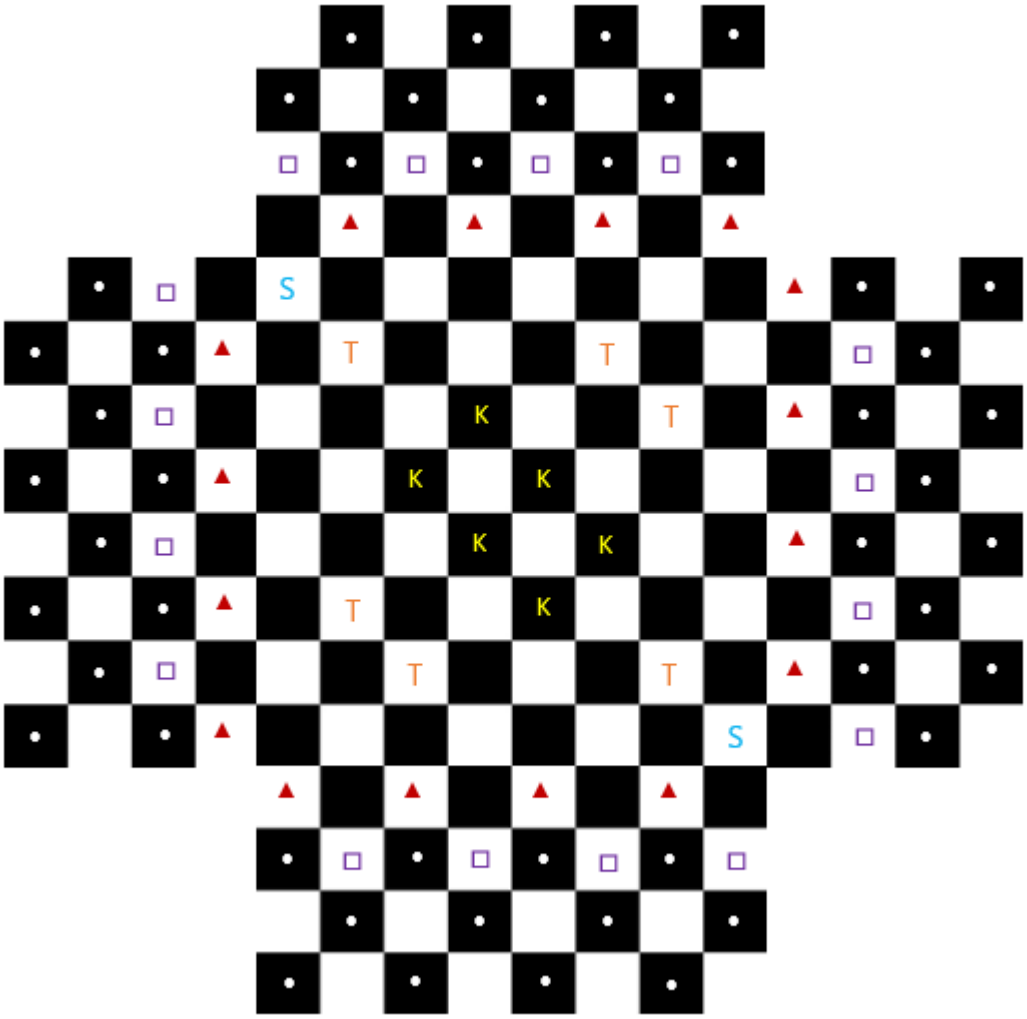
Checker Wars
Project Description Report

Group 16: Alex Choi, Andrew Macatangay, Sam Alammar, Luke Austin



CS 440
University of Illinois Chicago
October 2020

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

Checker Wars is a new, virtual take on the game of Checkers. It is being developed by Bohn Jell Entertainment to entertain game players and provide a new game for online competitive play. The game is a multidimensional version of Checkers with extra pieces and challenging elements that provide a more challenging experience with greater replayability.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

The video game industry is enormous. It has soared past the television and film business to become the biggest and fastest growing category of the entertainment business.

Checker Wars hopes to use this continual growth to establish a community of game players who play for fun and compete against each other in Checker Wars.

2b Goals of the Project

We wish to provide a new source of entertainment and competition to the growing field of online game playing and, specifically e-sports.

2c Measurement

We will measure the success of Checker Wars by the popularity of the game, both in number of downloads, and hopefully, streamers playing the game against each other via services such as twitch.tv.

3 The Scope of the Work

3a The Current Situation

Checker Wars is the first game developed by Bohn Jell Entertainment. The game will be the first product and therefore only source of revenue for the company.

3b The Context of the Work

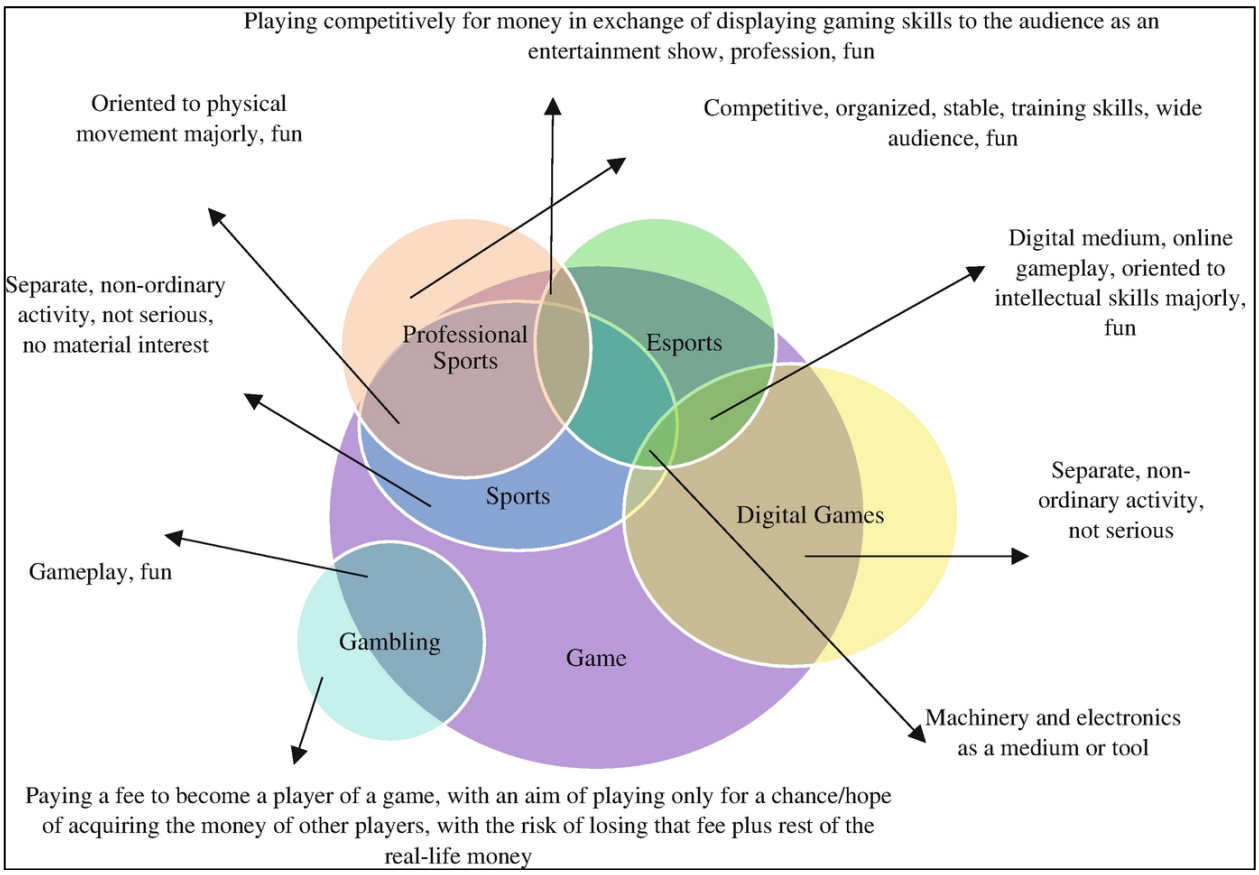


Figure 1: Game Environments Venn Diagram

The competitive gaming market is made up of many different disciplines. Originally physical, gameplay has moved into the virtual world, and e-sports are a large part of what encourages new players and keeps fueling the gaming industry. Without understanding where Checker Wars fits among these different sports and competitive markets, Checker Wars would be doomed to obscurity.

3c Work Partitioning

Table 1: Development Schedule

Event Name	Input and Output	Summary
1. Game start	User inputs credentials (in)	User logs in with username and password
2. Main Menu	Menu display (out)	Main menu is displayed where user can choose single-player or multiplayer gameplay
3. Single-player game mode	Gameplay board is displayed (out)	User plays against AI opponent to practice in-game skills
4. Multiplayer game mode	Gameplay board is displayed (out)	User connects to server and plays against human opponent
5. Game end	Score displayed (out)	User see's gameplay results and statistics. Score and rank increase or decrease depending on game result

3d Competing Products

Competitive single-player (with AI opponents) and multiplayer (both offline and online) games are all competitors of Checker Wars. Online Chess, Go, and similar strategy games are all competitors of Checker Wars. Some of these are standalone games, some of them are web apps, some are sold through services such as Steam or the Epic game store. These products' successes and failures must be studied and emulated to ensure Checker Wars is a success.

4 The Scope of the Product

4a Scenario Diagram(s)

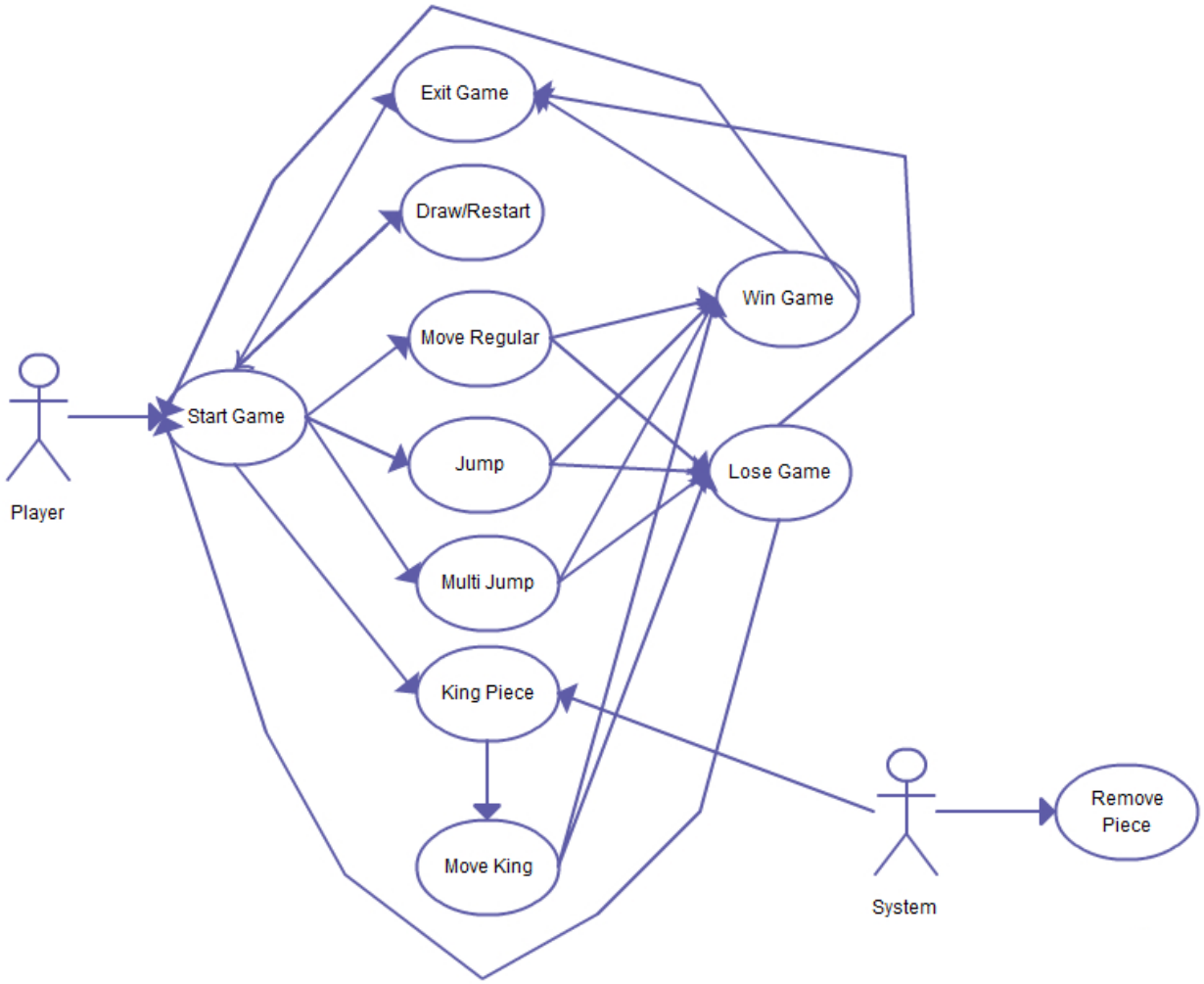


Figure 2: Checker Wars Scenario Diagram

4b Product Scenario List

Game Mode Select Scenario

Move Scenario

Move Pawn Scenario

Move Guardsman Scenario

Move Noble Scenario

Upgrade to King Scenario

Move King Scenario

Summon Space Scenario

Tower Space Scenario

Victory Scenario

4c Individual Product Scenarios

Game Mode Select Scenario: On the menu select screen, the user may select to play single player mode against an AI opponent or multiplayer mode against a human opponent via the internet.

