System

Sequence of Events:

1. Game is completed

1. System presents scoreboard

2. Player selects option to quit game
    1. System takes player back to main menu, presenting options to go into another lobby
3. Player either leaves the game entirely or joins another lobby

Alternatives: Player abruptly exits by closing tab or application

Exceptions: Player stays in

2. Functional Requirements

Requirement #: M1
Requirement Type: Functional
Event/use case#: maintenace1

Description: The system shall provide the user with the means to enter data.
Rationale: The user needs to provide information about an action he wants done.
Originator: Client
Fit Criterion: The request is received by the appropriate actors and matches the provided data.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: M2
Requirement Type: Functional
Event/use case#: maintenance1, maintenance3, maintenance4

Description: The system shall provide a means for the user to confirm an action.
Rationale: To get confirmation from the user on an action the system will perform
Originator: Client, system administrator
Fit Criterion: The system executes de action the user confirmed.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high

History: Created 10/23/20

Requirement #: M3
Requirement Type: Functional
Event/use case#: maintenace1

Description: The system shall be able to route messages among users.
Rationale: System users need to communicate their requests.
Originator: Client, System administrator
Fit Criterion: message N sent by user1 to user N, is the same message N user N received from user1.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: M4
Requirement Type: Functional
Event/use case#: maintenance1, maintenance4

Description: The system shall provide feedback to the user about his input.
Rationale: The user needs to know if the data he inputs into the system is correct.
Originator: Client, system administrator
Fit Criterion: The data feedback provided by the system matches the data the user provided

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: M5
Requirement Type: Functional
Event/use case#: maintenace2, maintenance3, maintenance4

Description: The system shall provide the user with feedback on performed tasks and their progress.
Rationale: The user needs to know what tasks the system is performing and their progress.
Originator: Client
Fit Criterion: The user is aware what task/s the system is performing and their progress.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: Medium
History: Created 10/23/20

Requirement #: M6
Requirement Type: Functional
Event/use case#: maintenace2

Description: The system shall be able to write its data contents into the backup server.
Rationale: The systems needs to be able to recoup from loss or corrupted data.
Originator: System Administrator
Fit Criterion: Current system data matches data on backup server.

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: High
History: Created 10/23/20

Requirement #: M7
Requirement Type: Functional
Event/use case#: maintenace3

Description: The system shall be able to read files provided by the user.
Rationale: The user needs to maintain and update the system.
Originator: System Administrator
Fit Criterion: System is able to read and process the file format provided by the user.

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: High
History: Created 10/23/20

Requirement #: PG1
Requirement Type: Functional
Event/use case#: PreGame1
Description: The system shall have a mechanism to allow the user to change avatars
Rationale: The user should be allowed to change their avatar.
Originator: Client, system administrator
Fit Criterion: Ensure that any user has the ability to change the avatar to their preferred. It could be an avatar they earned or bought.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/31/20

Requirement #: PG2
Requirement Type: Functional
Event/use case#: PreGame3, PostGame3, Gameplay3
Description: The system must allow for in-game communication during the party server
Rationale: Data sent by one user should be visible to everyone else in the party
Originator: Client, system administrator
Fit Criterion: Data entered by the user matches harassment policy.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG3
Requirement Type: Functional
Event/use case#: PreGame5
Description: The system must have a mechanism for users to report other unfriendly players.
Rationale: Multiple options must be presented so the user can decide what type of harassment was done.
Originator: user, system administrator
Fit Criterion: Users have full ability to send as many reports as they like.
Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP1
Requirement Type: Functional
Event/use case#: gameplay1

Description: The system shall be let the user know once a session has begun
Rationale: Once the user initiates a game, the game start. Thus the System must have a way of mentioning that end to the user
Originator: System administrator
Fit Criterion: To test this, all possible ways of starting a game must be covered for and must lead to the same result

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP2
Requirement Type: Functional
Event/use case#: gameplay2

Description: The system shall be let the user know once a milestone is achieved
Rationale: During the game, the user can use certain power charm moves after attaining certain amount of milestones and achievements. This will help the user be motivated to buy new items and continue to play the game to strive for a new item.
Originator: System administrator
Fit Criterion: To test this, all possible ways of earning a milestone must be covered for and must lead to the same result

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP3
Requirement Type: Functional
Event/use case#: gameplay4

Description: The system shall be let the user know when a power charm feature is being used
Rationale: Once a user initiates his power charm, the other users in the game lobby will need to strategically plan their next moves to compete for a win
Originator: System administrator
Fit Criterion: To test this, all possible ways of initiating a power charm feature must be covered for and must lead to the same result

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG1
Requirement Type: Functional
Event/use case#: PostGame1, gameplay5

Description: The system shall be let the user know once a user has won and the session has been completed
Rationale: Once a game starts, it must end. Thus the System must have a way of mentioning that end to the user
Originator: System administrator

Fit Criterion: To test this, all possible ways of ending a game must be covered for and must lead to the same result

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG2
Requirement Type: Functional
Event/use case#: PostGame2

Description: The system shall have a mechanism to allow the user to share their scores
Rationale: The user should be allowed to share their results with their friends on all the major Social Media platforms
Originator: Client, system administrator
Fit Criterion: Ensure that any user has the ability to connect to a valid account on other websites and successfully make a post of some kind on it.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG3
Requirement Type: Functional
Event/use case#: PostGame3, gameplay3

Description: The system must allow for in-game communication
Rationale: Data entered by the user must be consistent
Originator: Client, system administrator
Fit Criterion: Data entered by the user matches data specifications

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG4
Requirement Type: Functional
Event/use case#: PostGame4, PostGame5

Description: The system must have a mechanism for users to choose an option
Rationale: Multiple options must be presented so the user can decide
Originator: user, system administrator
Fit Criterion: User has full ability to make any decision they like

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

3. Data Requirements

ID# - DT1

Description: Game should be able to know the amount of currency a player possesses.

Rationale: A player should be able to know this data from the game so that he can buy units, items, etc.

Fit Criterion: N/A

Acceptance Tests: TDT1

ID# - DT2

Description: Game should contain a user's play history.

Rationale: A player should be able to know this data from the game so that he can see stats and improve on future games.

Fit Criterion: If it can accurately count the currency overtime this requirement should be met.

Acceptance Tests: TDT2

4. Performance Requirements

4.1 Speed and Latency Requirements

Requirement # P1

Description: Interactions between users and the product may have a maximum lag of 2 seconds.

Rationale: Any interaction that goes from the user to the product to another user may have the aforementioned maximum lag. Anything beyond that will not be acceptable since the users should not feel like they are waiting on the product.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

Requirement # P2

Description: Any text interaction strictly between two users may have a maximum lag of ¾ second.

Rationale: If users wish to communicate via text using the product, they must be able to in a near-instant fashion.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

Requirement # P3

Description: Any interaction between the product and any outside source may have a maximum 3 second lag from the product's side

Rationale: Since connecting with other sources upon the user's request may take a bit more time, need to accommodate for that.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

4.2 Precision or Accuracy Requirements

Requirement # P4

Description: All monetary transactions will be accurate to two decimal places where applicable. Where not applicable, it will be accurate to the one's place or no place's.

Rationale: Any monetary-related services must be as accurate as possible since the user should not feel like they are being scammed.

Fit Criterion: To test, all currencies throughout the world will be cross-referenced.

Acceptance Tests: TP2

Requirement # P5

Description: All services related to time in the product may be accurate up to seconds.

Rationale: Users should know the seconds remaining on any time-sensitive function of the product so that they can do what they feel is best for their situation.

Fit Criterion: N/A

Acceptance Tests: TP3

4.3 Capacity Requirements

Requirement # P6

Description: The product shall accommodate for a maximum of 5000 users at peak time, from 5:00pm to 10:pm. At other periods there will be a maximum of 1000 users.

Rationale: For the peak times that gamers are on, the product should be able to handle every single person that would like to play. Not doing so may turn people away from the product itself.

Fit Criterion: Listed above

Acceptance Tests: TP4

Requirement # P7

Description: During service times, the product shall accommodate a maximum of 3500 users at peak time and a maximum of 750 at other periods.

Rationale: Some functionality may not be as robust and working fully since there may be a service repair going on, nevertheless there needs to be a specific amount of users that should still be able to play

Fit Criterion: Listed above

Acceptance Tests: TP4

5. Dependability Requirements

5.1 Reliability Requirements

Requirement # D1

Description: The product shall not have a system wide failure except in very specific cases.

Rationale: The product must be available 24/7 to ensure that anybody around the globe that would like to play it can play it. Only exceptions may be an expected shutdown for fixes or security breaches; users must be required to know beforehand

Fit Criterion: To test, the product must be able to go through a shutdown and return back safely in a specified amount of time.

