

Virtual Board Game Engine

Seng499 Design Project II

July 31, 2020



University
of Victoria



Group ID: 37

Faculty Supervisor: Dr. Miguel Nacenta

Team Members:

#	Name	V Number	Program
1	Juan Cormish	V00839267	SENG
2	Oleg Matveenko	V00844836	SENG
3	Michail Roesli	V00853253	SENG
4	Emily Sluis	V00865746	SENG



Acknowledgment

We would like to extend our sincere thanks to our Project Supervisor, Dr. Miguel Nacenta, and our Course Coordinator Dr. F. Gebali. Additional thanks go to the UVic Software Engineering and Computer Science departments and their Technical Staff, and Teaching Assistants. Thank you to Tradable Bits for allowing us to use their Nice Objects in our code base. Finally, a warm thank you goes to our family and friends.

Executive Summary

The report details the design process of a Virtual Board Game Engine. The purpose of the project was to develop a fully customizable, online board games space that allows users to remotely play any playing card-based game. The design process of the project followed Requirements Engineering and Iterative Design methodologies. The primary design objectives were that the platform should be available to anyone with an internet connected device, should accommodate 1 to 8 players, and should work with any card game. The platform should also be intuitive to new users. Finally, the platform infrastructure should be implemented to allow for future scaling.

The literature survey found four products available online with overlapping scope and functionality to the project goal. These are [playingcards.io](#), [cardgames.io](#), [tabletop simulator](#), and [boardgame.io](#). Though exploring each platform, additional features were discovered and considered such as macros, token counters, logs, and 2d and 3d physics engines.

At the outset of the design process, team members were assigned roles, tasks, and specialties to ensure balanced workloads. However, these roles shifted as the project continued and team members fine-tuned their skills. Throughout the design process, the focus was on Human Computer Interactions forcing design decisions to be based on the user experience, usability, and intuitiveness rather than ease of implementation.

The final platform design has three main pages: a home page, a design page and a play page. On the home page the user can log in and choose to join or host a game. The design page allows a host to set up the board in the appropriate configuration for their game. Play areas are selected intuitively within a grid with a single click. The play page mimics the layout of people playing cards around a table as they would in person. Actions available to a user are displayed in the right-side bar, are context sensitive, and depend on which cards, opponents, or play areas they have selected. Users can assign the turn indicator to each other, but the system intentionally does not constrain turn order or timing.

Testing was done at every step of the design process using storyboards and use case analysis. Usability testing was performed upon completion of the implementation. Based on the testing and supervisor feedback, further work should include dealer functionality, customization of decks, a token count, hand reordering ability, and quality of life changes such as resetting the game board and allowing users to design macros specific to their game.

At the time of writing, the first version of the platform has been fully implemented and can be played locally. The code base is hosted in a private repository on GitHub. A full demonstration of gameplay is included in the project video.

Contents

Executive Summary	2
List of Figures	4
Glossary	5
I Introduction	6
II Project Goal	6
III Design Objectives	7
IV Literature Survey	8
IV.I Approaches	8
IV.I.I Playingcards.io	8
IV.I.II Cardgames.io	9
IV.I.III Tabletop Simulator	10
IV.I.IV Boardgames.io	10
IV.II Selected Approach	11
V Team Duties & Project Planning	13
VI Design Methodology & Analysis	14
VII Final Design Details	21
VIII Testing & Validation	24
VIII.I Storyboards	24
VIII.II Usability Testing	24
VIII.III Code tests	24
VIII.IV Test Plan	24
IX Discussion & Recommendations	25
X Conclusion	26
References	26
Appendix A: Design Sketches	27
Appendix B: App Screenshots	34
Appendix C: Full Overview UI Sketch	37