Move Scenario: Each player's turn is determined by the roll of two die. The player with the highest roll goes first. This player gets to decide whether the order of turns goes clockwise or counterclockwise from them. Each player receives 5 turns per round. Each round ends when every player has exhausted their turns. Though, similar to checkers, each attack takes up one turn, and if an attack is chained, each chained attack will not take up another turn.

Move Pawn Scenario: The user opts to move a pawn piece on the game board. Pawns may only move to black spaces on the board.

Move Guardsman Scenario: The user opts to move a guardsman piece on the game board. Guardsman pieces may only move on white spaces on the board.

Move Noble Scenario: The user opts to move a noble piece on the game board. noble pieces may only move on white spaces on the board and cannot attack other pieces.

Move King Scenario: The user opts to move a King piece on the game board. The king pieces can move through either white or black spaces on the board. When a king transitions from one space color to another, that uses a turn and that king may not be able to move until the next round. When a king transitions from one color space, white or black, to another, it will do so on a space adjacent to it.

King Space Scenario: A pawn that lands on a King space is upgraded to a King piece for that player.

Summon Space Scenario: If a king and a noble lands on a summon space and is able to summon a new pawn piece on an adjacent black square. A summon space used by a noble or a king can only be used once per turn.

Tower Space Scenario: If a king or a noble lands on a tower space, they occupy that tower. If three towers are occupied by a single player, that player is victorious.

Attack Scenario: The user moves to attack an opponent's piece. Guardsman can attack all other pieces on white spaces. Nobles cannot attack other pieces. King's can attack any other piece. All pieces attack in a manner similar to the game of checkers, in that they "jump" over the piece to "take" them, and land on the other side of the opponents piece. A piece can only successfully attack another piece if its movement is unobstructed (can successfully land on the other side of the enemy piece.

Victory Scenario: A win is determined when either three towers are captured by one army or if the other players have no more pieces. When one player is completely defeated, they are out for the rest of the game and the game will continue with the other players. If for some reason, no moves exist for all players, the game ends in a draw.

5 Stakeholders

5a The Client

This product is being designed, produced, and commissioned by Bohn Jell Entertainment. Bohn Jell Entertainment has designed the specifications for the game but is outsourcing development and graphics.

5b The Customer

Checker Wars is aimed at teens and young adult players, however the game appeals to players of all ages due to its similarities with the game of checkers. The main focus will be on players who use streaming services such as twitch.tv.

5c Hands-On Users of the Product

User: Game Players

a. Role: both serious, competitive gamers and casual gamers will enjoy playing Checker Wars. Games usually take 5 to 10 minutes to start and finish, allowing students or workers who want a quick mental break from their study or work to fit in a quick game.

b. Subject Matter Experience: All players will begin as novice, and if they continue to play and learn the ins and outs and meta of the game, over time they will progress to journeyman and then master.

c. Technological Experience: Users technological experience is only required at a novice level. The game is very easy to install, and the gameplay is very intuitive and easy to pick up. The game is easy to start playing, and hard to master.

d. Other User Characteristics:

i. **Age**: All ages
ii. **Gender**: All Genders
iii. **Motivation**: Entertainment or competition.

5d Other Stakeholders

Because the game is focused around the competitive gaming and the e-sports scene, other potential stakeholders will be computer gear companies such as Razer or Logitech for sponsorship of the game or competitive players of the game.

5e User Participation

There will be an open beta that players can opt in to join (for free) in order to help the developers test the gameplay mechanics.

6 Mandated Constraints

6a Solution Constraints

Description: The game will be on a computing device such as a PC, Mac, or iPad.
Rationale: The game will need Internet connection to play against opponents via the internet.
Fit Criterion: The product will be connected to Internet using the user's connection to the Bohn Jell Entertainment's servers.

Description: The game will most likely be developed in an object-oriented language.
Rationale: The game will be played on PC, Mac, and Android operating systems, and will have objects that need to be manipulated.
Fit Criterion: All these platforms support development in object-oriented programming languages and coding style.

Description: The game will be played by players all around the world.
Rationale: The server of the game will be located in United States, but users from other countries will be able to use these game servers.
Fit Criterion: Any potential player with an internet connection will be able to download and play the game wherever they are located.

Description: The game will require the user to create an account.
Rationale: Users will need to create an account in order to track their stats and update their rank on both the local and global leaderboards.
Fit Criterion: The user will complete the registration form to sign up.

6b Implementation Environment of the Current System

The game is to be played on Windows, Mac, iOS, and Android devices. The easiest way to accommodate this is to have the app developed as a web application, wherein a user is required to log in through a website to play the game. This will be the first step in development, and local clients for each platform will be developed once the web application is completed.

6c Partner or Collaborative Applications

Checker Wars currently has no partner or collaborative applications.

6d Off-the-Shelf Software

Checker Wars is being programmed from the ground up, and so does not contain any off-the-shelf software.

6e Anticipated Workplace Environment

Both single player and multiplayer gameplay will require an internet connection initially while the web application is the only source of gameplay. Further down the line when the standalone applications for the game are developed for each system, users will be able to play single player games against an AI opponent offline.

6f Schedule Constraints

As previously discussed, the web application of the game will be developed first. Depending on reception and popularity of the game, development further into standalone Mac, Windows, iOS, and Android clients will continue, as will general development to the underlying codebase of the game. We aim to have the game developed and ready for launch by the Christmas holiday season.

6g Budget Constraints

Our first goal is to find an investor or sponsor to join us in funding the project for development. Sponsors can include computer and peripheral companies such as Razer, who can market their products beside the game, and when the game is complete, sponsorship of professional players of the game.

In-app purchases will take the form of cosmetic additions to the gameplay, as we want to create a fair competitive environment for all players of the game regardless of funds available to player. Thus, it is important that we find investors and sponsors who share this vision with us. The scope of the game is not massive, so we hope that we will not need copious funds to complete development of the game.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

All terms planned for the game are of general use and knowledge or are common to all other games played. These include, for example:

RANK: A player's rank among other players of the game, both locally and globally.

SCORE: A player's individual score at the end of every game

7b UML and Other Notation Used in This Document

This document generally follows the Version 3.0 OMG UML standard. Any exceptions are noted.

7c Data Dictionary for Any Included Models

Players may export and share their configuration (.conf) files for other players to load and experience what other players' personal gameplay UI looks like.

8 Relevant Facts and Assumptions

8a Facts

The game will take up a small amount of space when developed for individual platforms, but will initially all be hosted on Bohn Jell Entertainment's servers, and loaded (from Bohn Jell Entertainment's servers or local user cache) whenever a user logs on through the website. The game will be under 20MB in size.

8b Assumptions

It is assumed that the end user is using a sufficiently powerful computing device to play the game (2GB+ of RAM, modern ARM, Intel, or AMD processor).

II Requirements

9 Product Use Cases

This section describes in more specific detail the steps the system takes during its performance. These use cases more specifically define the system and user requirements and their boundaries, as well as functional requirements, and to organize work.

9a Use Case Diagrams

The use case below shows the interactions between players, the current match, and both the game and player servers. Data generated by the match and each player's moves are sent to the player server to record history of play, and the game server to update the board as a result of moves. The diagram may appear simple, but all of the mechanics of each move within the game as described in previous sections is handled within the game server and sent back to the current game state to update the board.

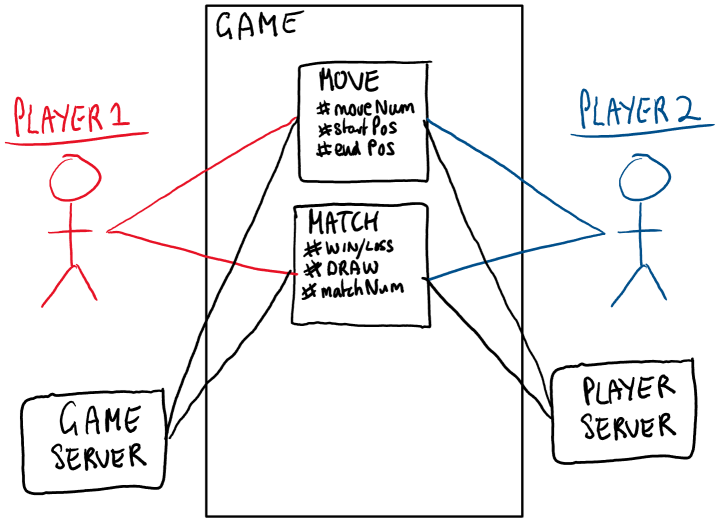


Figure 2: Match Use Case Diagram

10 Functional Requirements

ID# - FR1 – Save & Quit (Single-Player)

Description: The player must be able to save and quit at any point during the game when playing against an AI opponent.

Rationale: Life happens, and users sometimes need to pause what they're doing and come back later. This is a useful functionality for the game to have for players to enjoy playing the game.

Fit Criterion: Automatic test cases, as well as tester/user verified instances of gameplay being paused and quit, then resumed at a later time. Programmed unit tests with every release will test the reliability of the save and quit feature.

Acceptance Tests: FR1 – Save & Quit (Single-Player).

ID# - FR2 – Quit (Multiplayer)

Description: Players may disconnect during online gameplay or may leave the game for other reasons. The system needs to handle all of these instances gracefully. If a player disconnects, there needs to be a grace period that allows them to rejoin, but that grace period should not be too long, or other players will get frustrated.

Rationale: To facilitate a good gameplay experience for all users, when an opponent quits, the current player that is still connected needs to be able to wait for a period of time, then disconnect if they choose without penalty.

Fit Criterion: Player frustration surveys can be utilized to determine whether the balance between waiting for players to reconnect and moving on to a new game has been reached. Online Softbots will be used to test player disconnection results.

Acceptance Tests: FR2 – Quit (Multiplayer).

11 Data Requirements

Data requirements of Checker Wars are kept fairly small, but large enough for future product growth. Bohn Jell Entertainment needs to keep just enough data for the gameplay experience to be seamless, and for research purposes for either future games, or to improve Checker Wars AI opponents. All data will be kept on Bohn Jell Entertainment's off-site servers and will be backed up regularly and/or stored in RAID arrays to avoid any data loss.

ID# - DR1 – Player Data

Description: Each player's data, including number of games played, won, lost and drawn, ELO (rank/score), and time played.

Rationale: Players need to have ubiquitous access to their data so that no matter which computer or device they are using to play on their account, they can see their stats and record their gameplay history.

Fit Criterion: Automatic unit testing, as well as human testing will assure that data is retained and produced to the user in a timely manner, and if there are any issues presented, that a regular backup of all user data is available in case of data loss.

Acceptance Tests: FR1 – Save & Quit (Single-Player), FR2 – Quit (Multiplayer), DR1 – Player Data.

ID# - DR2 – Game Data

Description: All games played will be recorded for posterity on Checker Wars' servers. This includes the time, move of each player, and win/loss/quit.

Rationale: Keeping this data will help Bohn Jell Entertainment when developing better AI opponents for players. Often the creators of games themselves do not anticipate styles of play and the metagame that develops from their creation. There is no way we could program an AI opponent out of the gate that would beat humans forever, and so recording and analyzing human play will help with the development of more and more difficult AI opponents for players to play against.

Fit Criterion: Automatic unit testing, as well as human testing will assure that data is retained and produced to the user in a timely manner, and if there are any issues presented, that a regular backup of all user data is available in case of data loss.

Acceptance Tests: DR2 – Game Data.

12 Performance Requirements

12a Speed and Latency Requirements

For the single player experience, this should not be much of an issue, but for multiplayer, different players' internet speed and connections can have a big effect on the gameplay experience.

To start, we will have servers across America, Europe, and Asia to handle gameplay within these regions. Players will default to their specific region but may opt to play on other regions servers if they so choose.

To facilitate users ability for agency in the game, we will show each player's current ping updates every 2 seconds above their avatar in-game, so that if one player does lag or time-out, the other player will know that's the case, and will be able to wait OR decide to disconnect from the game to start a new one.

ID# - SLR1 - Latency

Description: Latency must always be minimized to the smallest extent possible in order to facilitate good user experience.

Rationale: All interactions with the environment need to have quick and obvious reactions from the system so that the user has timely feedback and knows the program is running correctly.

Fit Criterion: All responses handled by our servers will respond in a timely manner and be tested regularly for speed and accuracy. All responses given good internet connections should be in less than 2 seconds but have a 95% rate of response in less than 100ms.

Acceptance Tests: SLR1 – Latency.

12b Precision or Accuracy Requirements

In Checker Wars, precision and accuracy are paramount. The entire credibility of the game relies on users' movements being executed in exactly the manner they specify. Thus, there are no tolerances for failure of precision or accuracy.

Player score will be accurate to the integer level and not rounded.

Graphical precision and accuracy of board moves will display accurately within 50ms once the move has been registered with our servers.

12c Capacity Requirements

Because Checker Wars is a new game, we are not sure what the capacity requirements of our servers will need to be. Based on the success of other online multiplayer games,

we may need a large capacity, but Checker Wars is also not a very data intensive game, and so will not require the large bandwidth that other online (FPS, strategy) games require.

As a base, Checker Wars will need to be able to handle 10,000 consecutive games being played at launch. A team of data analysts will monitor base and peak loads and will update management as needed to determine when more resources need to be utilized.

ID# - CR1 – Server Capacity Requirement

Description: At launch Checker Wars needs enough capacity to handle possible influxes of new players.