Acceptance Tests: TD1

Requirement # D2

Description: The product shall not lose any data of any user in the event of any failure.

Rationale: User Data is one of the most precious commodities for the product and users should not feel like they have lost progress to any degree, otherwise they might not be inclined to come back.

Fit Criterion: To test, ensure that the data that is safely backed up in a secure location can be used to bring back potential lost data.

Acceptance Tests: TD2

5.2 Availability Requirements

Requirement # D3

Description: Any failure of the product, expected or unexpected, may not go on beyond 5 hours

Rationale: Whether the reasons for the shutdown is fixing a bug or a security breach, the problem should be solved within 5 hours. Regular users should not have to wait longer than that to play a game.

Fit Criterion: To test, all teams associated with the shutdown should have well communication between each other about the tasks at hand and the related deadlines.

Acceptance Tests: TM&S2

Requirement # D4

Description: The product shall be available for use by the users for 700/720 hours in a month, to account for any fixes or security breaches.

Rationale: The 20 hours allows for leeway for the developers to accomplish their job and fix any problems that may arise. At the same time, the product will be available for 97% of the time in a month which is a good percentage that needs to be required.

Fit Criterion: N/A

Acceptance Tests: TD3

5.3 Robustness or Fault-Tolerance Requirements

Requirement # D5

Description: The product should allow the user to be able to play for at least ten minutes even with connectivity failure, given that the user is playing against a non-user.

Rationale: The user should not be disrupted if they are using the product for as long as possible. This gives them some more time to interact with the product even though they may not necessarily be connected.

Fit Criterion: Ensure that the product has the ability to load up actions on a local level in the case of a network failure.

Acceptance Tests: TD4

Requirement # D6

Description: The product shall have an auto-save function so that the user does not lose any of their progress in the event of a fault, either on the product's side or the user's side.

Rationale: The user should never feel like they will lose all of their progress and accomplishments if something goes awry. Thus, the product needs to account for all of the possibilities.

Fit Criterion: To test, the system should be forcefully shutdown in a variety of scenarios. In every case, the data of the user shall not be corrupted.

Acceptance Tests: TD2

Requirement # D7

Description: The product shall decrease the level of graphics in the event of a server overload with too many users. This will help in the speed aspect of the users the are currently playing.

Rationale: Even if the product is getting the its limits in terms of the data that it can handle, it shall not decrease in the functionality and gameplay of the user. They shall continue playing the game in the same expected speeds, albeit with lower graphics.

Fit Criterion: To test, ensure that the system can automatically switch users to a lower set of graphics without human intervention, once a specific command has been fulfilled.

Acceptance Tests: TD4

5.4 Safety-Critical Requirements

This requirement does not apply to our project.

6. Maintainability and Supportability Requirements

6.1 Maintenance Requirements

Requirement #M&S1

Description: The system administrator must complete maintenance requests within 72 hours.

Rationale: Maintenance request need to be implemented to keep the system from entering an erroneous state.

Fit Criterion: Upon receiving a maintenance request the system administrator completes the request within 72 hours

Acceptance Tests: TM&S1

Requirement # M&S2

Description: The hardware technician must be able to add new servers within 48 hours.

Rationale: As the user population grows the system will need more computational and storage capacity.

Fit Criterion: New hardware is installed within 48 hours after receiving the request

Acceptance Tests: TM&S1

Requirement # M&S3

Description: The system cannot be down for more than 5 hours.

Rationale: System needs to be accessed by the users to prevent user dissatisfaction.

Fit Criterion: System operation is restored within 5 hours after going down.

Acceptance Tests: TM&S2

Requirement # M&S4

Description: System maintenance must be performed by a system administrator.

Rationale: Due to the systems complexity only authorized and trained personnel can perform maintenance tasks

Fit Criterion: System maintenance operations are performed by authorized users.

Acceptance Tests: TM&S3

6.2 Supportability Requirements

Requirement # M&S5

Description: The system must provide support about its user interface.

Rationale: New users need to know how to interact with the system.

Fit Criterion: User must be able to navigate the system interface after accessing the systems interface tutorial.

Acceptance Tests: TM&S4

Requirement # M&S6

Description: The system must provide help on how to play the game.

Rationale: New users need to know how to play the game.

Fit Criterion: Upon accessing the game help a new user must be able to win the game tutorial match.

Acceptance Tests: TM&S4

Requirement # M&S7

Description: Game and system help must be developed by the development team.

Rationale: Support must be clear, concise and developed by knowledgeable actors.

Fit Criterion: After accessing the game or system support a user will be able to play the game and navigate the UI for at least two times without help.

Acceptance Tests: TM&S4

6.3 Adaptability Requirements

Requirement # M&S8

Description: The game must be able to run on Google Chrome, Mozilla Firefox, Opera and Apple Safari.

Rationale: The game must be playable on the most popular web browsers in order to reach the highest number of customers

Fit Criterion: Game operates properly and without errors on the above mention web browsers.

Acceptance Tests: TM&S5

Requirement # M&S9

Description: The game must be able to run on Apple iOS and Android.

Rationale: The game must be playable on the most popular mobile platforms in order to reach the highest number of customers.

Fit Criterion: Game operates properly and without errors on the above mobile platforms.

Acceptance Tests: TM&S6

Requirement # M&S10

Description: The game will be distributed through the Apple App Store and Google Store platforms.

Rationale: The game needs to reach the highest number of customers possible.

Fit Criterion: Game will be available for download on the above mention App stores.

Acceptance Tests: TM&S7

6.4 Scalability or Extensibility Requirements

Requirement # M&S11

Description: The system must be able to handle our current number of companywide subscribers.

Rationale: Any of our current customers may want to play the game.

Fit Criterion: Server capacity must be able to handle our current number of server requests.

Acceptance Tests: TM&S8

Requirement # M&S12

Description: The system must be able to process multiple requests from different platforms.

Rationale: All system user must be able to access and play the game

Fit Criterion: Multiple users in different platforms must be able to play at the same time.

Acceptance Tests: TM&S8

6.5 Longevity Requirements

Requirement # M&S13

Description: The system should last for 7 years

Rationale: Our current research shows that game usage and life last for about 5 years

Fit Criterion: The system must be operational for 7 years.

Acceptance Tests: TM&S9

7. Security Requirements

7.1 Access Requirements

ID# - S1

Description: Only the developers should be able to see crash reports and logistics. However, users will be notified of crashes.

Rationale: Blocking users from accessing crash reports ensures confidentiality of the application and background processes. Furthermore, it ensures that bugs won't be altered to pose as an attack to the application.

Fit Criterion: General users should not be able to access crash reports.

Acceptance Tests: TS1

ID# - S2

Description: Only developers will have access to the source-code. General users should not have access the code or be able to read or write the source code.

Rationale: Blocking users from accessing the source code is crucial for preventing any hacking and to make sure the application's functionality does not change.

Fit Criterion: General users should not be able to access the source code

Acceptance Tests: TS2

7.2 Integrity Requirements

ID# - S3

Description: The system should not allow users to access other users' sensitive information.

Rationale: Keeping the users' sensitive information confidential is crucial as the user could have payment information and personal information as well.

Fit Criterion: General users should not have access to other users' personal information.

Acceptance Tests: TS3

7.3 Privacy Requirements

ID# - S4

Description: The application should have a terms and agreement page before the user uses the application that informs users of all privacy standards.

Rationale: Users will feel more comfortable using an application that lists the user's rights regarding their own security and privacy rights.

Fit Criterion: The application should display the terms and agreement page prior to starting the application for the first time and update the user on any changes.

Acceptance Tests: TS4

ID# - S5

Description: Users should be able to terminate their account along with any information stored in that account profile.

Rationale: Users should be treated fairly with their personal information being erased when they choose to delete their profile. This ensures that data will not be leaked or accessed after termination.

Fit Criterion: User should be allowed to delete their account along with the database containing the account should not contain account info after deletion.

Acceptance Tests: TS5

7.4 Audit Requirements

ID# - S6

Description: The application should track user data regarding ratings, multiplayer activity, and any update to the database.

Rationale: Keeping and tracking user data will allow developers to improve the application based on the user's activity.

Fit Criterion: Tracking user data will be allowed

Acceptance Tests: TS6

7.5 Immunity Requirements

ID# - S7

Description: The system should be hosted by third party security software. These should be constantly backing up and keeping information encrypted.

Rationale: With constant feedback from security firms, the application will be free of any unauthorized packets being set, and any other malware from corrupting files.

Fit Criterion: There will be multiple rounds of tests placed such as OWASP. These tests will have multiple iterations to protect personal data.