List of Figures

Figure 1: PlayingCards.io custom play page	8
Figure 2: Cardgames.io playing hearts with hints	9
Figure 3: Visual demo of physics engine using cards [3]	10
Figure 4: Overall Activity Behaviour UML Diagram Version 1	15
Figure 5: Game Design Activity Behavior UML Diagram Version 1	16
Figure 6: Gameplay Activity Behavior UML Diagram Version 1	17
Figure 7: Card Game Class Structure UML Diagram Version 1	18
Figure 8: Example paper sketch of game creation	19
Figure 9: Example storyboard for Crazy 8's	19
Figure 10: Overall Activity Behaviour UML Diagram Version 2	22
Figure 11: Gameplay Activity Behavior UML Diagram Version 2	23

Glossary

Assets	An entity used in board games such as player pieces
COVID-19	Coronavirus; an infectious disease
HCI	Human and Computer Interaction
Macro	An input sequence mapped to an output sequence. Allows for repetitive actions.
Storyboard	Illustrations visualizing interactions with application
UI	User Interface
UX	User Experience

I Introduction

In our free time, we enjoy playing games to pass the time and to connect with family and friends. Due to the COVID-19 pandemic, it has become more difficult to meet with friends and family while also maintaining social distancing. While it is possible to play some games online, it can be difficult to find particular games, especially if they have several house rules. To address this problem, we aim to provide a solution to play any card game virtually while not restricting the user to certain actions, so that they can play while social distancing and with any house rules they like.

While being able to stay connected with family and friends is important, we also found that this project also offered interesting problems in the area of Human Computer Interaction. The core problem was to develop an intuitive user interface (UI) while also providing an easy and flexible way to play and design games, and in our particular case, card games. This problem is applicable to many areas including game and web design, but it also has the potential to provide interesting solutions to various user experience (UX) problems.

This project aims to provide a platform for family and friends to stay connected, however we also foresee that there is potential for scaling this project globally sometime in the future.

II Project Goal

The goal of the project is to design a virtual board game engine to allow friends and families to play fully customizable card games with any unique house rules from the comfort and safety of their own homes.

III Design Objectives

In order to achieve the project goal outlined above, there are 5 primary objectives that need to be met. First, as the purpose of this system is to allow people to connect remotely over card games, the platform must be accessible to as many people as possible. This means anyone with an internet connected device and modern browser must be able to access the platform. In keeping with the goal, the platform must accommodate approximately as many people as one would usually fit around a table. So, the platform must accommodate at least groups of one to eight people.

The third objective was to design a flexible product that would allow users to play any games they could play with a deck of cards. The platform should simply provide a virtual game space for users to play niche family games or house rules of common games. The platform must track cards in the deck and in users' hands and indicate who's turn it is without constraining them to a rule set. The platform must also be intuitive so that users can read the tutorial pages, experiment with the play page, and have good enough understanding to play a game in 20-30 minutes. The play page must make it clear to users what actions are currently available to them and how to enact them.

Finally, the platform must be scalable for a future launch. If hosted on a sizable server, the system must be able to host a thousand concurrent games. The platform must be implemented with scalability in mind to avoid unnecessary bottlenecks. Assuming each of these objectives are met, the product will fulfill the project goal.

IV Literature Survey

In this section we'll identify several approaches in meeting our project goal followed by our chosen approach. Next we'll examine methods and approaches that helped meet our objectives for our chosen approach.

IV.I Approaches

In our research we identified several possible approaches which satisfied our goals for accessibility and customizability; they provide the ability to play online but don't restrict the user to playing only one variant of the game. Below we'll discuss these existing solutions including Playingcards.io, Cardgames.io, Tabletop Simulator, and Boardgames.io in respect to their card playing features.

IV.I.I *Playingcards.io*

Playingcards.io is a website which provides users with a way to play various 2D/top-down view board games. The user is offered a variety of possible games to play such as Checkers, Chess, Crazy 8's, Go Fish, Hearts, and Match Up. In the play area, users are able to move cards to any pixel location or hand, flip cards to hide them on the board, and shuffle decks. The user is also provided with the option of creating a custom game where you can make use of the sample assets used in any of their other games as can be seen in figure 1 below.