Rationale: Imperative to a good player experience is a seamless gameplay experience. If players are unable to log on, join up, or if it even takes greater than a few seconds for them to join and play, they will opt to play something else instead.

Fit Criterion: Server availability needs to be kept at +20% of peak server capacity. Once a new peak is reached, new server capacity needs to be prepared.

Acceptance Tests: SLR1 - Latency, CR1 – Server Capacity Requirement.

13 Dependability Requirements

13a Reliability and Availability Requirements

Gameplay and server reliability are extremely important to gameplay and user experience. Though there always needs to be some tolerance of bugs, faults, and exploits, they need to be pre-empted and handled appropriately. Meaning quickly and gracefully, without impacting player experience for long or in a negative manner. If a player experiences a bug or crash, the data from that crash should be sent to Bohn Jell for investigation to fix the bug that caused it.

ID# - RR1 – Server Reliability and Availability

Description: Checker Wars' servers need to have a reliability that facilitates user expectations and provides a good experience.

Rationale: Imperative to a good player experience is a seamless gameplay experience. If players are unable to log on, join up, or if it even takes greater than a few seconds for them to join and play, they will opt to play something else instead.

Fit Criterion: 99.9% server uptime for player and game servers.

Acceptance Tests: SLR1 - Latency, CR1 – Server Capacity Requirement, RR1 – Server Reliability.

13b Robustness or Fault-Tolerance Requirements

ID# - FTR1 – Graceful Crashing

Description: Crashes are an unfortunate reality but can be dealt with. There is the possibility of logic, graphical, and connection issues while playing Checker Wars, and these instances need to be handled in a positive way for the user and for stopping these same crashes from happening again in the future.

Rationale: Without handling these errors and fixing the for later, players will lose interest in the game out of frustration.

Fit Criterion: Crashes must resolve within 2 seconds of happening and must also be reported to the game or player servers so that they can be analyzed and patched.

Acceptance Tests: SLR1 - Latency, CR1 – Server Capacity Requirement, RR1 – Server Reliability, FTR1 – Graceful Crashing.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

As with all products and services, maintenance is required and important to ensure performance and dependability. Monthly maintenance will be scheduled for the first Monday night of each month from 2:00AM to 6:00AM. This schedule will impact the least amount of players from launch, but data analysts may be able to provide useful insights as to another time that will affect less players once we have established a regular player base.

ID# - MR1 – Monthly Maintenance

Description: Server maintenance will ensure players run into less issues as they play Checker Wars.

Rationale: Maintenance is required and important to ensure performance, dependability, and trust in Checker Wars.

Fit Criterion: Monthly maintenance is completed, and logs are kept of what hardware/software was updated, and what needs to be updated at the next monthly maintenance period.

Acceptance Tests: MR1 – Monthly Maintenance.

14b Supportability Requirements

Checker Wars' support will be intuitive to users who play it. Tutorial levels will be provided for new users, and a video explaining gameplay will also be available. No explicit supportability requirements except for the above are listed, as the needs for supportability will change as the product evolves.

14c Adaptability Requirements

Refer to: 6b Implementation Environment of the Current System

14d Scalability or Extensibility Requirements

As listed in 12c Capacity requirements, Checker Wars will need to be able to handle 10,000 consecutive games being played at launch. Games can have a small player base and suddenly explode in popularity, needing extra server capacity in a very short amount of time.

There are services available today that allow the server capacity to expand as demand expands, but these are often expensive and leave our users data in the hands of other companies. Ideally Checker Wars will host their own servers and increase capacity as needed, but services that auto-expand with demand should be kept in mind as a possibility if needed.

ID# - SER1 – Scalability Requirement

Description: Game popularity can fluctuate greatly from day to day, week, to week, and sometimes even minute to minute. If a large influx of new players suddenly wants to join the Checker Wars world, we need to be able to handle them joining up all at once.

Rationale: Games can have a small player base and suddenly explode in popularity, needing extra server capacity in a very short amount of time.

Fit Criterion: As listed in CR1, server capacity needs to always be 20% above peak server loads. A contingency plan that includes the ability to quickly access and use new game and player servers within 12 hours needs to be in place. Even if these servers are temporary until more permanent servers can be installed, losing out on new players because of server capacity issues can be a death knell for games.

Acceptance Tests: SLR1 - Latency, CR1 – Server Capacity Requirement, SER1 – Scalability Requirement.

14e Longevity Requirements

No specific timeframe is suited beyond the initial launch +1 year. If the game hasn't garnered a regular user base by then, the game will need to be reassessed and altered, or the company needs to pivot to a new IP.

15 Security Requirements

Because Checker Wars does not log or record any sensitive data (we do not log IP addresses or any sensitive information other than that given to us by the users) our security requirements are minimal.

15a Access Requirements

Only those with direct needs for the data will have access to it. This includes programmers or researchers using data to improve products.

15b Integrity Requirements

Checker Wars shall prevent incorrect data from being introduced. For further information, refer to section 13 Dependability Requirements.

ID# - IR1 – Data Accuracy

Description: Data will be assessed as to whether it's correct before being introduced to the game or player servers within realistic constraints.

Rationale: Accurate data is imperative. If incorrect data is introduced, it can destroy the credibility of Checker Wars or even John Bell Entertainment as a company. So, our customers can trust us, we must ensure accurate, reliable recording and storage of data.

Fit Criterion: Data will be audited every 6 months by our data scientists to ensure we are not recording incorrect or erroneous data within a 99.99% tolerance range (0.01% of data can be corrupted in some way)

Acceptance Tests: IR1 – Data Accuracy.

15c Privacy Requirements

By design and to avoid any potential legal issues, Checker Wars will not record any personally identifiable information. The value of that kind of data is not worth the risk of holding on to it. Some information (account name, email address) that users give us will follow GDPR guidelines worldwide to protect user privacy.

If a customer opts to delete their account, it will be permanent. We will only keep data for active customers, and if a customer deletes their account and later wants to rejoin, they will have to create an entirely new account.

16 Usability and Humanity Requirements

16a Ease of Use Requirements

At first glance, the user must be able to utilize all controls with minimal difficulty. In this way, the user will solely concentrate on the content of the game rather than the frustration behind the controls or the lack of knowledge of the controls. As such, the controls must be intuitive and approachable. To achieve this, we will focus on several criteria, as defined by the client:

Efficiency of Use: The control scheme must be clearly defined in a way such that at first glance, the user will already be able to figure out which controls are being used.

Since the game may end up being multi-platform, the user must know whether the game will be utilizing touch controls, certain buttons on a keyboard, or a control pad. Upon playing a tutorial or reading an instruction manual, the user must be able to connect the context of what will occur in game, to the intended user input. The game must be able to react immediately to a user input and in a way such that the user predicts or immediately observes the in-game change.

Ease of Remembering: The game will be unique in its design and have limitless replayability. Although the game does contain some complex design, at its core-functionality, it is easy to replay, and the mechanics are easily remembered. Various options will be added to allow the player to customize the game in various ways and multiplayer and AI would be utilized to allow players to play with each other or by themselves. In this way, the player will be able to approach the game and have a unique experience every time.

Error Rates: In order to minimize the amount of errors that the user commits, the controls of the product will be reactive and yet, during certain movements in the game, the game will request for the user's confirmation of action. In this way, the user may be able to plan a move by inputting in action, and yet will not observe any consequence until the user confirms that action.

Overall Satisfaction: The user must not feel intimidated while playing the game, nor must they feel frustrations from the controls of the game. The mechanics are already unique to the game and so the feel of the game will follow suit. As such, the game has a wide demographic and already stands out from the market. The controls of the game are intuitive and certain design elements will be utilized in order to attract the attention of the user, such as color palette and art style.

Feedback: While playing the game, the user will be able to determine their action prior to the confirmation of that action. If a player wished to move a piece, the possible movements will be telegraphed to the player before the player confirms the action. This would also apply to attacking or utilization of a special space on the game board. In that way the player will be able to determine what they can or cannot do. Also, during certain sequences of the game, if the user hovers over an option with their controls, that option will be highlighted in order to allow the player to see the option that they will possibly select.

Upon the satisfaction of all these requirements, the product shall be easily utilized by a user with minimal technology experience at first glance, and it shall be easy for all, within the target demographic, to enjoy with minimal frustration.

ID# - EOU1 – Ease of Use Requirements

Description: The product shall be able to be used with only basic knowledge of the device that it resides on.

Rationale: To ensure that any user, regardless of technological experience, will be able to enjoy the product without any frustration pertaining to the mechanics of the product thereof.

Fit Criterion: Utilizing a test group of individuals with various technological experiences and of a multitude of backgrounds, at least 80 percent of the test group must be able to play the game successfully without any complaints or suggestions.

Acceptance Tests: EOU1 – Ease of Use Requirements

16b Personalization and Internationalization Requirements

Menu Options and Accessibility: On the menu screen, there is a button which will allow the user to change up various aspects of the program. The user can configure various graphics options or the resolution. The graphics options allow a user to play on computer systems of varying performance. The resolution of the window and the orientation of the window is also customizable so that the program can accompany monitors of varying sizes and will be allowed to be put into a windowed or full-screen mode. The sound settings are also able to be changed in this menu which will allow for various volume adjustments of music or sound effect volumes. There is a sound setting for the orientation of the user's output device. This setting appeals to whether the user is using headphones, 2.1 speaker setups, or a surround setup. Various theme changes are also available to the player in the options menu. The user can change the look of the various pieces or game board through the cosmetic purchases made, the default themes and color palettes, or through a user-generated theme.

Options for Users with Disabilities: The menu options screen feature additional options to include users that have disabilities. There is a colorblind mode which will change the color palette in a way such that colorblind players will have no issue distinguishing objects from each other on a screen. The product shall also feature adaptive gamepad support. This is support for non-conventional gamepads that are tailored towards users with certain motor-function disabilities. Another option that was implemented is text-to-speech and subtitles which will cater to users with visual and hearing impairment.

Game Configuration Options: In order to maximize the target demographic, the game will include many different customization options. For example, the AI difficulty shall be modular in that the player may be able to select a difficulty level that appeals to their playstyle or experience. Also, various aspects of each game are able to be customized. For instance, the level design may change from game to game, a timer can be implemented in order to speed up player choices, or the amount of actions may change depending on the user's choice to speed up gameplay. By doing this, not only will we create a game which can appeal to a wider audience of players, we will also be satisfying the conditions that are outlined in section 16a by allowing flexibility of gameplay and making the game endlessly playable. The game board layout and the number of players per game will also be customizable in that different hazards may be added to the game board and the board shape may change to accompany such players. Another implementation would involve team play in which players may choose or have

their teams randomly selected in a way such that players can play with each other in order to achieve victory. Another idea that will be implemented, called "Deus Ex Mode", will allow an AI to oversee the progress in a game and intervene with the current game in different ways by changing the game board layout, swapping the positions of units, removing certain units from the game, swapping player teammates, or helping out the player that is doing the worst. Tournament play will be allowed in that the players can play multiple rounds of games against different players based upon their success rate and a "fog of war" is a customization option which will block the players view of the entire board and only limit their view to their a specific range around their pieces.

Localization: The game is not localized to specifically United Stated users. The game has numerous customization options which will allow for different languages. If a player chooses a specific language, the entire UI will cater to that language user. Also any cosmetic purchases that are made in-game will be localized to the country of purchase meaning that if a person is in the U.S., the currency used will be dollars, whereas a person in South Korea would encounter a store in which the currency will be South Korean Won. This game should not be bound by any country's laws since this game will appeal to a general audience. As such, the game will not have any restrictions being released in any other country since this game does not involve any controversial topics, nor does it have any depictions of adult themes.

ID# - PIR1 – Personalization and Internationalization Requirements

Description: The product shall allow the user to configure their preferred settings and cater to users with a multitude of language backgrounds.

Rationale: Users located in different countries and with different device configurations will be able to access any settings to tailor their own user experience to their individual preferences.

Fit Criterion: The product must be able to work with all common languages with exceptional translation and through a panel of testers, must be customizable with all testers' devices.

Acceptance Tests: PIR1 – Personalization and Internationalization Requirements

16c Learning Requirements

This product will be able to be picked up by any person, regardless of the person's experience with the product or lack thereof and be instantly familiar in terms of navigation of the products various menus and controls. The mechanics of the game itself will be simple enough that a person who plays the tutorial will automatically recognize how to play the game and all the rules associated with it. The game shall provide instant feedback towards all the player's decisions which will help internalize the rules of the game.

Within a short amount of time, the player shall be familiar with the controls and mechanics of the game, but in terms of mastery of the game and game mechanics, it will involve the amount of games the player has played. A more proper estimate suggests that the game can be efficiently played by a player within 20 minutes or within the time frame in which the user completes the tutorial.

ID# - LKR1 – Learning and Knowledge Requirements

Description: The product shall be accessible to any person with basic knowledge of their device and be effectively played upon completion of the tutorial.

Rationale: A game with little learning curve will allow users that are intimidated by extensive game mechanics, to immediately enjoy the game and develop their own approach to their preferred playstyle.

Fit Criterion: A user with no prior knowledge of the game must be able to pick up the game, play through the tutorial, and be able to successfully play the game within a time frame of 20 minutes.

Acceptance Tests: LKR1 – Learning and Knowledge Requirements, EOU1 – Ease of Use Requirements.

16d Understandability and Politeness Requirements

A player with a minimal understanding of the language supported should have no trouble understanding any of the messages or their context thereof. All the messages within the product are made in a way such that they are clear and concise, and all messages are consistent in their fluidity and style and feature virtually no colloquialisms of the languages in which they reflect. All the context-based messages accurately and precisely interject in such a way that the user understands what is always going on. The product shall be understandable in such a way that a general user of a language may be able to approach the product with no ambiguity of understanding.