Acceptance Tests: TS7

8. Usability and Humanity Requirements

8.1 Ease of Use Requirements

Requirement # U&H1

Description: The game should open up to show the players' character and should have an option to customize it on the main screen before jumping into a game.

Rationale: It is important to have the ease of changing the avatars and customizing it since our game economy is dependent on players buying into the game. We need to make sure it is easy to work

Fit Criterion: It should open the customization screen in the first ten seconds. The customization screen needs to be visible before a game starts

Acceptance Tests: TU&H1

Requirement # U&H2

Description: The game description must have a "how to" button on the main screen for users who want to refresh their game rules

Rationale: Since 3D Tic-Tic-Toe is an extension of a simple historic game, we need to make sure that the rules are set and are easy to understand.

Fit Criterion: A 10-year-old player should be able to understand the rules of the game when first read.

Acceptance Tests: TU&H2

8.2 Personalization and Internationalization Requirements

Requirement # U&H3

Description: The game must be available in multiple languages but most importantly in English, Korean, Japanese, Mandarin, Arabic, and the European languages.

Rationale: Since it is going to be an international game, we need to have multiple languages in the game. Especially the "how to" section of the game and settings.

Fit Criterion: It should be at least translated into the top 5 most common languages by release and continues to add languages for our international player base.

Acceptance Tests: TU&H3

8.3 Learning Requirements

Requirement # U&H4

Description: When first opening the app and making an account, the user goes through a tutorial in a language of their choice.

Rationale: 3D Tic-Tac-Toe has important features and rules that each player needs to understand to fully enjoy the game.

Fit Criterion: The game should put the new player with 3 bots as a practice game and an option to redo the game until the player feels confident in their skills and understanding of the game.

Acceptance Tests: TU&H4

8.4 Understandability and Politeness Requirements

Requirement # U&H5

Description: The game should be easy for a middle schooler.

Rationale: Since most middle schoolers now have a smartphone and are connected to the internet. We need our game to be easy enough for middle schoolers to play.

Fit Criterion: The game will be tested with middle and high schoolers.

Acceptance Tests: TU&H5

Requirement # U&H6

Description: The game will have symbols and buttons on the interactive game screen while playing.

Rationale: Since we will have powers ups and other buttons during the game, we need them to be in a visually pleasing environment for players to understand and utilize while playing.

Fit Criterion: All important buttons, power ups, and timers must be clear to the user when playing.

Acceptance Tests: TU&H6

8.5 Accessibility Requirements

Requirement # U&H7

Description: The game must be accessible to players with disabilities.

Rationale: We want our players from every background to have a good experience while playing.

Fit Criterion: We will test the game with players with Special needs and mild autism to make sure we reach every potential player.

Acceptance Tests: TU&H7

8.6 User Documentation Requirements

Requirement # U&H8

Description: The developers must put the “terms of use” and “in game harassment policy” documents in the game.

Rationale: Since we want our users to have fun playing the game and take out any possible toxicity and harassment, we will make all of our players sign each policy.

Fit Criterion: The developers must make a "complaint form" for players to report other players for toxic behavior and any kind of harassment. We also need to make sure that harassment policy is updated every 3 months, so 4 times in a year.

Acceptance Tests: TU&H8

8.7 Training Requirements

Requirement # U&H9

Description: For training, the user will learn about the app and its features through a step-by-step tutorial.

Rationale: The app has crucial features that would be handled best when the player has learned the app's functions. Otherwise, it would be hard to understand if the user does not follow the tutorial.

Fit Criterion: The user must be trained to be competent enough to use the interactive buttons and are able to have fun.

Acceptance Tests: TU&H9

9. Look and Feel Requirements

9.1 Appearance Requirements

Requirement # LaF1

Description: The product shall adhere to all the branding standards of Board Games 2.0 INC.

Rationale: All of the products must look the same, thus this product must as well.

Fit Criterion: To test, design mockups will first be checked by the design department of Board Games 2.0 INC. to ensure that all standards are being to a certain extent.

Acceptance Tests: TLaF1

Requirement # LaF2

Description: The product shall have different colors to be associated with the rank of a particular user.

Rationale: The different colors will aspire the lower-ranked users to get to the higher rank. It will also establish color schemes that will be changing but consistent for all users.

Fit Criterion: To test, all colors will be cycled through to make sure they are still appealing and do not break any of the client's aforementioned standards.

Acceptance Tests: TLaF1

Requirement # LaF3

Description: The product shall have a bright and inviting appearance

Rationale: The intended audience is the general public and so they should feel like they're wanted when using said product.

Fit Criterion: To test, the product must be cross-referenced with other popular games with a bright color scheme.

Acceptance Tests: TLaF2

9.2 Style Requirements

Requirement # LaF4

Description: The product shall incite a willingness in the user to play the game

Rationale: The product shall create a desire in the user that may be mysterious through text, images, etc. which will want the users to try the game out.

Fit Criterion: To test, 65 percent of initial customers shall feel a desire to play the game

Acceptance Tests: TLaF2

Requirement # LaF5

Description: The game shall be encouraging to first time players helping them through the motions

Rationale: The product shall have a low bar for entry and making progress in the beginning paired with encouraging words.

Fit Criterion: To test, a sample of customers will be shown the game and given the option to play or leave the game. 70% of them must at least try to play one time.

Acceptance Tests: TLaF2

Requirement # LaF6

Description: The product shall make the user feel a sense of accomplishment with their wins.

Rationale: The product should allow the user to feel like the problem is easy enough to solve but also hard enough to create that sense of accomplishment inside of them.

Fit Criterion: To test, the initial customers will be surveyed how they feel and 70% of them must feel like they solved something that was non-trivial but doable.

Acceptance Tests: TLaF2

10. Operational and Environmental Requirements

10.1 Expected Physical Environment

Requirement #O&E1

Description: The product should be able to work on any smartphone or desktop environment.

Rationale: The application does require an internet connection for multiplayer play, so it's important that a smartphone or desktop is required .

Fit Criterion: User should be able to play anywhere in their home or where their device is functioning with internet connection .

Acceptance Tests: TO&E1

10.2 Requirements for Interfacing with Adjacent Systems

Requirement #O&E2

Description: The application should work with the operating system of different mobile devices/computers as well as be at an efficient ram and data storage.

Rationale: The game should be practical by not taking up too much storage on the user's device. We would like to keep it efficient and compact while not slowing down any performance.

Fit Criterion: The product should work on all mobile devices computers at an efficient rate.

Acceptance Tests: TO&E2

10.3 Productization Requirements

Requirement #O&E3

Description: The product shall be distributed free of charge as an application package on the Google and Apple App Stores for download.

Rationale: Product needs to be distributed widely for better market penetration.

Fit Criterion: Game App should would be available on the mobile app stores.

Acceptance Tests: TM&S7

Requirement #O&E4

Description: The product shall be downloadable and installed by the user of any mobile platform without recourse.

Rationale: Product should be easy to download and install.

Fit Criterion: Game App is downloaded to a mobile device from the device app store.

Acceptance Tests: TM&S7

10.4 Release Requirements

Requirement #O&E5

Description: An update version of the product shall be released once a year.

Rationale: The product needs to be competitive and up to date with current game trends and technologies.

Fit Criterion: Developers will work on bug fixes and incorporate new features into the updated version. Budget should not exceed 50K per release.

Acceptance Tests: TM&S7

11. Cultural and Political Requirements

11.1 Cultural Requirements

This requirement does not apply to our project.

11.2 Political Requirements

This requirement does not apply to our project.

Requirement # C&P2

This requirement does not apply to our project.

11.3 Legal compliance Requirements

Requirement # L1

Description: System must be FDA compliant

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

Requirement # L2

Description: System must be compliant with local privacy laws

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

Requirement # L3

Description: System must be compliant with local accounting laws.

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

Requirement # L4

Description: System must be compliant with local communications and networking laws.

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

11.4 Standards Requirements

This requirement does not apply to our project.

12. Requirements Acceptance Tests

Test# TDT1

Description: Cross-reference all money related info on the website to accommodate for all. Checks user profile data and sees if currency and bought items are accurate

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TDT2

Description: Access to database. Checks users' play history to see if accurate

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TDT1

Description: Speed test with a timed deadline. Will be tested 1000 times in one setting to get the most accurate results. Will be tracked by the system itself.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TP1

Description: Speed test with a timed deadline. Will be tested 1000 times in one setting to get the most accurate results. Will be tracked by the system itself.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TP2

Description: Cross-reference all money related info on the website to accommodate for all.

Test Passed: All currencies are accurate to two decimal spots, if possible.

Test Fail: If there exists a currency that is not accurate

Test# TP3

Description: Use VPNs to check the time in various places around the globe to compare with a third-party universal clock.