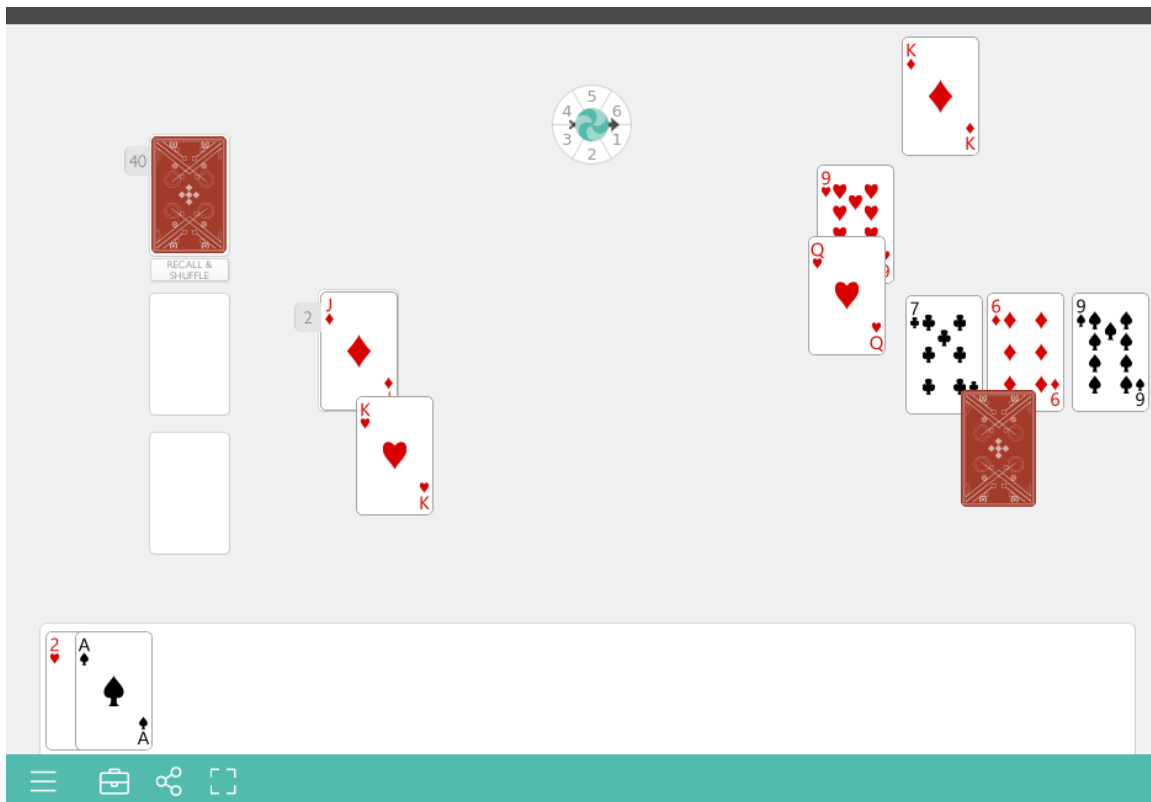


Figure 1: *PlayingCards.io* custom play page

There you can specify predefined locations for decks, which cards are used, and whether your hand is displayed. Once the games are ready, you can export the room to a file for quick setup in the future. You can then share the room code and play with anyone who has the link to the room. It also allows users to redesign a room mid-way through a game. One of the limitations to this implementation currently is that it does not offer dynamically growing deck locations, multi selections, public rooms, or any macros to enhance the experience of using a computer instead of playing in person.

IV.1.II Cardgames.io

Cardgames.io is another website which provides various 2D top down view preprogrammed board games with a focus on card games. It features 36 games currently including non-card games (i.e. Minesweeper, Checkers, Chess;) and card games (i.e. Crazy 8s, Cribbage, Spider, Solitaire, and Gin Rummy). When visiting their page, we noticed that they share similar core values of focusing on providing easy access to online games and that they also want to “help [people] play with friends and family during the COVID pandemic” [1]. In terms of multiplayer functionality, it provides people to create public and private rooms, but does not give the user any functionality to edit or create custom games so that house rules can be incorporated. Although it is not flexible in rule selection, it provides the user with helpful features such as macros, tips/hints, scores, and a stats page as seen in figure 2 below.