ID# - UPR1 – Understandability and Politeness Requirements

Description: The product shall include messages and responses which are concise and intuitive and minimize all ambiguity of meaning of any object used in-game.

Rationale: Any user with a basic understanding of their own language should be able to approach the game and immediately understand the context of the game and its underlining tool tips.

Fit Criterion: Users that fit into an age group of between 3 – 10 should be able to recite the meaning of all messages shown with 75 percent accuracy.

Acceptance Tests: UPR1 – Understandability and Politeness Requirements, EOU1 – Ease of Use Requirements, PIR1 – Personalization and Internationalization Requirements.

16e Accessibility Requirements

Approximately 20 percent of all males are red-green colorblind, and as such it would be a loss if no accessibility options were available for users with such disabilities. As seen in section 16b options for users with disabilities there will be various options in the settings menu found on the title screen to cater to these individuals.

There are options for users with common disabilities such as colorblindness, an option to accompany adaptive gamepads suited for users with motor function disabilities, subtitle options for users with hearing disabilities, and text-to-speech options for users with visual impairments.

There are no visuals that would require a epilepsy warning either and a health and safety warning intro screen is already implemented which will direct the user to either the instruction booklet or a website which underlines possible health risks associated with extended play of the game. As such, this product is compliant with the Americans with Disabilities Act and the International Game Developers Association (IGDA) Special Interest Group on Game Accessibility guidelines.

ID# - AR1 – Accessibility Requirements

Description: The product shall be approachable by any user with a common disability.

Rationale: Certain precautions and options must be taken into consideration in order to allow the product to be approachable by players with disabilities and in some countries, to be compliant with current disability laws.

Fit Criterion: 100 percent of users with common disabilities must be able to play the game with little to no difficulty.

Acceptance Tests: AR1 – Accessibility Requirements, UDR1 - User Documentation Requirements.

Upon purchase of the product, the user will be given various paperwork pertaining to the usage of said product. All paperwork would be in the language that is native to the country of purchase. This includes an instruction manual, which details the game's rules, mechanics, possible health warnings as previously mentioned in section 16e, and the various available options. This manual will be written in such a way that the user will be able to understand all of the mechanics of product and will be able to quickly find important information quickly in the event that they need to review material about the mechanics of the game. This will also include recommended and minimum specifications for a PC user even if this product will be able to be played on most mainstream computer configurations including Apple, Linux, or Windows based systems.

Another section of the instruction manual will include installation instructions even if the game will be primarily found on DRM (Digital Rights Management) systems such as google play store or steam. There will be virtual documentation which will need to

be reviewed and signed off by the user. These will involve terms and services for the online play of the game and for cosmetic purchases. These will outline the various rules and define possible instances that may result in a ban from online services or suspension of purchases. The instruction booklet will not be updated; however, the terms of service may need to be updated to accommodate future additions to the product.

ID# - UDR1 - User Documentation Requirements

Description: The product shall include terms-of-service and an instructional manual for the user.

Rationale: To legally protect the company and underline the consequences of a breach of contract to players that utilize the cosmetic store or play online. Also, to inform the users of the mechanics of the game and its installation and health and safety information that might be pertinent to certain players.

Fit Criterion: At any point of time, 100 percent of users must be able to reference the instruction manual and review the solutions that they might have regarding the mechanics of the product. Also, all users utilizing any of the online content must agree to the terms-of-service.

Acceptance Tests: UDR1 - User Documentation Requirements.

16f Training Requirements

The user will not need any formal training in order to play the game. A basic understanding of device mechanics will be needed. For example, a user playing with a touch-capacitive screen must be able to know how to operate a touch-capacitive screen and a user operating a desktop computer must be able to operate their IO devices such as keyboards, mice, or gamepads. Installation is covered in the instruction manual, but most of the installation should be straightforward and defined through the DRM system that the user is utilizing. Besides that, all the technical training of the game will be covered in the tutorial provided in-game. If the user has basic literacy, context of all buttons used in the game are clear and concise. The product shall provide instruction to user's in-game if the user has a basic understanding of the device that the game is installed on.

ID# - TR1 -Training Requirements

Description: The product shall not require any formal training from the user apart from the included tutorial from first-time users.

Rationale: The installation and operation of the product will be simple enough such that the user will feel no frustration in proceeding to play the game past the tutorial.

Fit Criterion: The user will be able to play the game within 20 minutes or less upon the completion of the tutorial and installation of the game.

Acceptance Tests: TR1 -Training Requirements.

17 Look and Feel Requirements

17a Appearance Requirements

The appearance of the product shall not incorporate any controversial images or involve and seizure-inducing images or animations. The images and colors shown must not affect gameplay in any sort of capacity. The design and color palette must be acceptable and attractive to users. Upon starting of the game, Bohn Jell Entertainment's logo is displayed using animations, as well as, other companies that were involved in the development and the planning of the game. The client's corporate branding is also included in the intro screen. The colors and animations used to display each corporation are consistent with each company's signature colors and do not, in any way, deface any of the corporate brand's image.

Different design elements and the colors come together to form the main theme of all the game screens which must not hinder gameplay. Animations between menu screens and on the game board are made to be fluid so that they will not hinder user gameplay. A prototype of all the colors, images, themes, and animations shall be compliant with the client's requirements as well as any of the user purchasable cosmetic items.

ID# - APR1 - Appearance Requirements

Description: All design elements of the product shall not hinder the gameplay of the user and shall appeal to corporate branding standards.

Rationale: In order to maximize user amusement, the user must be able to focus on the gameplay in and of itself and the corporate branding of the client and associated development companies must satisfy their needs and not tarnish their brand as to ensure future client relations.

Fit Criterion: All associated companies must approve the image in which their brand is portrayed, and a test panel of users must accept the design of the product with a percentage of no less than 80 percent.

Acceptance Tests: APR1 - Appearance Requirements.

17b Style Requirements

The style of the UI (User Interface) must appeal to the user and be acceptable for users of all ages. The style of the product and its packaging will reflect the theme and feel of the game, in that it must be playful and friendly, and yet exude the feeling of an incoming conflict. It must be approachable and yet reflect the ominous and dramatic feeling of an upcoming battle. With that being said, the style of the product must also tread carefully in that it should not be too overbearing as to dishearten a potential user from playing the game.

The packaging shall be professional and yet sprightly in such a way that the user will feel enticed to read it. The packaging for the physical copy of the game will reflect the overall feel and mechanics of the game and will be conservative in its sizing in order to adhere to other country's specification requirements. As for users looking to download the game from a DRM system, the store pages will show images and videos as well as a summary which reflects the product's gameplay. As a result, the product's style must be made in such a way that at least 60 percent of potential users who see the packaging or the store page will be enticed to learn more about the product just at a first glance.

ID# - STR1 - Style Requirements

Description: The style of the product shall not affect the user's experience and shall not overbear the user in any way.

Rationale: To ensure that the user will return to the product and feel a sense of immersion which will enhance the gameplay of the player.

Fit Criterion: 90 percent of all surveyed users must feel satisfaction and respond with relevance in terms of the style of the product.

Acceptance Tests: STR1 - Style Requirements, APR1 - Appearance Requirements

18 Operational and Environmental Requirements

18a Expected Physical Environment

The product will exist on whatever platform the user downloaded or installed the game onto. This may be any mobile device, desktop, or laptop computer systems. As such, this game will be able to be played while on the move or while the user is stationary in an indoor environment. Based upon this, the product will be utilized in any environment which is satisfactory for the device that is being used. This product shall be used in any sort of environment if it is within an acceptable boundary for the device. This may involve a lowlight situation or a completely controlled environment such as inside a household.

ID# - EPE1 - Expected Physical Environment

Description: The product shall be completely usable in any environment in which the device it resides, will properly function.

Rationale: The software should be able to work in any environment so long that the device that it is being played on, works without issue. That way the user can play the product wherever they feel like within the device's environmental constraints.

Fit Criterion: So long as the device functions, the game must also be able to be played on 100 percent of the devices.

Acceptance Tests: EPE1 - Expected Physical Environment

18b Requirements for Interfacing with Adjacent Systems

This product shall work on any legacy operating systems as well as on the most modern operating systems. This product will be compatible with older versions of windows from Windows XP to Windows 10 and will be available for all versions of modern-day versions of macOS and Linux. The product will also be compatible with all versions of android and IOS.

As a result of these compatibilities, the application being developed will run smoothly on any of these platforms and will be able to run with any overlays that are available from other programs. The operating system that will house the product will not only provide storage but will also allow the product to communicate with other user's that have the product and with the online store that allows for cosmetic purchases. The data sent between users will solely be from online multiplayer play and will contain no data such that the communication will compromise the user's personal information. Also, the communication with the store will only include data that is pertinent to the main user such as credit card information and prior and available purchases for the current user.

The multiplayer information will be updated during every round and the store information will be updated upon opening of the store or purchase of an item. Multiplayer and store information will be sent over a wireless internet connection and will be unavailable to users that play offline. Upon purchase of a cosmetic item on the online store, this data will be saved on the user's computer so that the user will be able to access their cosmetics during offline play.

ID# - RIAS1 - Requirements for Interfacing with Adjacent Systems

Description: The product shall be compatible with all modern operating systems, current, relevant mobile devices, between users on different devices. and with any DRM software that it is being distributed from.

Rationale: This will allow more users to be able to enjoy the product from a multitude of devices and to be able to take advantage of the product's features with little frustration.

Fit Criterion: 100 percent of all users, regardless of device, must be able to play the product with each other with no difficulty.

Acceptance Tests: RIAS1 - Requirements for Interfacing with Adjacent Systems.

18c Productization Requirements

The product will be distributed in both a physical and virtual copy. The physical copy will be distributed using a CD, catering to users that do not have access to a reliable internet or for users who prefer to collect physical CDs. The physical CDs will only be

supported by users that are using a standard computer platform. The downloadable version will be available to users of mobile devices and for computer users.

The installation process for the physical copy will be easy to understand as the user will only need to run the CD. From there, the InstallShield Wizard® installation utility will be activated and will guide the user through the installation process. While this is occurring, no additional input is needed for the user during the rest of the installation. For the downloadable version, the user will need to have an appropriate DRM software. This software allows for the management of digital licenses and will help prevent unauthorized redistribution of the product. Examples of supported platforms for DRM services are Steam on Windows, Linux, and MacOS computers, Google Play store on Android-based devices, and the Apple Appstore for Apple mobile devices. There are other examples which will be utilized such as Epic Game Store, GOG, and Uplay.

These services have their own installation processes which are relatively pain-free. The services will automatically allow for distribution of the software and save a library of games associated with the user. At any moment of time, the user can select which game they want to install where a download and installation process will automatically start. At the completion of installation, the user will be able to instantly access the game with no issue. The DRM services also allow for game developers to host their game stores which will attach to the in-game purchases made. These services will automatically keep track of the user's purchases and will automatically update and install the user's newly purchased cosmetic items. The product shall be accessible via a physical or a downloadable content and will feature redistribution protection through DRM systems. The product installation shall be simple and involve minimal input from an untrained user. As such, the product must be able to be installed within 5 minutes of the user purchasing the product.

ID# - PRODR1 - Productization Requirements

Description: The product shall require minimal user input to install onto the device of their choice and be distributable through physical and downloadable copies.

Rationale: By achieving this, the user will not be concerned with the installation at all and will instead focus on the content of the product.

Fit Criterion: The product must be able to be installed within 5 minutes of the user purchasing the product.

Acceptance Tests: PRODR1 - Productization Requirements.

18d Release Requirements

The product shall be consistently updated anytime a bug fix is required. Also, the product shall be updated with new content in a period of 5 years or earlier depending on whether the client feels that the product has successfully served its purpose. During this period, any new findings on bugs or stability issues will be addressed in a timely manner. This will be done through a ticketing system which will keep track of any

complaints or issues from users where a dedicated team of developers will suggest actions to be included in the next update. This may include, but not be limited to, any issues with compatibility of preexisting systems, any new possible issues with compatibility with a new operating system update, any compatibility with aforementioned gamepads, errors in code that may have been overlooked, and possible issues with performance due to a lack of optimization of the product with the platform it is running on. In terms of new content, new content will be added yearly or biyearly, in the form of new cosmetic shop items or new game additions. This will be primarily determined by the client. Any and all updates to the game shall not interfere with previous released version and will be thoroughly tested in a manner such that each issue that is being addressed will be completely rectified without affecting the game in other aspects.

ID# - RELR1 - Release Requirements

Description: The product shall be supported by the development team for a time period designated by the client, in which issues pertaining to the programming of the product will be addressed and additional content will be added.

Rationale: With regular updates, the game will not get stale to the user in that any issue that is brought up will be promptly fixed and will not affect the user's experience in any way. All additional content will ensure that the player's support for the game will be longer-lived.

Fit Criterion: Within a period of 5 years or otherwise outlined by the client, the product needs to be updated yearly or more occasionally if needed.

Acceptance Tests: RELR1 - Release Requirements.

19 Cultural and Political Requirements

19a Cultural Requirements

Regardless of gender, race, culture, religion, or sexual orientation, this product shall not offend anyone. Any newly added content will also take this requirement into effect. This product will keep all users in mind and is developed in such a way that any historical or modern association will be purely coincidental. This product is not associated with any controversial themes or groups and therefore will appeal to any audience of different countries.