Test Passed: Times are accurate to ten seconds, give or take.

Test Fail: Times are not accurate for more than a ten-second leeway.

Test# TP4

Description: Amount of approximate resources for a specific capacity will be calculated and then they will be used up, forcefully. System should be able to run even with said amount of resources used up.

Test Passed: System is still 95% functional with resources used up.

Test Fail: System functionality for said limit is below 95%.

Test# TD1

Description: Ability of System to be able to be run with minimal resources

Test Passed: System running even with 25% or less resources down

Test Failed: System not able to run with 25% or less resources down

Test# TD2

Description: Ability of System to retain user data

Test Passed: No issue causes any loss of user data

Test Failed: Issue causes loss of user data

Test# TD3

Description: Practice runs will be done for the actors that will help do the fixes

Test Passed: All problems can be solved in the given timeframe

Test Failed: Problems cannot be solved and result in user's inability to use product

Test# TD4

Description: Test for system to run despite any potential issues. The problems will be replicated to see if the system is sustainable in said conditions.

Test Passed: System can run with 95% functionality

Test Failed: System cannot run with 95% functionality

Test#TM&S1

Description: Maintenance request with a timed deadline. Request will be submitted to the system. The system will log a on submit time stamp. Once completed the actor will submit an on complete time stamp.

Test Passed: On complete time stamp is within 72 hours of on submit time stamp

Test Fail: On complete time stamp is **not** within 72 hours of on submit time stamp

Test#TM&S2

Description: System down for maintenance. The system is set to maintenance mode and the system administrator logs the time.

Test Passed: System is operational within 5 hours of the logged time.

Test Fail: System is **not** operational after 5 hours of the logged time.

Test#TM&S3

Description: Access to system administration. Authorized and unauthorized users will try to log into the Gameserve system.

Test Passed: Users with system administrator access can log in. Users without access credentials cannot.

Test Fail: At least one user without system administrator credentials logs into Gameserve.

Test#TM&S4

Description: UI and game play tutorials. User watches the tutorials once.

Test Passed: Users is able to play a game, login, logout, access the game lobby and see its profile.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby and see its profile.

Test#TM&S5

Description: Test browser compatibility. User will enter the game webpage in the following web browsers Google Chrome, Mozilla Firefox, Opera and Apple Safari.

Test Passed: Users is able to play a game, login, logout, access the game lobby, see and set its profile on the browser.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby, see and set its profile on the browser.

Test#TM&S6

Description: Test mobile platform compatibility. User will open the downloaded mobile application on their mobile device.

Test Passed: Users is able to play a game, login, logout, access the game lobby, see and set its profile on their device.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby and see and set its profile on their device.

Test#TM&S7

Description: Test mobile platform availability. User will download the application to their mobile device from the mobile app stores.

Test Passed: Users is able to download the app and play a game, login, logout, access the game lobby, see and set its profile on their device.

Test Fail: Users is **not** able to download the app and play a game, login, logout, access the game lobby, see and set its profile on their device.

Test#TM&S8

Description: Test system capacity. System will be operating with our current number of company subscribers.

Test Passed: Systems operates at peak performance (CPU usage <= 88%, Memory usage <=60%, network capacity <=90%)

Test Fail: System does **not** meet all criteria for peak performance.

Test#TM&S9

Description: Test system lifespan.

Test Passed: System is operational 98% of the time for 7 years after launch day.

Test Fail: System is **not** operational 98% of the time for 7 years after launch day.

Test# TS1

Description: Create a non-threatening crash report. Send user notification of crash.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TS2

Description: Make a new account and check if it is possible to access source code.

Test Passed: User is not able to see source code.

Test Fail: User is able to see source code.

Test# TS3

Description: Make a new account and see if user is able to see their friends' private information.

Test Passed: User is not able to see any private information.

Test Fail: User is able to see private information.

Test# TS4

Description: Create new account and see if terms of service page pops up. If user declines terms of service, user should not be allowed to proceed.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TS5

Description: Users should have a delete account button to delete the account. Check if database contains the deleted user after deletion. If user still appears in database, fail.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TS6

Description: Check a players multiplayer activity such as time spent on app, gameplay activity, and ratings. Fail if information can't be found on.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TS7

Description: Attempt to breach security, if breach successful, fail.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TU&H1

Description: User Changes the avatar

Test Passed: The avatar successfully changes permanently.

Test Fail: The avatar stays as the old one.

Test# TU&H2

Description: User clicks on the "how to" button on main screen.

Test Passed: The game retrieves the how to document and offers the player for a practice game with bots.

Test Fail: The game does not retrieve the document.

Test# TU&H3

Description: User changes the overall language.

Test Passed: The game refreshes and shows the requested language and set that language as the preferred language for future games.

Test Fail: The game stays on English or does not update to the next language.

Test# TU&H4

Description: User goes through a training tutorial.

Test Passed: The account is made successfully, and first thing showed is mandatory training.

Test Fail: The training module is not visible to the player.

Test# TU&H5

Description: The game is easy to understand.

Test Passed: Middle schoolers can understand and progress in the game.

Test Fail: It is difficult for kids to understand and will not play.

Test# TU&H6

Description: User utilizes the interactive buttons during gameplay.

Test Passed: The buttons are easy to understand and easy to use.

Test Fail: The buttons are confusing and are clustered so they are difficult to use.

Test# TU&H7

Description: Every user should be welcomed.

Test Passed: Users with mild disabilities are also able to understand the game.

Test Fail: The users with disability do not understand the game.

Test# TU&H8

Description: The "term of use" and "in game harassment policy" documents.

Test Passed: The documents must be signed before a user finish making their account and must follow the policy.

Test Fail: Players do not sign the policy documents and they are able to play the game.

Test#TL1

Description: Legal Test. Product documentation and legal summary is submitted to legal team.

Test Passed: Legal team approves the document

Test Fail: Legal team does not approve the document

Test# TLaF1

Description: Check with Board Games 2.0 INC to ensure that most of the standards are being fulfilled regarding any designs or appearance

Test passed: 95% of the standards are being fulfilled.

Test Failed: less than 95% of the standards are being fulfilled.

Test# TLaF2

Description: The product shall test with a certain group of customers where they will go through the whole process of playing 5 games. Then they will be asked to fill a survey to feel their willingness to play the game again.

Test passed: 87% or more of the customers say they would like to play it again.

Test Failed: less than 87% of the customers say they would like to play it again.

Test# TO&E1

Description: Download game on iOS device, android device, and desktop device.

Test Passed: Able to play game on all 3 platforms .

Test Fail: Unable to play on any of the 3 platforms.

Test# TO&E2

Description: Check storage required per device, app should not take up more than 1.3 GB.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

	Requirements																												
Test	P1	P2	P3	P4	P5	P6	P7	O&E3	O&E4	M&S1	M&S2	M&S3	M&S4	M&S5	M&S6	M&S7	M&S8	M&S9	M&S10	M&S11	M&S12	M&S13	L1	L2	L3	L4	P8	D1	D2
TD1																											x		
TD2																												x	
TD3																													x
TD4																													x
TP1	x	x	x																										
TP2				x																									
TP3					x																								
TP4						x	x																			x			
TM&S1										x	x																		
TM&S2												x																	
TM&S3													x																
TM&S4														x	x	x													
TM&S5																	x												
TM&S6																		x											
TM&S7								x	x										x										
TM&S8																				x	x								
TM&S9																						x							
TL1																							x	x	x	x			

Figure 6 Requirements – Test Correspondence Summary

(CONTINUED)

12.1 Requirements – Test Correspondence Summary

	LaF1	LaF2	LaF3	LaF4	LaF5	LaF6		
LaF1	x	x						
LaF2			x	x	x	x		

Figure 7 - Requirements - Acceptance Tests Correspondence

III. Design

1. Design Goals

Graphics: The 3D rendering in this game must be very smooth and crisp as there are many people that appreciate that, and it is one of the biggest aspects regarding a specific game. However, at the same time, it needs to be available to the masses and should not only be restricted to those with a superior graphics card.

Colors: The colors should be uniform across all platforms, so the same experience is present for the user, whether they are on the web or on mobile. The colors should ideally be on the brighter side as that gets the attention of people and they are more inclined to use the application with popping colors.

Simplicity: The whole program from top to bottom should be very easy to use and get right into it. There can be an optional tutorial to walk beginner players through the game but along with that, even if somebody doesn't want to, the buttons and their purpose should be very clear. Also, the URL should be relatively simple so that it remains in the minds of the users.

Connectivity: There should be seamless communication between users on web and mobile. The system should be designed well enough that all different devices are treated the same in terms of connecting with each other. Thus, somebody on the Web version should have no problem playing with somebody that is on the mobile version.