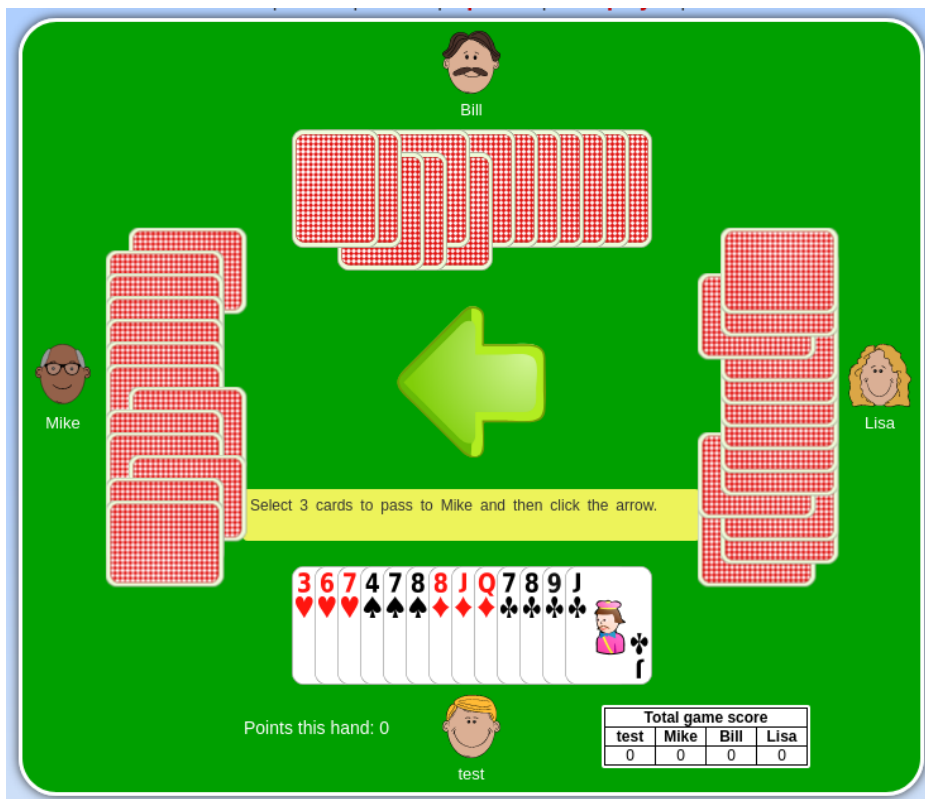


Figure 2: Cardgames.io playing hearts with hints

Although this approach does not satisfy the flexibility aspect we're seeking, these features would be much more difficult to implement and test if more customizability options were given to the user.

IV.I.III Tabletop Simulator

Tabletop simulator is a game on steam which provides a 3D environment for users to create their custom games or play from a variety of games. They provide users with a large sample of assets to use which gives users lots of flexibility in playing or creating their own games with their own house rules. On top of the existing assets, users can also "import custom assets, automate games with scripting, set up complete RPG dungeons, manipulate the physics, create hinges & joints, and of course flip the table when you are losing the game" [2]. A visual representation is shown of using cards in this 3D environment to stack cards as shown in figure 3 below.



Figure 3: Visual demo of physics engine using cards [3]

The whole purpose of the game is to attempt to simulate the real-world experience and then provide some flavour and helpful shortcut features. Some additional features include voice/text chat, mods to modify the simulator itself, joining and creating servers, and hotseat mode which allows users to play locally on the same computer. The limitations of this solution are that it is not accessible from anywhere on any device and shortcuts take some time to get used to in order to play as quick as you would in real life.