Depending on which country the game is purchased, the game may consider events that involve popular, but not controversial, holidays that the country may observe. Any cosmetic purchases will also take this into heavy consideration and will be carefully developed in such a way that they will in no way be considered offensive. Any user-created content shared online will be reviewed and if for any reason a user-created theme is considered offensive, strict and rapid action will be given which will notify the user of why their content may be offensive and will be terminated.

ID# - CULTR1 - Cultural Requirements

Description: The product shall not offend any user regardless of the user's gender, race, culture, religion, or sexual orientation.

Rationale: By achieving this, the product will not be subject to negative attention and will appeal to a larger target demographic.

Fit Criterion: The game will be accepted by 100 percent of users regardless of the user's gender, race, culture, religion, or sexual orientation.

Acceptance Tests: CULTR1 - Cultural Requirements.

19b Political Requirements

This product will not include any material borrowed from any other media properties unless authorized and intentions are clearly put on paper. As such, unless the other party has consented, this product will not infringe on any other company's intellectual properties. The source code of the project will only be accessible to the development team and certain other individuals within the company of development. During development, all software developments pertaining to the product will solely belong to the company of development, Bohn Jell Entertainment.

The product must appeal to all regulations enacted by the countries of localizations and must not break any laws that may reference software and intellectual property laws. The game shall be reviewed and hopefully accepted by that country's software rating board, such as ESRB (Entertainment Software Rating Board), before release of the product in that country. If any of the employees associated with the development of the project may be involved in controversial events, that employee will be subject to possible consequence which may involve termination of such employee. The company will swiftly react to any negative publicity with appropriate and professional responses.

ID# - POLR1 - Political Requirements

Description: The product shall satisfy the ethical code of conduct, company policy on employee conduct, and a country's copyright and labor laws.

Rationale: This will avoid negative press about the game and will avoid legal issues that may arise with the development of the product.

Fit Criterion: The product must be compliant with 100 percent of company guidelines on conduct.

Acceptance Tests: POLR1 - Political Requirements, CULTR1 - Cultural Requirements.

20 Legal Requirements

20a Compliance Requirements

None of the images, animations, mechanics, or general content shall involve any contentious figures. All software relating to this product will comply with any of the laws of countries of sale and shall be reviewed by any pertinent media rating boards. This product will be required to comply with the Data Protection Act and all associated content of the product will be explicitly reviewed in order to not infringe on the intellectual properties of any other companies. Any of the content created for this game will be unique for this game.

All labor laws will be complied with. Employees working on the development of this project are subject to certain requirements that are outlined by country's labor laws, such as the Fair Labor Standards Act. Any paperwork will be reviewed by an appropriate lawyer and any legal issues will be dealt with quickly. All work done with additional content and user-generated content will be subject to heavy scrutiny to minimize any potential legal issues. Copyright laws for every country of sale will be reviewed and taken into consideration.

ID# - COMPR1 - Compliance Requirements

Description: The product shall be compliant with the Data Protection Act, copyright laws, and the Fair Labor Standards Act, and shall be devoid of any controversial elements.

Rationale: To avoid lawsuits and controversy.

Fit Criterion: A legal team must 100 percent approve of the development and distribution of said product.

Acceptance Tests: COMPR1 - Compliance Requirements.

20b Standards Requirements

This product shall be required to be approved by certain parties including the client and the project manager. Upon approval, the product shall be tested thoroughly and adhere to the company's standard of quality. The product must be in working condition within a certain timeframe and must be devoid of any apparent issues in programming. Before distribution, the product must be ESRB compliant and must be of enough working order. The product must also be compliant with our own ethical code of conduct

ID# - STANDR1 - Standards Requirements

Description: The product shall adhere to the company's standards of quality.

Rationale: This will ensure that Bohn Jell Entertainment will be seen in a positive light to existing and potential clients.

Fit Criterion: The product must comply with the company's 10 standards of quality.

Acceptance Tests: STANDR1 - Standards Requirements.

21 Requirements Acceptance Tests

Refer to the Fit Criterion of each ID# for the required acceptance tests.

21a Requirements – Test Correspondence Summary

	Requirements																											
Test	FR1	FR2	DR1	DR2	SLR1	CR1	RR1	FTR1	MR1	SER1	IR1	EOU1	PIR1	LKR1	UPR1	AR1	UDR1	TR1	APR1	STR1	EPE1	RIAS1	PRODR1	RELR1	CULTR1	POLR1	COMPR1	STANDR1
FR1	X																											
FR2		X																										
DR1	X	X	X																									
DR2				X																								
SLR1					X																							
CR1					X	X																						
RR1					X	X	X																					
FTR1					X	X	X	X																				
MR1									X																			
SER1					X	X				X																		
IR1											X																	
EOU1												X																
PIR1													X															
LKR1													X	X														
UPR1												X	X		X													
AR1															X	X												
UDR1																	X											
TR1																		X										
APR1																			X									
STR1																			X	X								
EPE1																					X							
RIAS1																						X						
PRODR1																							X					
RELR1																								X				
CULTR1																									X			
POLR1																									X	X		
COMPR1																											X	
STANDR1																												X

Table 2: Requirements - Acceptance Tests Correspondence

21b Acceptance Test Descriptions

Refer to the fit criterion of the required ID# above for the description of the acceptance tests.

III Design

22 Design Goals

Though most computer games place a priority on speed over accuracy, accuracy is actually the most important part of the equation for an online strategy game like Checker Wars. Our design goals include both, but the goal emphasizes accuracy of recording and graphical representation.

23 Current System Design

There is currently no existing system, as this is a novel online game.

24 Proposed System Design

This section will make heavy use of UML, class diagrams, and also sequence diagrams where needed.

24a Initial System Analysis and Class Identification

What will be needed to create the game are a CheckerWarsGame class to hold the Gameboard (which holds the players and towers for the game). This will extend from the tile class, which extends from the Tiles class(es). The Scenario class will contain the different available moves (Attack, Move, Victory), which is connected to the Player/Enemy class. This class is connected to the Piece class, which extends from the different Pieces (Pawn, Guardsman, Noble, and King). These connections are detailed in brief illustration below, which is further refined in section 27.

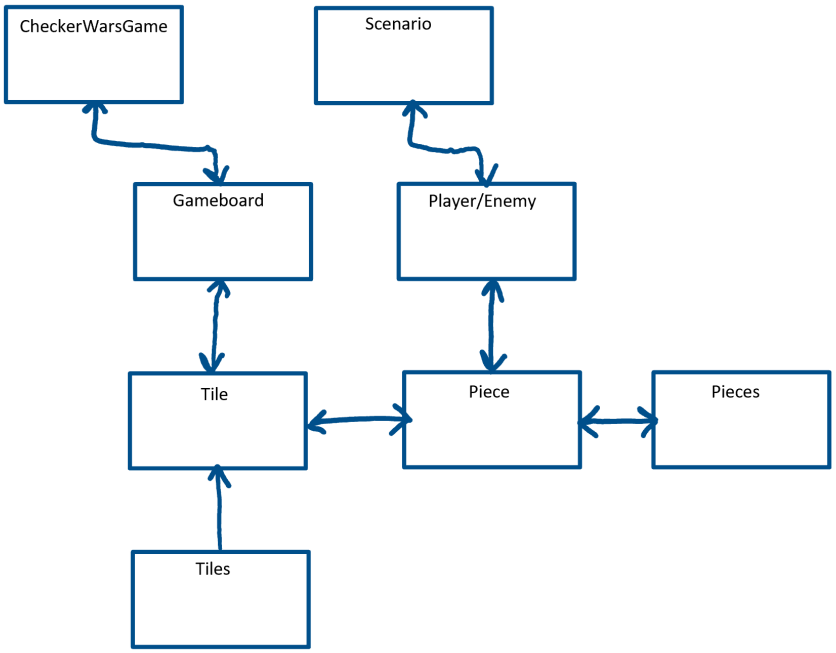


Figure 3: Initial Class Diagram

24b Dynamic Modelling of Use-Cases

The two main moves that will be happening within the game are the move scenario, and the victory scenario. It is important that the game server checks for the victory scenario after every turn is played, so that players will immediately know who has won, and also so in the future players will be able to receive updates of the potential moves that would cause a win or a loss (for training mode).

The move Scenario and the Victory Scenario are laid out as sequence diagrams below. The details are left up to the developers on how to implement the checking for each case. For the move scenario there are many different possible moves, so a lot will have to be done to check this condition.

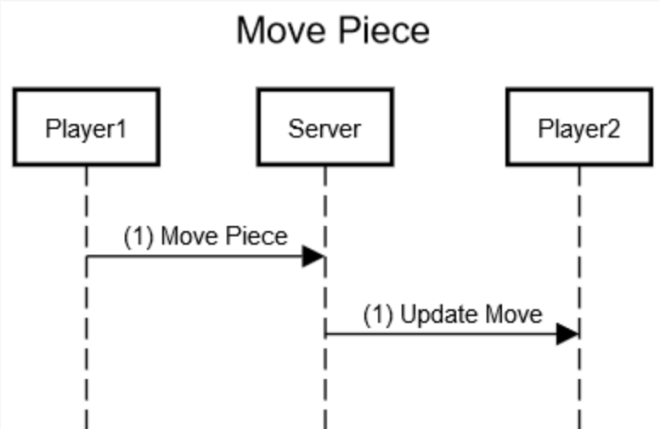


Figure 4: Move Piece Sequence Diagram

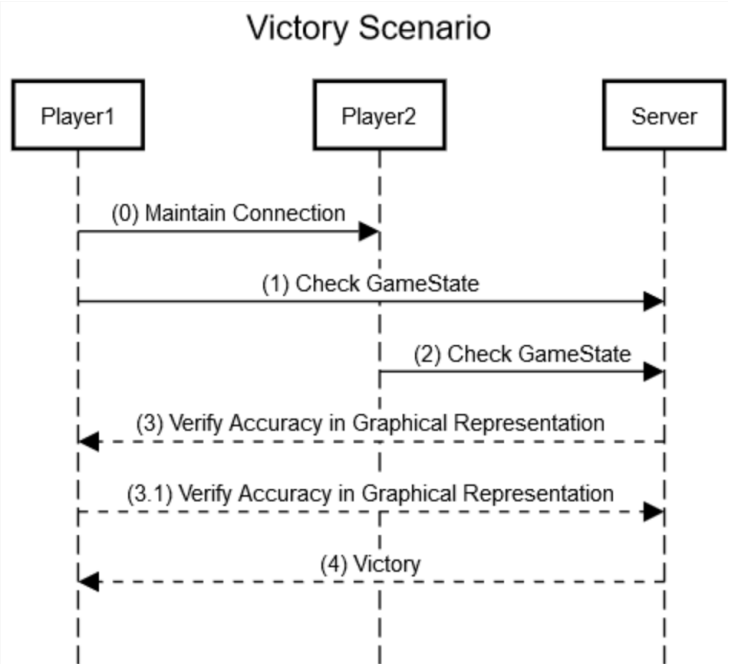


Figure 5: Victory Scenario Sequence Diagram

24c Proposed System Architecture

We will use the client-server model for Checker Wars. In this model, the client connects to the web server (through CheckerWars.com or their respective game client), which controls the GUI layer, the Application Layer (which holds the logical representation of the game), and the Data Layer which gets data from the database and from web services.