2. Current System Design

There is no current system design.

3. Proposed System Design

3.1 Initial System Analysis and Class Identification

After grammatical analysis of our project description, we have identified the following main classes:

- Game class
- User class
- Scoreboard class
- Player class
- Server class
- Administrator class
- Avatar class
- Powerup class
- Gameboard class

We have developed the following UML class diagram:

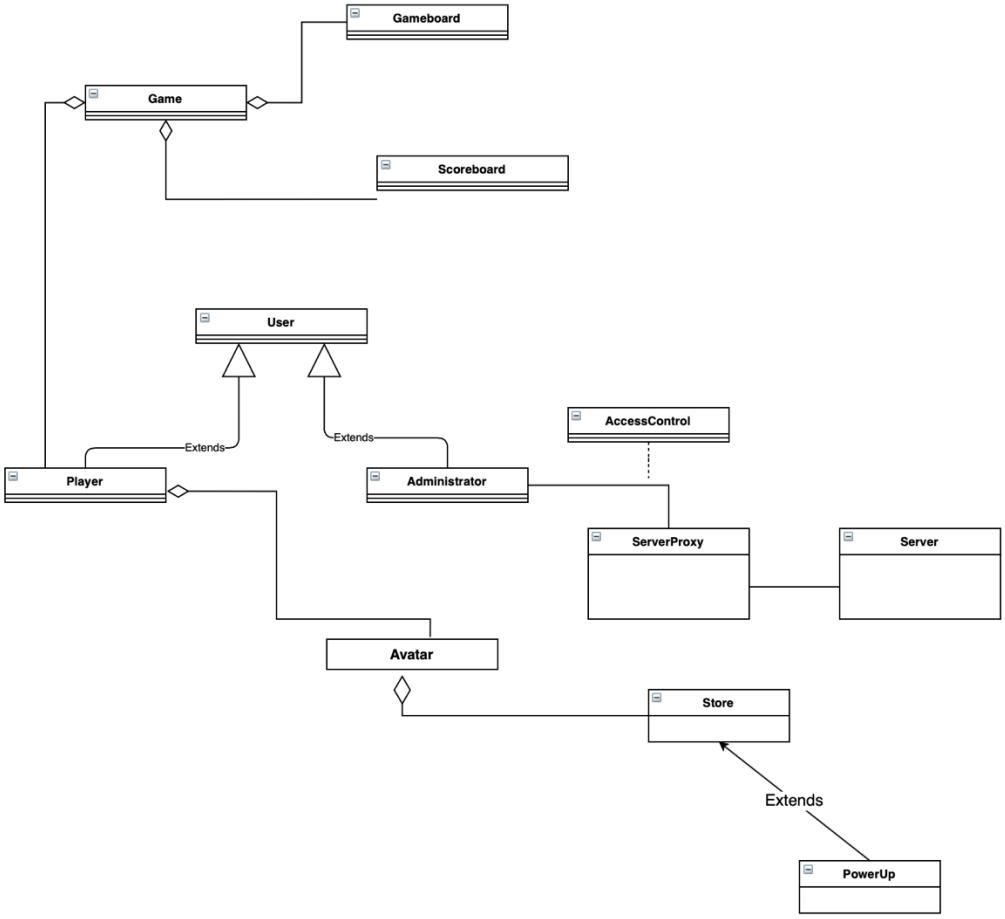


Figure 8 PRELIMINARY CLASS DIAGRAM

4. Dynamic Modelling of Use-Cases

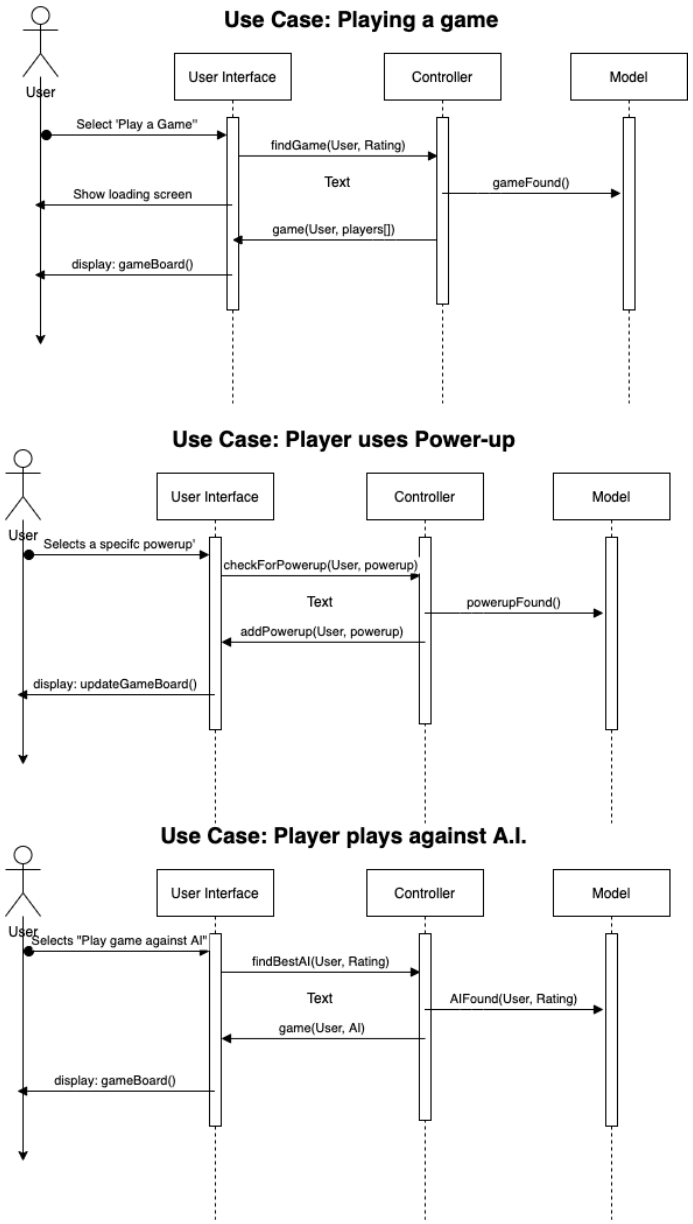


Figure 9 SEQUENCE DIAGRAMS FOR USE CASES

We note that there is a necessity for different classes that will handle important components such as the Artificial Intelligence bot, the option for using power-ups, and just playing a game with anybody around the globe in general.

4.1 Proposed System Architecture

For front end, we used the Model-View-Controller (MVC) model.

- Model: Back end admin, it will have all the computations.

- View: It will display to the players the data received from the controller.
- Controller: It will translate the output of Model and send to view.

For back end we used the Client-Server setup. Client-server setup was the best possible design because it allowed us to have multiple games with using less servers.

4.2 Initial Subsystem Decomposition

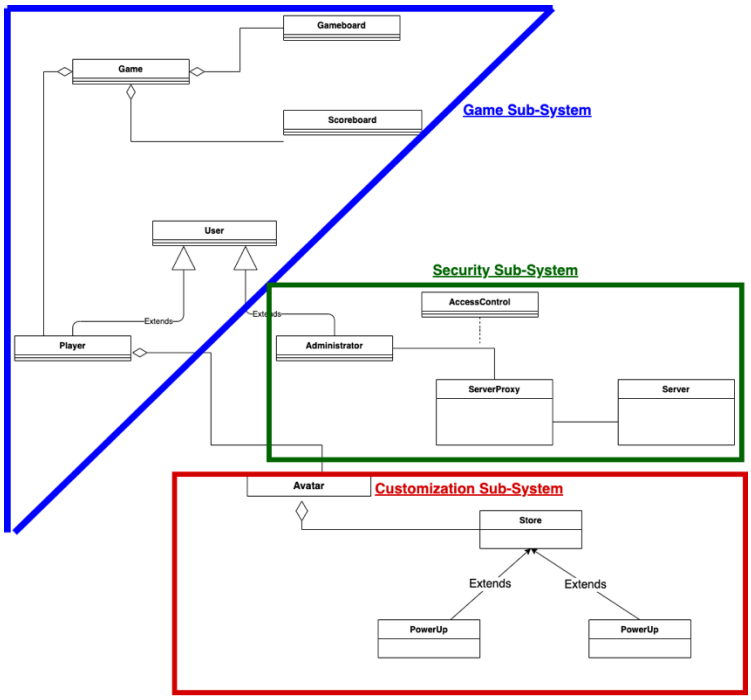


Figure 10 DETAILED CLASS DIAGRAM 1

As shown above, the complete class diagram is an extension of the preliminary class diagram shown in 3.1. Each class now shows the main parameters and methods that would be implemented. The 3 main subsystems were the following:

1. Customization

   Includes the Avatar, Store, CustomizeAvatar, and PowerUp classes. These classes allow the user to customize each aspect of their game ranging from changing their power ups and avatar look for their next game to buying new ones at the store.