IV.I.IV Boardgames.io

Boardgames.io is an open source project on GitHub which provides the networking and storage code necessary to create online board games. It features game state management, lobbies, logs, and extendable plugins that can be programmed and added to

existing games. One interesting aspect of this platform is that it provides users with automatically generated bots to play games. This lets programmers test their app, and also users can play on their own. Although it provides a full platform for creating board games, few card games currently exist. The main purpose of this platform is to let users contribute and program their own games onto their base board game platform.

IV.II Selected Approach

In selecting our approach, we considered several quality attributes that we deemed most critical to our project; accessibility, understandability, portability, scalability, customizability, and extensibility. We also identified the following features in the existing approaches and methods.

- Grid vs Freeform card layout
- Macros & shortcuts for managing cards
- Public/private rooms
- Logs
- Plugins & Mods for adding features to existing games
- 2D vs 3D
- Voice/text chat
- Design custom games within the app
- Preprogrammed games
- Hints & Tips during play
- Open source contribution
- Program on the fly while game is being played
- Stats
- Scores
- Tokens & assets

The approach that we decided to go with is a mix of all of them. We decided to go with a 2D open source app that users could contribute to. We chose to do a 2D app because it is less complex and means we don't have to deal with as many factors when considering user experience design decisions. By creating an engine, we are able to expand on the customizability for the user which pushes us to making it open source so that it can be built upon. This also provides us with the potential of creating other games and to continue development past our prototype. For our engine, we valued the following features for our prototype:

- Grid: simplicity in design and implementation and allows for future accessibility design
- Macros: enhance the experience by using a device
- Logs: keep track of previous actions to provide info to user
- Design custom games within the app: easier for the general user who may not have programming experience

For our prototype, we looked into a more user experience focused solution and tried to create a game engine with the goal of being generic and flexible while still intuitive and easy to use. We also chose to use a web browser for our platform as this would allow more people to access this engine given that browsers are cross platform.

V Team Duties & Project Planning

At the start of the term, our team decided that in order to work more effectively, we would distribute roles in a similar manner to how it would be done in a small business Startup where each of our members are involved and in charge of a core aspect of the project. The roles were assigned as shown below.

- Front-end designer game play - Emily
- Front-end designer game design - Juan
- Back-end for media management (user uploaded cards, images) – Emily
- Back-end designer for general functionality – Misha, Juan
- Authentication/user designer – Oleg
- UML diagram manager – Juan
- Environment and maintenance specialist – Oleg
- Dev-Ops coordinator – Misha
- Faculty Sponsor Liaison – Emily
- Documentation Lead – Misha
- Database lead - Oleg

Each role takes into consideration some of the main deliverables we identified:

1. Project Planning & Selection
2. Selecting a Stack & Environment
3. Design Review
4. App & UI Design
5. Minimum Viable Product Implementation
6. Prototype Improvements
7. Final Overview

During the execution of our project, we found that although we were able to identify the key milestones in the project, during development it was much harder to make clear distinctions between components of the system. It was also especially difficult to divide work in the project where some members may not be as familiar with some development tools or lack experience developing certain components of the system. This made it harder to maintain a consistent product development rate. In response to these problems, we shifted roles in the team and worked more dynamically on various aspects of the project. In particular, we had to specialize and reassign roles in our team to be able to meet deliverable deadlines.

VI Design Methodology & Analysis

With the problem this application solves being primarily an interface for players, the focus in design was on human and computer interaction. Two methodologies were applied to the design process in an attempt to maintain this focus. Through using an iterative design methodology backed by Requirements Engineering the goal was to create a positive user experience.

The risk in designing applications like this one is the mistake of having implementation decisions negatively impact the design. To ensure design decisions were made to improve user experience and were not driven by implementation, it was critical to stick to the iterative design approach. Through this approach each design decision was first introduced, reviewed, mocked up, and then reviewed again. Every level of the design was analyzed and the reasoning behind design decisions was traced back to improvements to the user experience. Outputted by this process was activity diagrams, low-fidelity wireframe models, and storyboards.