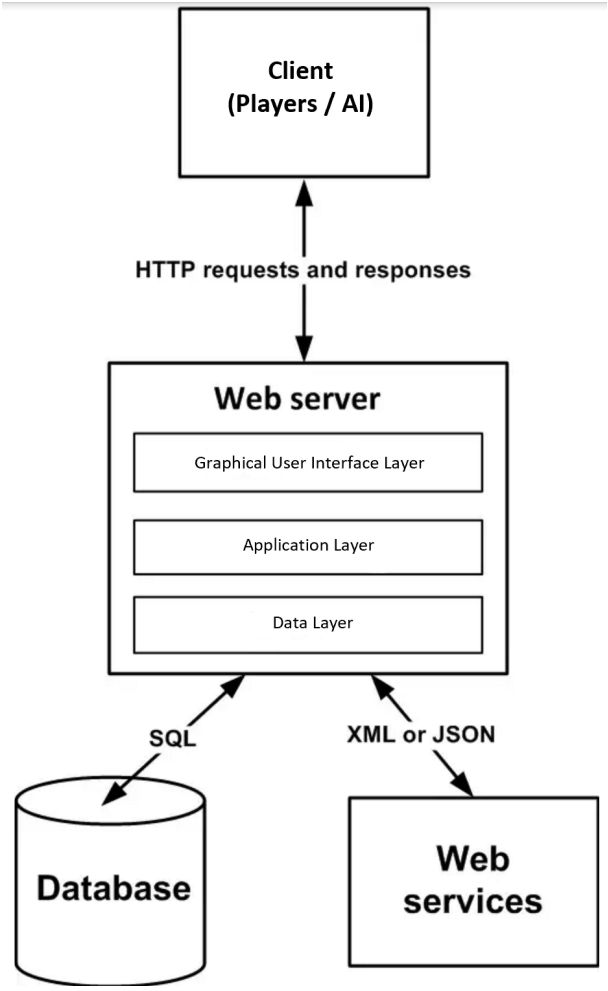


Figure 6: Client-Server Model for CheckerWars.com

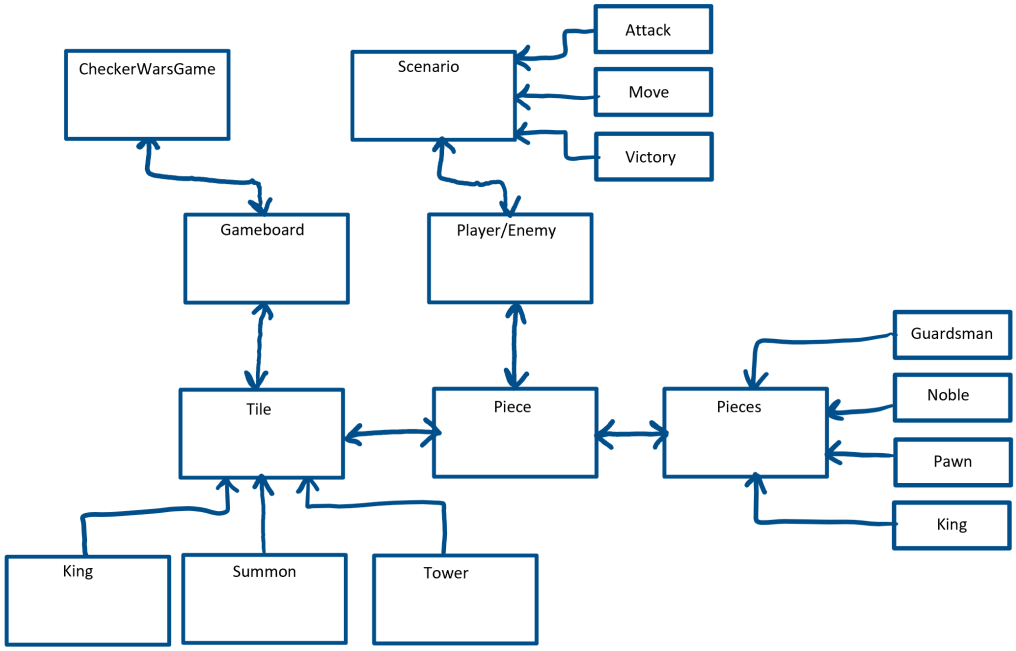
24d Initial Subsystem Decomposition

Scenario is now split up into the three possible moves that extend from it: Attack, Move, and Victory. The Scenario class is in charge of taking the move executed by the Player/Enemy class and executing that move on the server.

Pieces now extends Guardsman, Noble, Pawn, and King pieces. The Pieces class determines which pieces move and are represented on the gameboard for both users.

The Tile class now extends King, Summon, and Tower spaces. The tile piece is responsible for determining which spaces are occupied, which spaces are special tile spaces, and communicating this information with the server.

The CheckerWarsGame and Gameboard classes hold information about the games played by each of the users, and the state of the current gameboard.



25 Additional Design Considerations

25a Hardware / Software Mapping

No hardware will be needed (except for the server infrastructure, which will be rented initially so that we do not have a big outlay on server costs before we know if the game will be successful.)

25b Persistent Data Management

The serves that hold all the data for the game will be located in data warehouses and will be online 24/7. The system will run in a distributed fashion, so that a single server failure will not result in the loss of data.

25c Access Control and Security

Users will never have direct access to the data on the servers. Only the developers will have access to that data, so access control and security is not a big concern. Standard industry security protocols will be sufficient.

25d User Interface

Below is a diagram of the layout of the board:

- T represent the Tower tiles.
- K represent the King tiles.
- S represent the Summon tiles.
- Red triangle represent the Guardsman.
- Purple square represent the Nobles.
- White dots represent the Pawns.

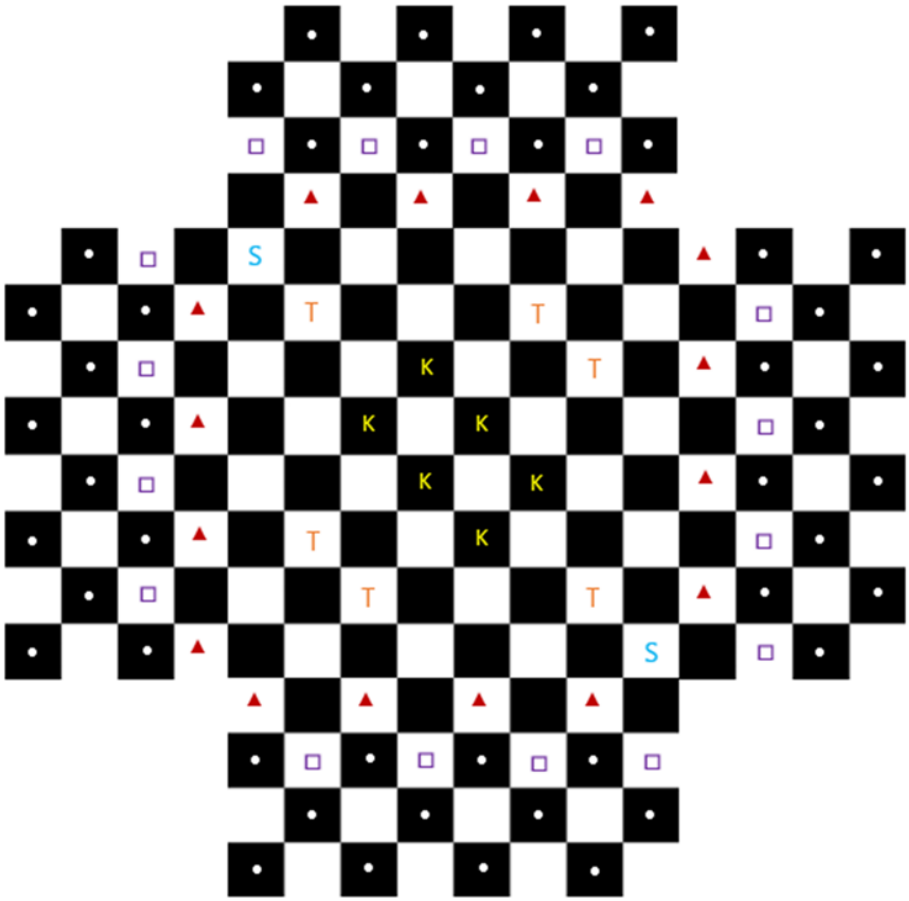


Figure 7: Checker Wars GUI

26 Final System Design

Below is the completed UML class diagram of the game detailing how all the pieces, players, and gameboard are connected together.

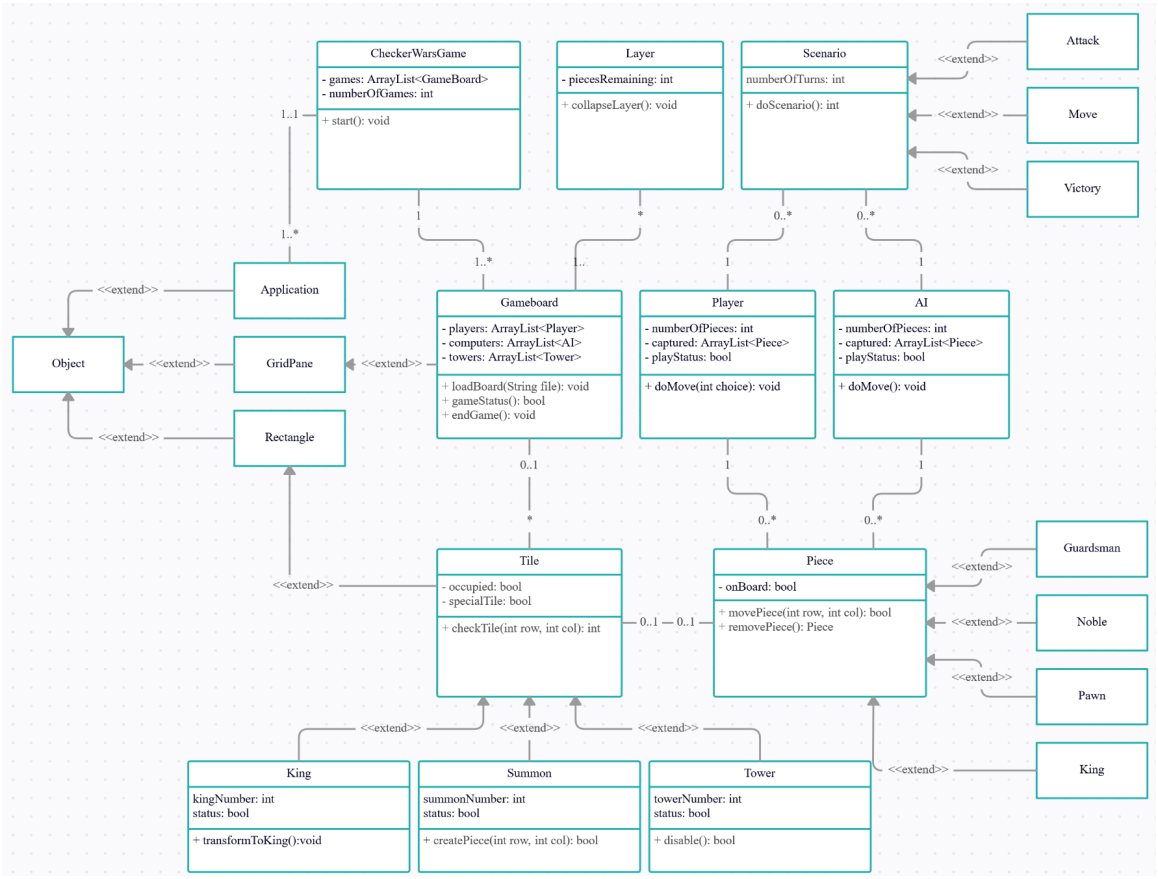


Figure 8: Final System Design (UML)

27 Object Design

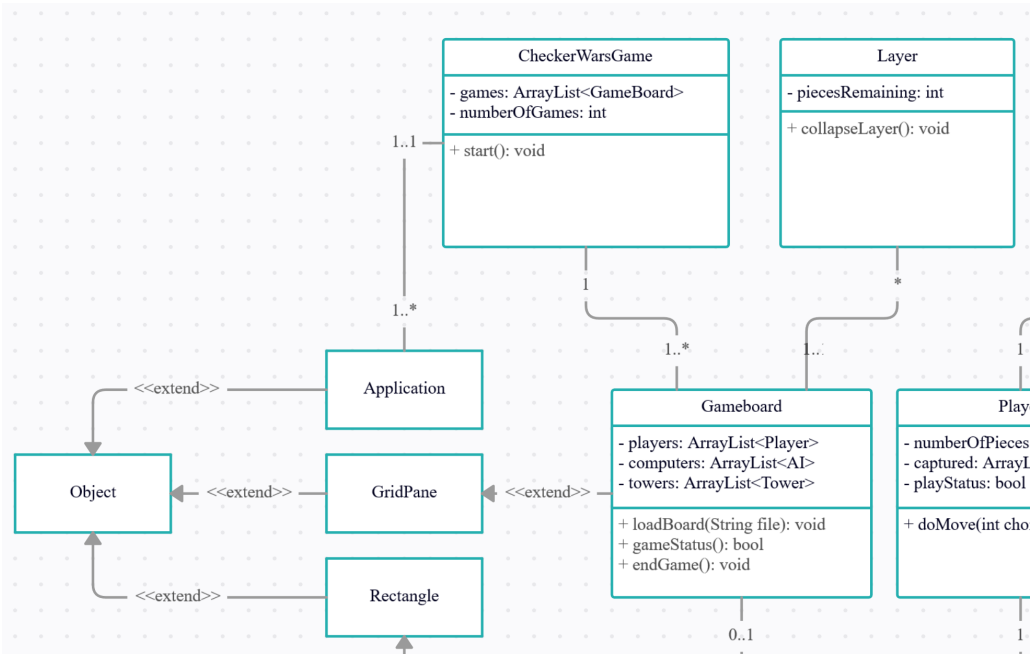


Figure 9: CheckerWarsGame and Gameboard classes

CheckerWarsGame Class: This class contains private members games, which is an ArrayList of GameBoards, and numberOfGames which is an integer counting the number of games in the match. It also contains the public method start, which will begin the game.

GameBoard Class: This class contains 3 private members, players which is an ArrayList containing the players of the game, computers which is an ArrayList containing the AI players (if any) of the game. And towers which is an ArrayList containing the Towers in the current gameboard. This class contains public methods, loadBoard, gameStatus, and endgame, which are self-explanatory. This class extends the GridPane class, which extends object. This GridPane represents the gameboard to the users playing the current game.