2. Game

   Includes the Game, Gameboard, Scoreboard, User and Player classes. These classes set up the basic necessities of the game which include the player, the game itself, the gameboard, and the user/players in the game.

3. Security

Includes Administrator, Server, Proxy, Server, and Access Control classes. These all extend from the User parent class and revolve around the backbone structure of the game. The Administrator class creates a key which can send, make, and close requests. The Server/ServerProxy classes allow updates to be made to the game as well as updates to the database. Lastly, the AccessControl class controls authorization for the game.

5. Additional Design Considerations

5.1 Hardware / Software Mapping

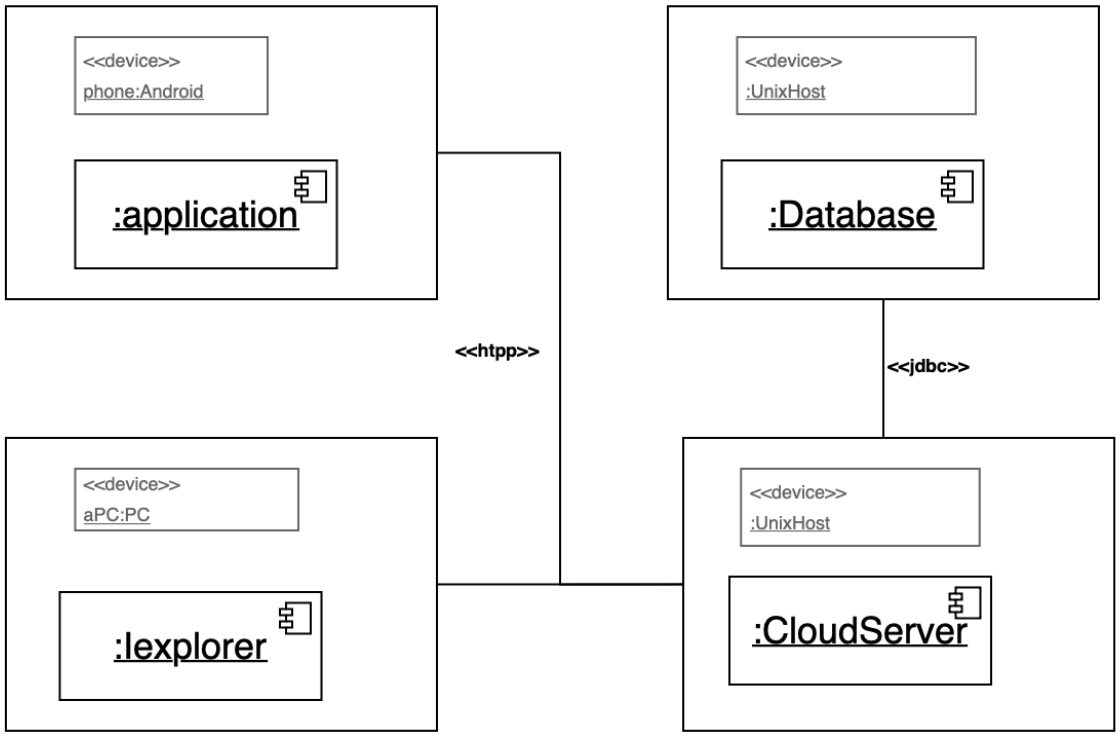


Figure 11 HARDWARE COMMUNICATION DEPLOYMENT DIAGRAM

5.2 Persistent Data Management

If for any reason the APIs or the servers associated with the app shut down, all user data and game data should be saved, which includes their account information, game, purchases, and pending friend requests via Cloud. The data stored on the provided Cloud server should be the same data last saved some time before the shut down occurred. Once an account has been created, it is created onto the cloud server as well and it is continuously updated for any possible loss of data. Once the servers are back up and the game is running again, the user should be able to log in and have all of their data recovered.

5.3 Access Control and Security

There needs to be a stringent setup for protection from outsider attacks and any possible mishaps from the inside. This will necessitate a whole set of new classes such as the **BackupPlan Class** which will hold all the necessary methods of pulling up the backup in case the current setup fails. Along with that, there will be a **OutsiderDetection** class which will immediately restrict access and let the System Administrator know in the case of an outside attack.

5.4 Global Software Control

Our system does not have a global software control component.

5.5 Boundary Conditions

The boundary conditions are held by the installation, launching, and closing of the app.

Installation: When installing the app from Apple app store, or Google play store, or Microsoft store. The user should be able to locate the app and install it on their preferred device.

Launching: Pressing 'open' from the mobile store, or pressing the app logo on their phone or computer to launch the app.

Closing: Going back to the home screen on the phone should pause the application but closing the app will stop the application. The game progress and the updates must be saved on both the device and the cloud data base, so we have an up to date back up available at all times.

5.6 User Interface

User interface design:

Figure 12 RELIMINARY USER INTERFACE DESIGN: GAME LOBBY ON IOS VERSION

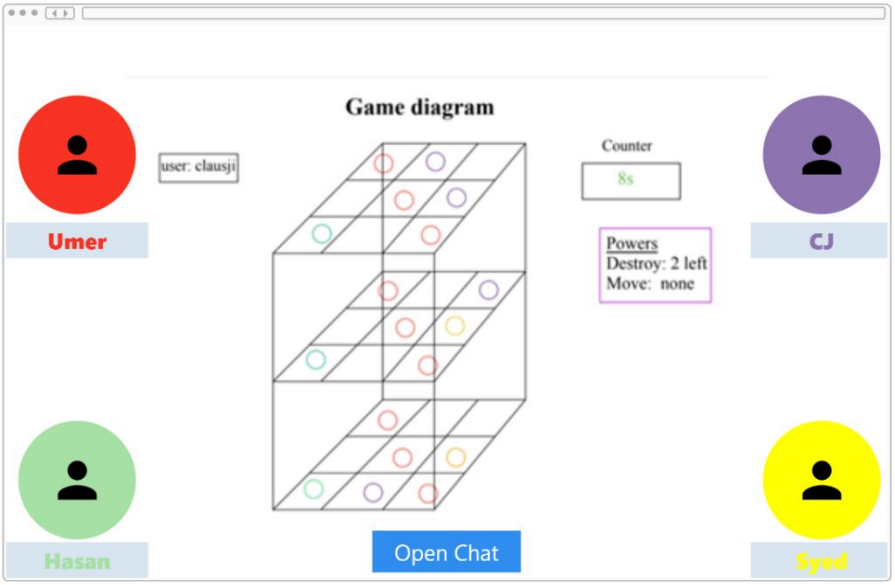


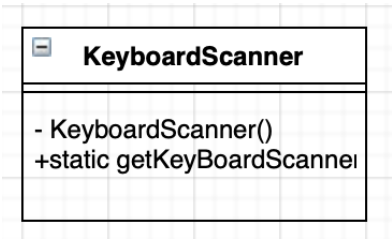
Figure 13 PRELIMINARY USER INTERFACE DESIGN: GAMEBOARD DESIGN ON WEB VERSION

5.7 Application of Design Patterns

We used the following design patters in our system design,

- Singleton pattern:

We used this design pattern because wanted to have only one instance of the Keyboard class. This keyboard instance will be used to coordinate keyboard actions across the system.



4.7.1 SINGLETON DESIGN PATTERN IN OUR SYSTEM

- Proxy pattern

We used this pattern to secure the access to the server by only authorize administrators. We added an access control class and a server proxy class. The access control class will check to see if the administrator has server access, if it does the access control class will return the server proxy to the administrator to make calls in the server. If the client does not have access no server proxy will be return.

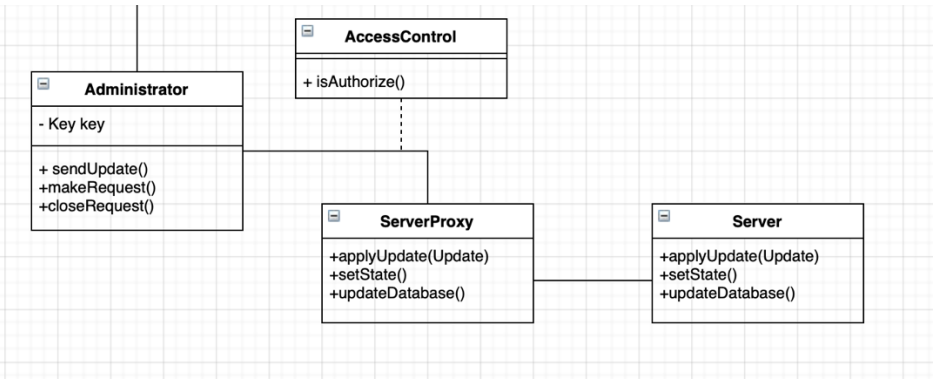


Figure 14 PROXY DESIGN PATTERN IN OUR SYSTEM

6. Final System Design

After all of the previous considerations, here is the final UML diagram with all of the respective subsystems separated. Note that after further consideration we switched around our subsystems and added a general one for normal back-end activities (System Administrator) and then further added another set of classes whose entire focus is protecting the program from outside attacks and what to do in the case of that scenario.