The core features and goals of the application were initially laid out through the creation of activity diagrams as seen in Figures 4-6. These diagrams were kept and used only as guiding references and not as requirements of the design.

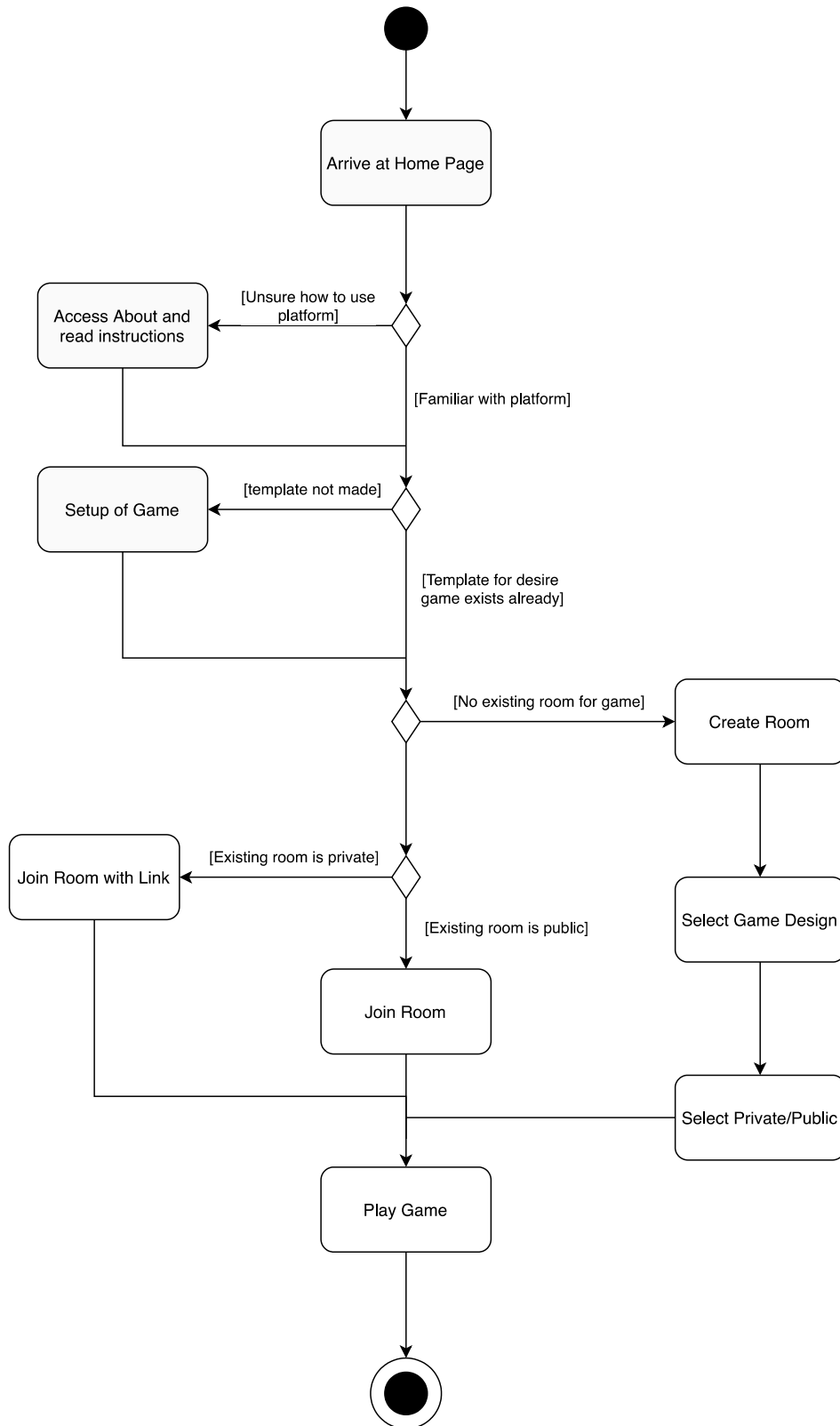


Figure 4: Overall Activity Behaviour UML Diagram Version 1

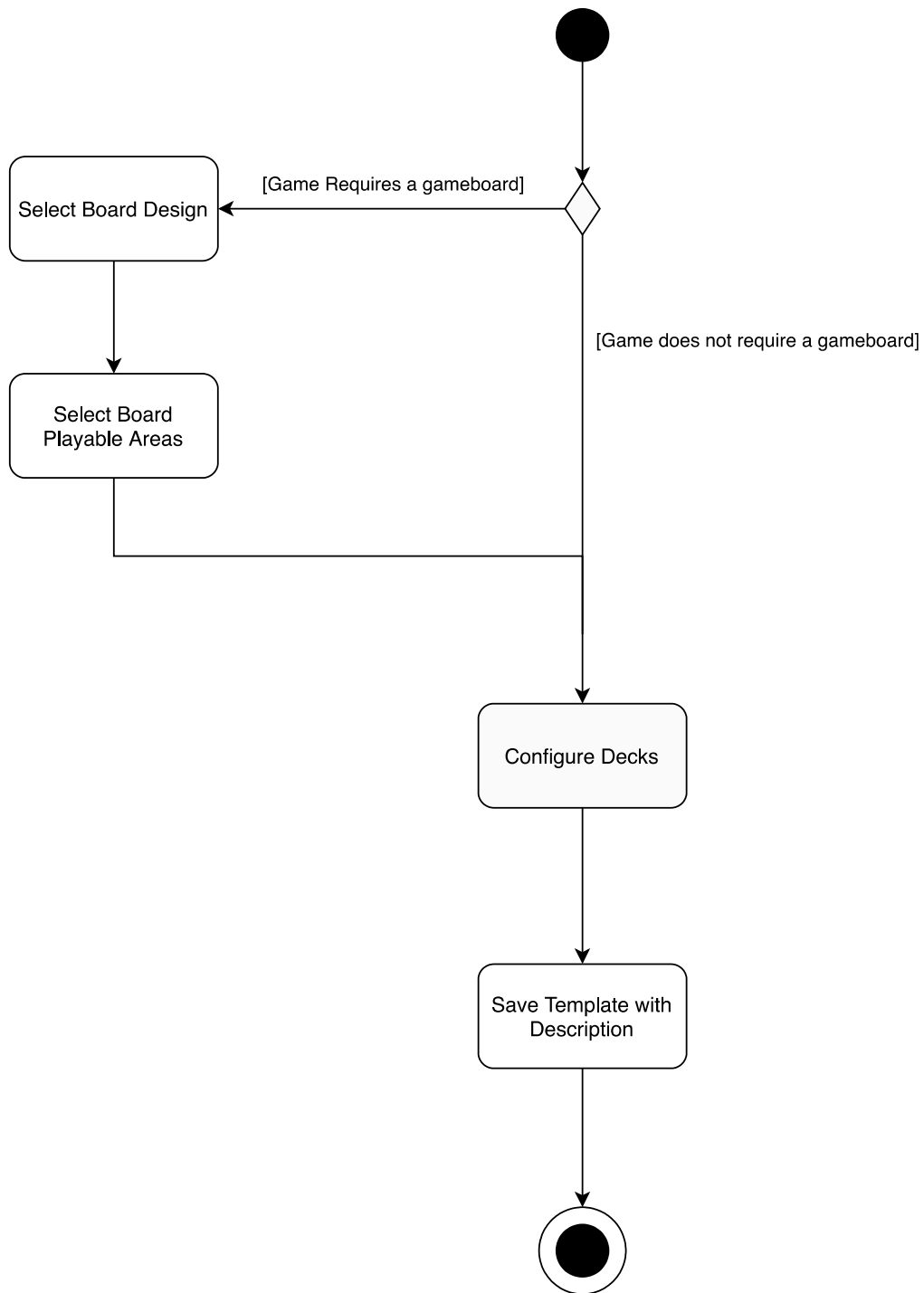