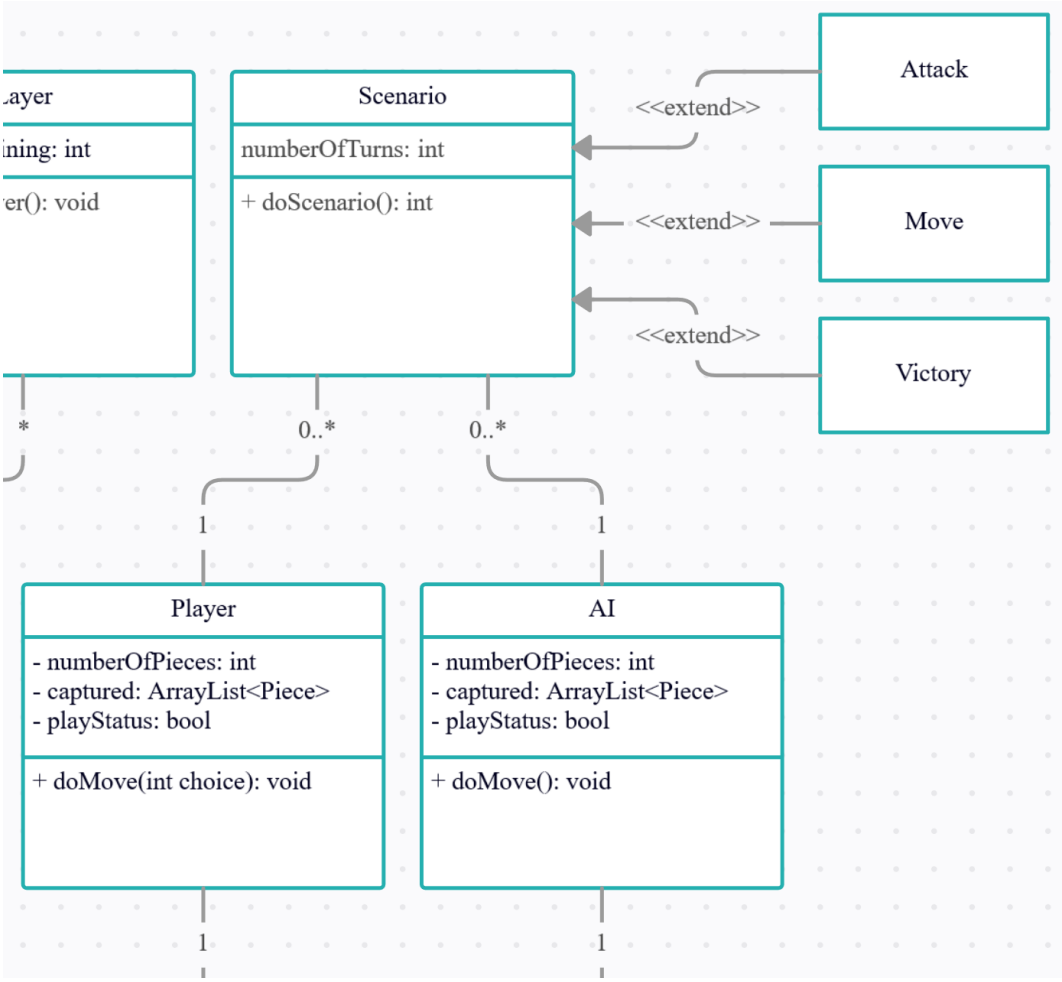


Figure 10: Scenario and Player/AI Classes

Scenario Class: This class contains the private member numberOfTurns which counts the number of turns played so far by each player. It also contains the public method doScenario which extends Attack, Move, and Victory.

Player/AI Class: The player and the AI class contains the private members numberOfPieces, captured which is an ArrayList of which pieces have been captured,

and playStatus, which is a Boolean value representing which players turn it is in the game. It also contains the public method doMove which represents the move the player is executing.

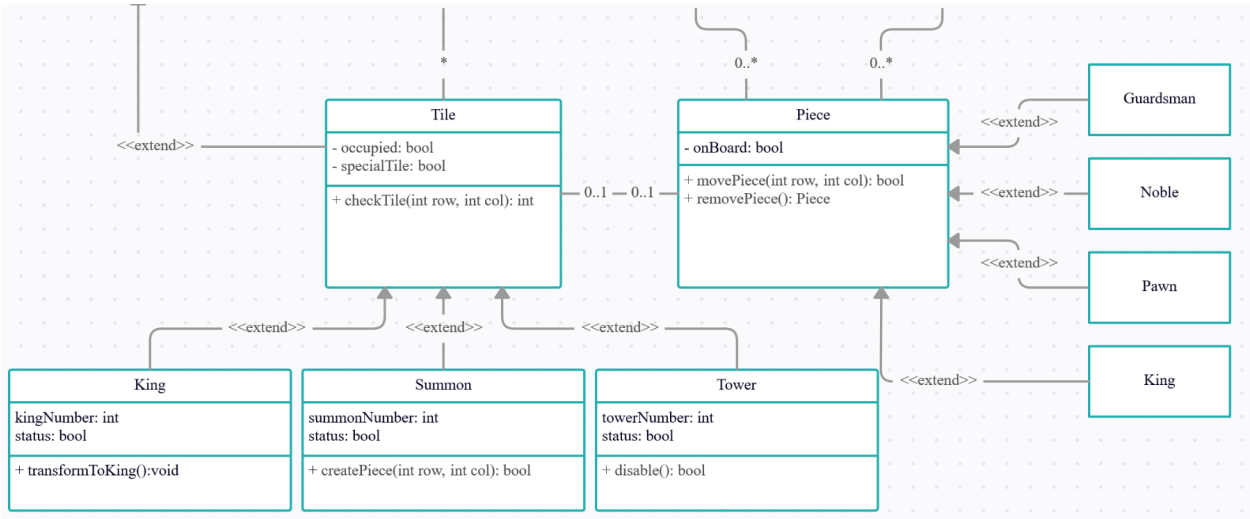


Figure 11: Tile and Piece Classes

Tile Class: The Tile class contains the private members occupied which is a Boolean value showing whether that tile is occupied by a piece, and specialTile, which is a Boolean value indicating whether the tile is a special tile piece (King, Summon, or Tower). The Tile class contains the public method checkTile, which checks what piece is in the current tile.

King Class: The King class extends to the Tile Class. It contains the private members kingNumber and status. It contains the public method transformToKing which transforms the piece to a king if the piece lands on the King tile.

Summon Class: The Summon class extends to the Tile Class. It contains the private members summonNumber and status. It contains the public method transformToSummon which, if a king or a noble land on a summon space the player is able to summon a new pawn piece on an adjacent black square.

Tower Class: The Tower class extends to the Tile Class. It contains the private members towerNumber and status. It contains the public method disable, which determines if the tile is enabled or disabled. If a king or a noble land on a Tower tile, they occupy that tower. If three towers are occupied by a single player, that player is victorious.

Piece Class: The piece class contains the private member onBoard which is a Boolean value that indicates if the piece is on the board. It contains the public methods movePiece and removePiece, which are self-explanatory. The Piece class extends the 4 possible pieces, which are Guardsman, Noble, Pawn, and King.

IV Project Issues

28 Open Issues

When dealing with issues that we have had with our project, we took a bit more of a pragmatic approach. One of the main issues that we have discussed about involved the usage of user accounts and what that may entail. In modern times most games, if not all, have included some type of system dealing with user accounts or a digital rights management system. There are already many pre-established DRM programs that are very widely used such as Steam, Uplay, Battle.net, Epic Games Store, and Origin which may tailor to games from independent studios. On consoles, this is not an issue since every user that plays online on a console requires a user account which can be seamlessly add to the game, but on PC it's a completely different story. If our game becomes mainly browser-based, if a user wants to keep track of their long-term record or progress in a single-based game should be retained in the case of having a user account. Also, if we decide to publish it on a DRM or non-DRM service, then we need to think of ways to link up all the current user accounts with the players of the games. In certain cases, this user account must be linked to any in-game purchases the user may have. We have not yet come to a consensus on how the design and implementation of the cosmetic purchases will work.

There have also been issues thinking up security concerns as a result. If the user has a user account, then there should be some type of security to protect the user's data as well as their opponent's data if they are playing online. Another issue with security would be game piracy. If the game ends up with a DRM service, this is not an issue. However, a browser-based game needs to have certain security measures in order to deter hackers which may alter gameplay and users that will potentially scrub data about the game off the website. Also, if we decide to allow players to save card numbers, we need to think of a way to protect that information from leaking to outside parties.

In terms of legal issues, the game may incorporate an in-game chat. As a result of that there needs to be some way to ban unruly players and potentially block harmful communication between players. At Bohn Jell Entertainment, we respect each and every player's background regardless of gender orientation, race, or culture and we want to avoid potentially having a community of players that is harmful or antagonistic. We also need to make sure that our game does not use any copyrighted material without the owner's written permission. This will apply to in-game cosmetics, as well as, any soundtracks or images used within the game.

Another issue to be addressed is how we are going to deal with optimization. Some companies may choose to optimize a game for a specific platform and port that version to another platform. We do not want to have players that are discouraged from playing Checker Wars on any platform that it's released on just because the game does not run well. For the most part, the game will not consume much of a computer's performance, however, we do not want to give an unfair advantage to computer players versus console players if we decide to make the game multi-platform, released on different platforms, and with cross-platform multiplayer, users on different platforms will be able to play with each other. As such we need to consider optimizations depending on

the operating system or platform and general stability depending on system architecture.

Distribution of updates may be a unique issue as well. If there are general reports of system instability on a specific platform, we need to think about whether that may translate to other platforms as well.

29 Off-the-Shelf Solutions

As said in the previous section, any user account information protection or purchase information can be protected if the game is released on a digital rights management program. Since all DRMs require the user to register an account in order to utilize their game purchases, using a DRM would circumvent all of the previous issues with the user-account and most of the issues dealing with security and in-game purchases.

For the most part, consoles utilize their own DRM systems, however on a PC, the user may choose to use an existing DRM service or utilize the browser version. A DRM generally has ways of dealing with hackers as well. If the user decides to use the browser version, instead of having to program all of the security and user account systems, we can outsource that programming to companies such as LEAN Security which specifically deals with online game security. Any security dealing with purchases made in-game on the browser version could also be outsourced to another company such as Norton or PayPal.

29a Ready-Made Products

User Accounts, In-game Purchases, Security: Any DRM solution such as Steam, Uplay, Epic Games Store, GOG, Battle.net, Bethesda Games Store, Origin

By using a DRM service for PCs or Macs, this would significantly circumvent many issues. However, DRM services are not a fool-proof way to deter pirating of games, requires first-time users to download a service that they may not otherwise use, and may cause the price of the game to rise due to the fact that a DRM service will take a specific percentage of all game and in-game purchases. Also, by using a DRM service, we are giving up total control of distribution and direction of the game.

Online Browser Game Security: Lean Security, Norton, PayPal

By using these systems, this would reduce the need to prepare most programming for in-game purchases and potentially user accounts. However, effectiveness of these services is questionable at best. Most of these services would contain all the user's data so if that company is involved in some sort of controversy, it could affect Bohn Jell Entertainment's image and vice versa.

Console In-game Purchases: Sony PlayStation Store, Xbox Game Store, Nintendo Store

For the most part, these services are fool proof. They would deal with in-game purchases and these consoles already require a user to create an account. There would

additional cost dealing with publishing on the respective consoles much like the DRM services.

29b Reusable Components

Any program design or libraries that deal with the project's compatibility with any outside programs such as DRM may be reused in future projects. Also, if instead we design a system for dealing with user accounts, in-game purchases, and security may be reused. If we decide to release a sequel, expansion, or downloadable content for the game, the main games mechanics may be reused with additional features added on.

29c Products That Can Be Copied

At the core, the idea and mechanics of checkers is not copyrighted since the game is thousands of years old so any ideas pertaining to the game is okay to pursue. Also, during previous projects, we have already developed a version of checkers which is working. The issue is we would need to overhaul the core functionality and refine the game to make it appeal to more audiences. Also, the program was written in C++ which would need to be translated into a more appropriate language. Mobile platforms, consoles, and libraries that deal with compatibility of web design programming languages are more compatible with a programming language like Java. While porting over the game from one programming language to another, improvements to the game's code could be made at the same time.

For the most part, our program does not have any competitors with which to base our programming on since a multiplayer, checkers-based game has not been developed before and as such we cannot borrow any ideas. With that being said, certain ideas can be borrowed from RTS, real-time strategy, games like Warcraft or from grid-based games like Crypt of the Necrodancer.

As for the images and theme of the game, we would have to develop those from scratch even though we could use a game like 3D Tabletop simulator for inspiration.

For the most part, by developing our own assets we do not have to worry about copyright, however, by using other games as inspiration, we could cut the production of the game by up to 20 percent or more.

30 New Problems

30a Effects on the Current Environment

Since this project is our first project, we expect that this game will present new problems in the workplace. Although we have general plans of how development will go with this project, overall, our process has not been tried and tested. As such we expect that at first, we will have some issues and changes that will be made to the process of development during the actual development of this first product. The workplace will definitely be more stressful because extra effort will need to be taken in order to find out a proper way to maximize efficiency of the workplace.

In terms of budget and budgeting, we are seriously considering hiring a personal accountant and a financial planner. Currently, no one on the team has a solid background in finances and bookkeeping so we may need to outsource all of this to a third party. This should be no issue since our company is still in infancy, but there is still some anxiety in thinking that this game will be our own sole source of income after development. As such we need to plan accordingly preparing for the worst case and cutting our losses and hoping for the best. Our employee pay will be of utmost importance, but there is an inherent risk that if the game does badly, then considering bankruptcy may not be such a bad choice since our financial portfolio is not diversified enough. This is where a competent financial planner and an accountant could rectify this possible issue.

As of right now, we expect that reaching the deadline should be no problem with the current set of employees, however, there may end up being setbacks during development. This could mean that a rather colossal bug may end up being found or that more staff would need to be hired. As a result, we may have to resort to hiring more staff or a specialist if any major issues occur.

Changes in scheduling may be required in order to accommodate sick days or vacations that an employee may require especially with the current pandemic. In order to deal with this, we have made our development process with a much more conservative estimate. Theoretically, we could finish the total development project up to 1 month quicker than our current deadline, but that is assuming that the development process is without any issues.

Even with a conservative deadline with development, there will still be possible issues with distribution. Overall, we project that the majority of game purchases will be made digitally, however, for those wishing to purchase a physical copy, an entire distribution network will need to be sourced in order to get our product on store shelves.