Final UML Diagram: (Below!)

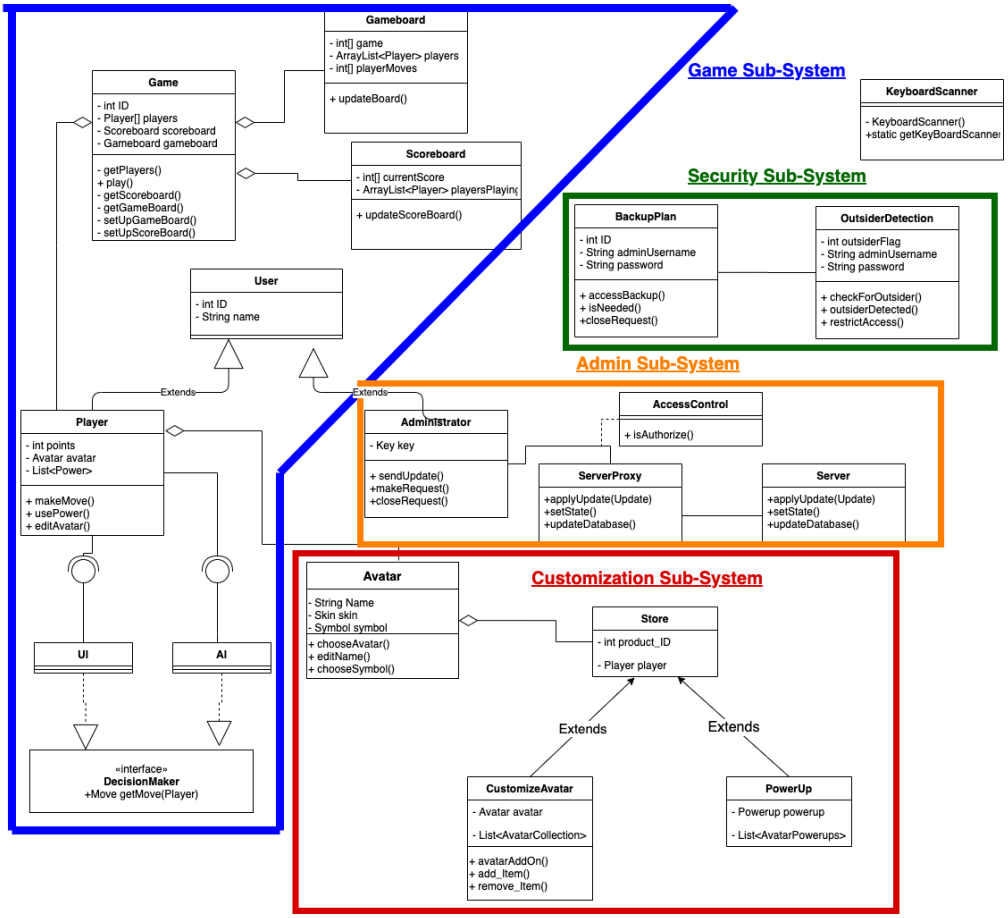


Figure 15 FINAL UML DIAGRAM

Important Classes to note:

Game Class: This is the main class that holds everything together and the most important class for the front-end.

OutsiderDetection Class: This class is dormant most of the time but used whenever there are possible attacks from the outside.

Store Class: It is important that the user have the ability to buy items from the store whenever they want

Administrator Class: Holds everything in the back-end together and with it being encapsulated as just another user, simplifies things.

7. Object Design

7.1 Packages

We will have one package for each of our subsystems,

1. Game package

2. Security package

3. Administrator package

4. Customization package

7.2 Game Subsystem

Our Game subsystem is composed of:

The Game class which is the parent class for the player, gameboard, and scoreboard, holds all the variables of those components.

The Gameboard class falls under the parent Game class which has the array list of players and player moves left, also keeps an updateBoard() function after each player takes their turn.

Scoreboard class also falls under the parent Game class and keeps a variable of the current score, also has a updateScoreBoard() function which keeps track of the score after each turn made by a player.

The player class falls under the parent Game and User classes, this keeps track of the points, avatar, and power ups available by each player.

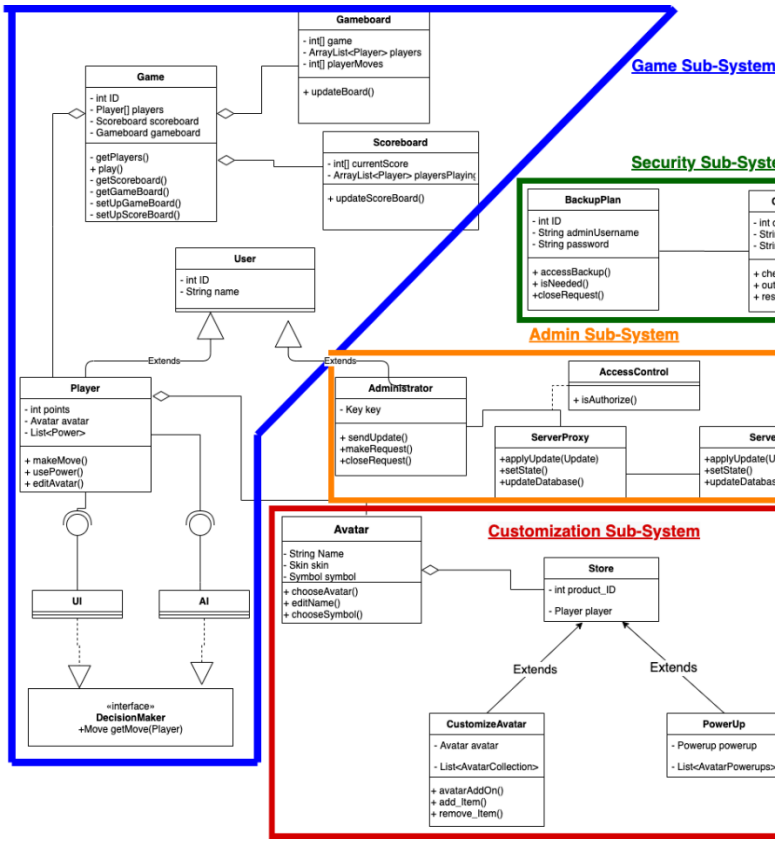


Figure 16 GAME SUBSYSTEM SHOWN IN THE BLUE BOX

7.3 Security Subsystem

Our Security subsystem is composed of 2 classes, the Backup Plan class and the OutsiderDetection class. The backupPlan class is used to save every progress from the user and save it into cloud incase the server goes down or the game crashes, all the progress will be saved. The OutsiderDetection class is used to check for security breaches and possible security attacks done towards the game.

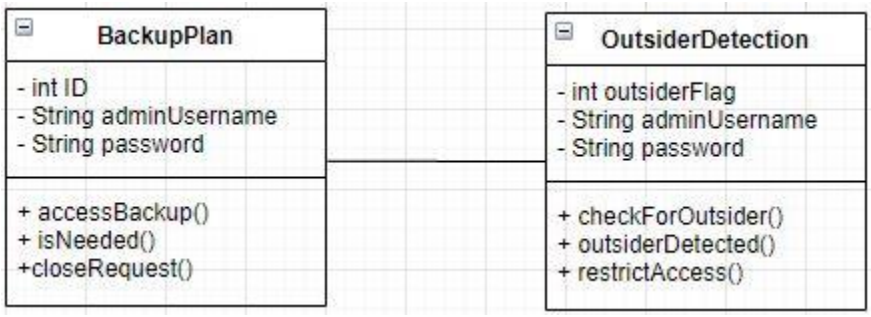


Figure 17 SECURITY SUBSYSTEM

7.4 Admin Subsystem

Our Admin subsystem is composed of 4 classes,

Administrator class extends the user class and is used to grant administrator status to an user.

The server class represents our system server that runs the games and stores the systems persistent data.

The other two classes (AccessControl and Server) are used to control access to the server. This subsystem implements the proxy design pattern.

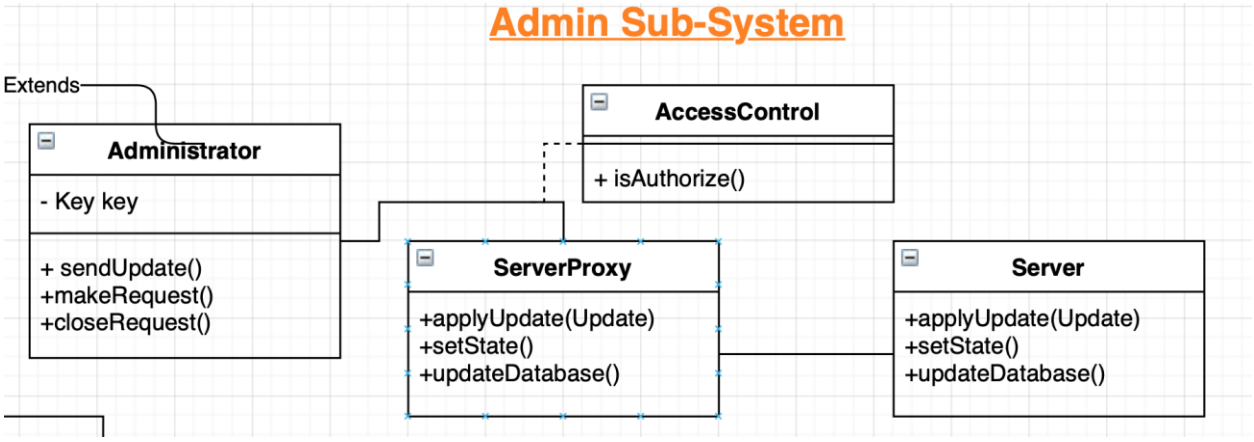


Figure 18 ADMIN SUB-SYSTEM DIAGRAM

7.5 Customization Subsystem

Avatar Class: This class is related to the Player class since every single player 'has' an avatar with a specific skin, symbol, etc.

Store Class: This class contains all of the products that a user may add to make their own avatar stronger through the use of powerups or to customize the avatar to their liking. It offers a lot of flexibility as well as making money for the company.

CustomizeAvatar Class: Since this is something that will be sold in the store, it is a child of the 'Store' class and it has all of the different ways a user can make their avatar more appealing. This setup is nice because if there is a need to add a new skin, it can be added here and then automatically reflected in the Store.

PowerUp Class: This class holds all of the potential power-ups that the user can apply to their avatar. Again, since it is something that will be sold in the store, it is a child of the 'Store' class. That way, there will be a seamless connection because whenever new powerups are added to the overall game, all users will be able to see them and add them instantly.

IV. Project Issues

1. Open Issues

- Due to the fluid regulatory environment related to privacy laws in Europe our application may not comply with all the new upcoming regulations. Our legal team will have to do a second review once the proposed regulation is passed by the EU legislative council.

- We are also in the process of updating our Gameserve software. The development team will communicate any updates that will impact our system and provide recommendation on how to implement them.

- The hardware system administrators are also changing the servers in our data warehouse. The new servers will have updated capacity and processing power. We will coordinate with them for system downtime and testing.

2. Off-the-Shelf Solutions

2.1 Ready-Made Products

Currently in the market there are no games with the same functionality and features as our current system design. However, there are a number of tic-tac-toe games that implement the traditional game model and could be uses as a guide for our development.

2.2 Reusable Components

Potential components:

1. Phaser
2. JavaScript
3. Angular
4. React components

2.3 Products That Can Be Copied

As mention before there are other companies that have developed tic-tac-toe games. These games, although they followed a similar model to ours are copyrighted and would not be used for our development.

3. New Problems

3.1 Effects on the Current Environment

Jobs that relate to creating a similar game may be impacted. However, this is a very minor problem as there is a good market for gaming competition in the gaming industry. This app directly competes with other games on ios, android, and windows platforms.

3.2 Effects on the Installed Systems

The application must be compatible with the latest release of iOS and Android. Also, the mobile device must have at least 3.1gb of storage to download. Web browsing/windows users will not have to worry about this if they are playing from the website.

3.3 Potential User Problems

The application will not be able to run if there is no internet connection available as this is a multiplayer based game. The application will not be able to perform to its full extent if the user plays as guest and does not create an account for avatar perks.

3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

If the user is using a different version than the most updated version on their mobile device, the application will not be able to run at its full potential and thus users won't be able to access the app till the latest version is installed to the latest version.

3.5 Follow-Up Problems

Users may not be able to find users on a public lobby if there are less players in their country or if there are not many users online in queue as well.

4. Migration to the New Product

4.1 Requirements for Migration to the New Product

Due to the fact that this is a new application, there will be no previous data needed to transfer into this release.

4.2 Data That Has to Be Modified or Translated for the New System

As mentioned earlier, no new previous data will be needed to modify or translate into this application release. The data that will be used on iOS will be shared to Android and Windows during the games release and as time goes by.

5. Risks

- The cost estimated was less than required.
- Covid-19 effected the production levels
- We had to cancel small updates to meet the release date
- Accurate information about hazardous use needs to be thoroughly vetted and pushed out immediately to prevent any medical issues from an overuse of the game.
- Proper harassment prevention document needs to be written, approved, and vetted before releasing it to the users and making sure everyone signs it and follows it.
- The game needs to run on the current and 3 older system updates for each respective software system.
- Back up servers needs to be read incase of a server shutdown or crash.

6. Costs

Monetary Costs:

- Salaries
    - Development team
    - Coding Team
    - Marketing Team

    - QA Team
    - Research Team
- Advertisements
- API Payments
- Github Pro

Time Costs:

- Development time
    - Research
    - Coding
    - Testing
    - Marketing
- How long does each batch lasts and needs updates.

Lost Opportunities:

- Lack of international scope.
- Focusing on English only.

7. Waiting Room

Some possible ideas would be to

- Include an option for video calling with friends
- Multiple game options
    - Speed 3D Tic-Tac-Toe and so on
- Play against bots that have avatars of popular people
- Collaborate with popular streamers to get more publicity
-

8. Ideas for Solutions

- Look into games that are popular these days and try to incorporate what made them successful.

    - Among us – Game for a group of friends, competitive, relatively quick rounds, simple graphics etc.
    - Fortnite – Battle Royale type of set up

- Have options for intense matchups between two people which viewers can appreciate
- Option for simpler graphics for somebody with a bad internet connection

9. Project Retrospective

Communication: we used Discord, and this was very helpful as everything was centralized in one place. For example, there were different channels for the coding and development project, there was a voice channel for meetings, and we could have shared any files, etc. with all of the team members. In that regard it was very productive.

On the other hand, it is to be expected that there will be communication struggles so what helped us was to have two set times for meeting and ensure that if somebody had to postpone, they try to do it at the very least two days before.

If the current online learning setup continues, then communication would easily be the biggest hurdle that a group has to overcome. It is important for everybody to be on the same page and make sure that everybody is pulling their weight and doing all of the expected work.

Organization: It is important to have a systematic manner of dividing up the work. There is a nice balance that should be sought for because some sections may be bigger than others. Another problem with dividing up the work is that everybody may not be on the same page. So it's important to all be on the same page so one team member is not doing something different from another team member whether that be regarding the classes that have to be written and their relationship or anything else.

V. Glossary

Users/players are use interchangeable in use cases and they refer to any person that plays the game.

System owners are persons that maintain or update the system from the back-end server.

All terms are used in this document:

Avatar: A player's customizable character

Rank: The position in hierarchy based on a player's wins

Points: Rewarded by completing certain milestones during games

Badges: Rewards granted after reaching a certain number of points

Power-charms: Special abilities a player can use during the game to get an advantage over other competitors

Google Accounts: Any account created through the Gmail service by the user which they have the option to connect to their User account on our platform.

Facebook Accounts: Any account created through the Facebook service by the user which they have the option to connect to their User account on our platform.

Google Play Store: The App store that the application will be published on for users that have Android Phones.

Apple App Store: The App store that the application will be published on for users that have iPhones.

AWS Servers: Servers to hold our data

Google Play Games: The social application provided by Google, giving users the ability to compare scores with their friends and get accomplishments.

Apple Game Center: The social application provided by Apple, giving users the ability to compare scores with their friends and get accomplishments.

Gameserve: The management and maintenance software use for the back end of 3D Tic-tac-toe.

VI. References / Bibliography

Bell, J. (2012). *Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports.* Chicago: Underwater Archaeological Society of Chicago.

Fowler, M. (2004). *UML Distilled, Third Edition.* Boston: Pearson Education.

Robertson, & Robertson. (n.d.). *Mastering the Requirements Process.*

Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). *Operating System Concepts* (Ninth ed.). Wiley.