30b Effects on the Installed Systems

Since we are implementing our product on a variety of platforms, our product needs to be of the utmost quality and consider all users' computer systems and platforms. In terms of consoles and mobile development, optimization of the game on these platforms should not be an issue. Since over 95 percent of all users are using a more recent system update on these platforms and since generally speaking, backwards compatibility of older updates is generally not an issue, we should not have any issues on these platforms. Website development of the product should be rather straightforward. However, it is a bit of a different story with PC and mac development.

In terms of mac and PC compatibility, looking at other applications, certain applications are compatible with only the newest iterations of the operating systems and others are only compatible with older iterations of operating systems. In order to ensure a maximum coverage of users we will support Mac users with the Mountain Lion macOS update and newer and for windows, windows 7 and later. This would roughly cover all users with updates from 2015 to now which would be a vast majority of users. In terms of Linux users, this application work with all Linux kernels.

For the future, there may be an issue with using the application with an ARM-based CPU, but overall, it should not be an issue. This is brought up because Apple may decide to switch to ARM based CPUs. Since our application will not require heavy performance, working on an ARM architecture CPU should not be an issue despite the extra requirement that the ARM would have to use an x86 emulator. Despite this if ARM chipsets become more widely used, we may have to apply a future patch to optimize the game for this architecture.

30c Potential User Problems

For the most part, we will try to optimize our game to all intended platforms. This will take extensive and arduous development, but in the end, should end up with no user complaints.

Users need a way to contact a customer service team if any issues arise which could be rectified.

The most important issue that we will try to address is the game refinement. There may be some unseen, potential issues with stability and performance as our sample size during testing is not that major.

Also, we are planning on requiring all users to register an account or have an account on a DRM service that the game is available on. This would mean that we would alienate potential customers that may not want to create an account or download a DRM service in order to play the game.

Another issue is that users may affect each other while playing a game. Certain boundaries should be made to limit the ability of a user to harm another user or obtain another user's information.

Lastly, a potential issue may arise with cosmetics or with certain graphics. There are over 50 million people in the world with epilepsy and many more that may be adversely affected by flashy images. As such much care must go into the creation of new visuals and a warning screen should be implemented to warn potential users of harm from visuals or long-term play.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

As said in a section above, our development process is going to have to remain a bit more fluid since this is our first product and no adamant plan has been made. To mitigate this, we could seek outside help or advising from third parties. Since the game is being made to be online, multiple servers will need to be implemented in order to handle online traffic. Also, internet service stability may or may not be a potential issue in the future. Other companies may plan to have servers in a multitude of states or countries, but as of right now, Bohn Jell Entertainment will not have that luxury, so contingency plans and redundancy may need to be built-in in case internet service or power goes out.

The number of servers to handle the online traffic may not be enough and as such may require outsourcing.

As it stands, there may not be enough employees in order to handle all jobs efficiently.

Distribution of the physical product has not been planned yet and will require extensive thought on which stores we can partner with.

Also, the location of the company may be an issue to certain employees that will commute to company grounds.

30e Follow-Up Problems

If the product performs well, then we may have a potential issue with online stability due to either server problems or due to the application not working well with too many clients connecting to the server. Especially since we plan on the game allowing for cross-platform play.

DDOS, distributed denial of service, attacks are very common and as such, we may not be able to deal with these without outside help since our staff may or may not have experience with online security.

The advertising budget may not be suitable to get a quick influx of new players. As a result of a lack of advertising, which is often one of the biggest expenses for large scale products, we may be losing out on many potential customers.

As a result of a lack of an advertising budget, most likely, the initial influx of users may be slow and the popularity of the game as a result will slowly rise.

Since this product will be our first and only product for a while, spending will need to be very conservative and as such may bring unintended effects on employee morale.

Localization could also be a potential issue depending on the quality of translation of the product. As a result, certain messages within the product may confuse the users.

31 Migration to the New Product

Not Applicable

32 Risks

- Not enough servers to handle the initial volume of users. Based upon our advertising budget and game genre popularity, probability would be low of it becoming a major issue.

- Filing for bankruptcy as a result of low sales and insufficient budget. Probability would be very high of it becoming a major issue.

- Understaffed as a result of poor planning would have a medium probability of it becoming a major issue since we have already implemented conservative due dates.

- Low quality within the product causing user frustration. Probability would be low because all the quality assurance would be handled by all senior members of the programming teams.

- Slow and gradual influx of users leading to slow revenue. Probability would be low of it becoming a major problem.

- Encountering a large bug that may hinder project progress. The probability of it becoming a major issue may be high or low depending on whether the source of the bug is known or not.

- Encountering a legal issue dealing with copyright would have a low or high probability of it becoming an issue depending on how important the graphic or mechanic is to the overall design of the game.

- Language and translation issues is a potential risk, however, the probability of it becoming a major issue would be low.

- Bad performance due to system incompatibility or due to bad optimization would be a high probability of it becoming a major issue.

- Product instability with the system's operating system would have a high probability of it becoming a really bad issue.

- A potential hacking would be a low to very high probability of it becoming a major problem. At the bare minimum, one user's game may be compromised due to an opposing player hacking the gameplay mechanics. At the worst, user information is leaked by the web-based version or physical version of the game.

- Problems in the distribution chain of the product. The probability of this becoming a major problem is low since the digital version should not have any problems at all and we project that most users won't use the physical version.

- Another company infringing on our own property would be a low to high risk since a larger company would have no issues with distribution and advertisement which would lead to a disadvantage to Bohn Jell Entertainment, and a small company would generally not affect the situation at all.

- Management malpractice and controversy would be a high probability of this becoming an issue because it may cause the understaffing problem to be an extra issue.

- An employee calling in sick or taking a longer absence of leave would be a low probability of becoming an issue since our deadlines already account for this.

- Low productivity would be a high probability of becoming a problem since this would hinder development and also it would contribute to a lesser workplace environment.

- General logistical issues such as scheduling conflicts would have a high probability of becoming a major issue since it would affect productivity and would increase the amount of stress in the workplace.

- Bad budgeting would have a high probability of becoming an issue since it would not only hinder staffing, but it may also cause a large general decrease of efficiency in the workplace.

- Low advertising budget would have a high probability of becoming an issue since much of the target audience would remain untapped since they would not know if Checker Wars even exists. Also, it may give certain users the wrong idea of the game if it is marketed falsely.

33 Costs

- Based upon preliminary results and based upon average costs for independent video games our initial estimate for the total cost of development was $80,000 at the minimum to $160,000 at the higher end.

- We estimated that the total time of development would be 2 months at the maximum which is a rather conservative estimate which would consider any potential major setbacks and employee grievances during development. If we break this estimate down further, we estimate that it would take roughly 3 weeks to work out the main game mechanics, an extra 2 weeks to deal with programming of user account systems and purchase systems, and the rest 3 weeks dealing with distribution, optimization, and general logistics such as advertising, hiring of new employees, and budgeting.

- Since our advertising budget is small compared to the loss of opportunity would cost an estimated $10,000 to $30,000 in profits. This would be due to not hitting the full potential of the target audience.

- Independent of game development, the addition of new servers at the company to accommodate more online players would cost around $15,000 to $10,000.

- In terms of function points, development of the mechanics of the game cost 2500 function points, 1000 for game logic, 1000 for system compatibility and optimization, and 500 for user interface design. The in-game marketplace for cosmetics took an additional 1000 points and the client-server system design took 600 points. Cross-play implementation was 800 points. This led to a total of 4900 function points.

- Pay per employee ranged from $1,000 from the most basic employees up to $10,000 for the software developers involved in the project.

- Additional cost that were dealt with involved rent and utilities costs which averaged between $2,000 to $4,000 per month.

- Costs to utilize DRM systems ranged between $100 for Steam to $20,000 for PSN, PlayStation network, which included a dev-kit for $15,000 and right-to-publish for $5,000. The website version costed approximately $200 to $300 as website creation costed between $100 to $200 and upkeep costed $50 dollars a month. For physical copies, 300 were made and with an average cost of $27 to produce and ship to physical store, ended up costing $8,100.

- To deal with potential legal issues, a third-party legal team was hired which cost around $2,000 to $4,500 which was set at an hourly pay.

34 Waiting Room

Quite a few ideas for future implementation of the project were considered while creating this product.

- Creating a 3D version of the project with an upgraded graphical interface.

- Allowing user-generated content in order to change certain aspects of the product including the look of Checker Wars and/or the mechanics behind the game.

- Creating a full single-player mode complete with story points, unique scenarios, and potentially specially made heroes which would have their own set mechanics and skills which they can use to change up the game. Possibly with fully voiced dialogue.

- A hero mode in which a multiplayer match may include special characters which have special skills and may or may not be randomly selected by players.

- Possible AI intervention to balance out player versus player games.

- New pieces and tiles that would perform their own special tasks.

- More premade maps which will be balanced and each of different sizes.

- A user-creation mode where a user can create their own levels complete with their own scenarios.

- A randomizer which would completely randomize aspects of the map to allow players to have a unique experience every time they play.

- Loot boxes in which the user can purchase a box in the store to obtain rare cosmetics.

- A marketplace in which users may perform transactions pertaining to rare cosmetics.
- A ranked mode that will keep track of the players overall record and rank them accordingly.
- Special events which will have their own activities to obtain rare cosmetics.
- Maps which may have multiple areas within areas.
- AI-driven enemies which may attack either player.
- Item-based gameplay in which the players can allow items to appear in-game which may change gameplay.
- Unlockable secrets and Easter eggs which will give more incentive to players who play the game longer or for players who discover other aspects of the game.
- Achievements and unlockables for return players of the game.
- Incentives for players who come back to the game after a long period of time.
- Seasons in which game balance may change.
- An RPG, role-playing game, mode where certain pieces may have a level which will increase with each piece that piece removes. These pieces would become stronger or tougher after each level-up.
- Free incentives to long-term players depending on how many games have been played.
- A puzzle-mode which would have scenarios where the pieces are setup and the players would have to determine the best course of action to clear the board.

35 Ideas for Solutions

Although many issues in developing have been found thus far, certain present and future issues may be addressed.

- Although we are already using a client-server system design, a model-view-controller design or event handler design may be appropriate as well to help develop the multiplayer mode which would wait for input from the user and update as needed.
- A decorator design pattern may be useful for the design of the UI and a factory method design pattern may be useful for spawn spaces.

- In order to get Java to work with other programming languages, other plugins or libraries may need to be used which will seamlessly merge two languages into a single project.

- Object-oriented design will be a given since Java is being used, however, it may also aid in security as Java has quite a few built-in libraries and object-oriented design may help limit the user to only access specific part of the game.

- CRC cards may be used in order to aid in the object-oriented design and to think of new ways to create the methods of the classes.

- JavaFX may work well as a GUI design tool; however, this game could also utilize certain pre-built videogame engines such as the unreal engine.

- Since we have added more servers, a more intensive client-server test should be implemented before release of the product.

- Usage of IntelliJ or Microsoft Visual Studios Pro may aid in the productivity for development.

- Using pair-programming may be a better choice since most programmers at the company are still at a lower experience of programming.

- Creating smaller modules one-at-a-time will increase cohesion and decrease coupling allowing for ease of debugging, flexibility, and reuse of modules.

36 Project Retrospective

In terms of what worked well and what didn't right off the bat, it was a bit too risky to base the company's future off of one product even though the budget for a small company is low. Certain budgeting could have been done such as limiting the platform availability strictly to Mac or PC users. That would allow for a much larger portion of the budget to be allocated to other sections that may be more important such as to advertising. Also, this would allow much of the team resources to be allocated to more important parts since less updates, optimizations, and program porting would have to be dealt with.

With the company being as small as it is, it would still have been nice to hire employees to create and join other teams such as an in-house QA team, advertising team, financial team or legal team. This would greatly reduce the amount of overhead dealing with third parties.

The allocation of the actual programming of the product worked out very well and this sort of architecture could possibly be honed and used again for future project. The idea to setup a conservative schedule was a great one since it reduced stress in the workplace and increased productivity as it allowed employees to be a bit more flexible with how they approached their work.

More employees would have allowed the company to work on multiple projects concurrently which would allow for more work to be done. It would also reduce overwork in the workspace which is very common in the videogame industry today.

Planning out the project's logistics in advance would have made the situation a lot smoother as the fluid structure of development lacked a solid foundation in which the employees could rely on. As such, the employees needed to be extra flexible in dealing with various issues.

Adding more servers to accommodate more players was definitely a good idea. At the bare minimum, we would have had underutilization of servers which would have been a small issue compared to having the game be unplayable due to not having enough servers.

Overall, the production of Checker Wars was very rocky to put it bluntly due to multiple issues in creating a new company, but this first project acted a solid platform in which to refine the process in building future applications and videogames.

V Glossary

AI: Artificial Intelligence.

Bohn Jell Entertainment: The name of the company that developed the game!

Rank: A player's rank among other players of the game, both locally and globally.

Score: A player's individual score at the end of every game.

GUI: Graphical User Interface.

Twitch.tv: A popular online streaming platform.

e-sports: Electronic competitive video games considered sports by the community.

DRM: Digital Rights Management.

IO Devices: Input/Output devices.

CD: Compact Disk for storage of media.

GOG: Good Old Games, a digital videogame storefront.

Uplay: A digital videogame storefront owned by Ubisoft.

Epic Game Store: A digital videogame storefront owned by Epic.

Steam: A digital videogame storefront owned by Valve.

QA: Quality Assurance.

VI References / Bibliography

[1] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

[2] Robertson and Robertson, Mastering the Requirements Process.

[3] B. M. C. L. O. B. Arthur Mezheritskiy, "Dungeons and Dank," 2018.

VII Index