Figure 5: Game Design Activity Behavior UML Diagram Version 1

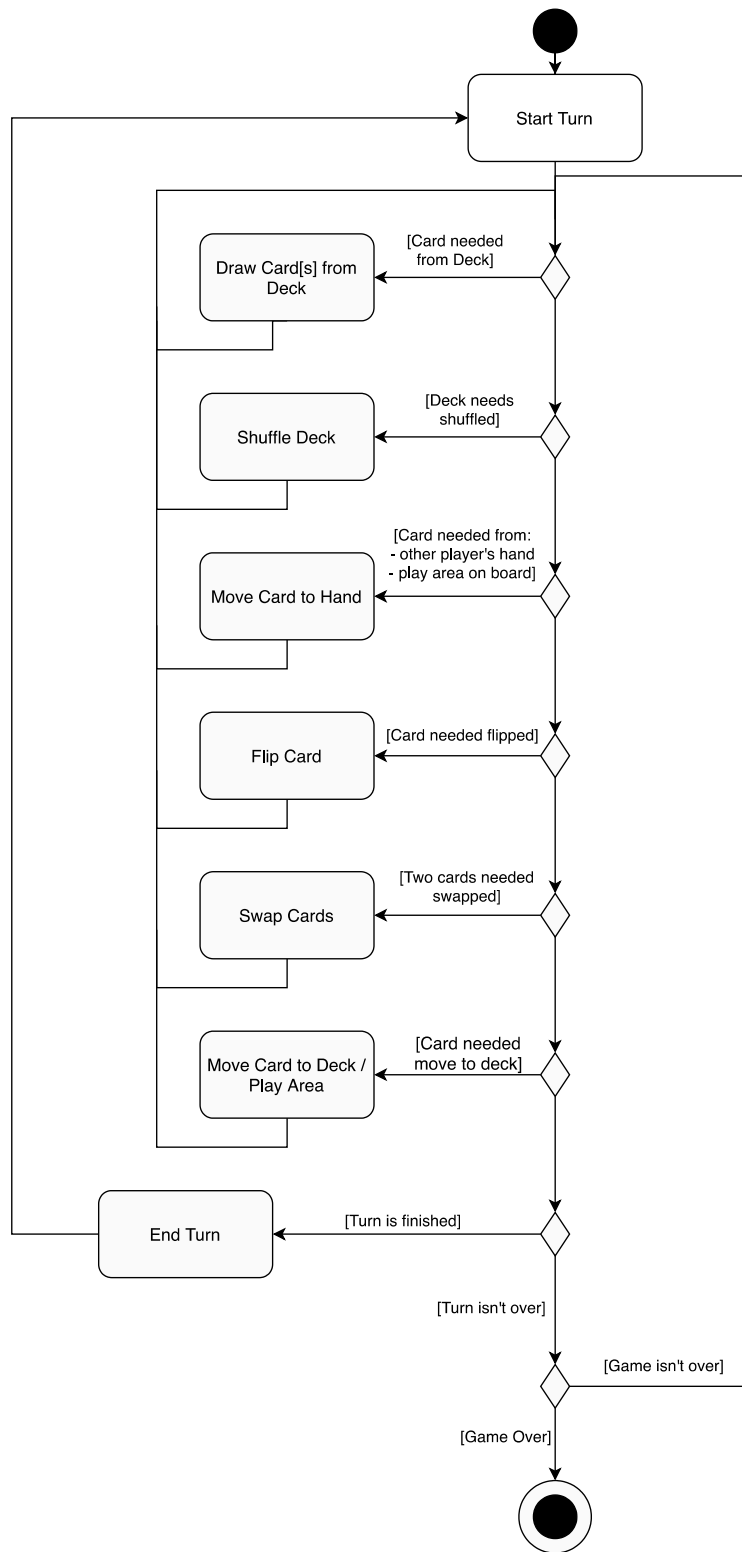


Figure 6: Gameplay Activity Behavior UML Diagram Version 1

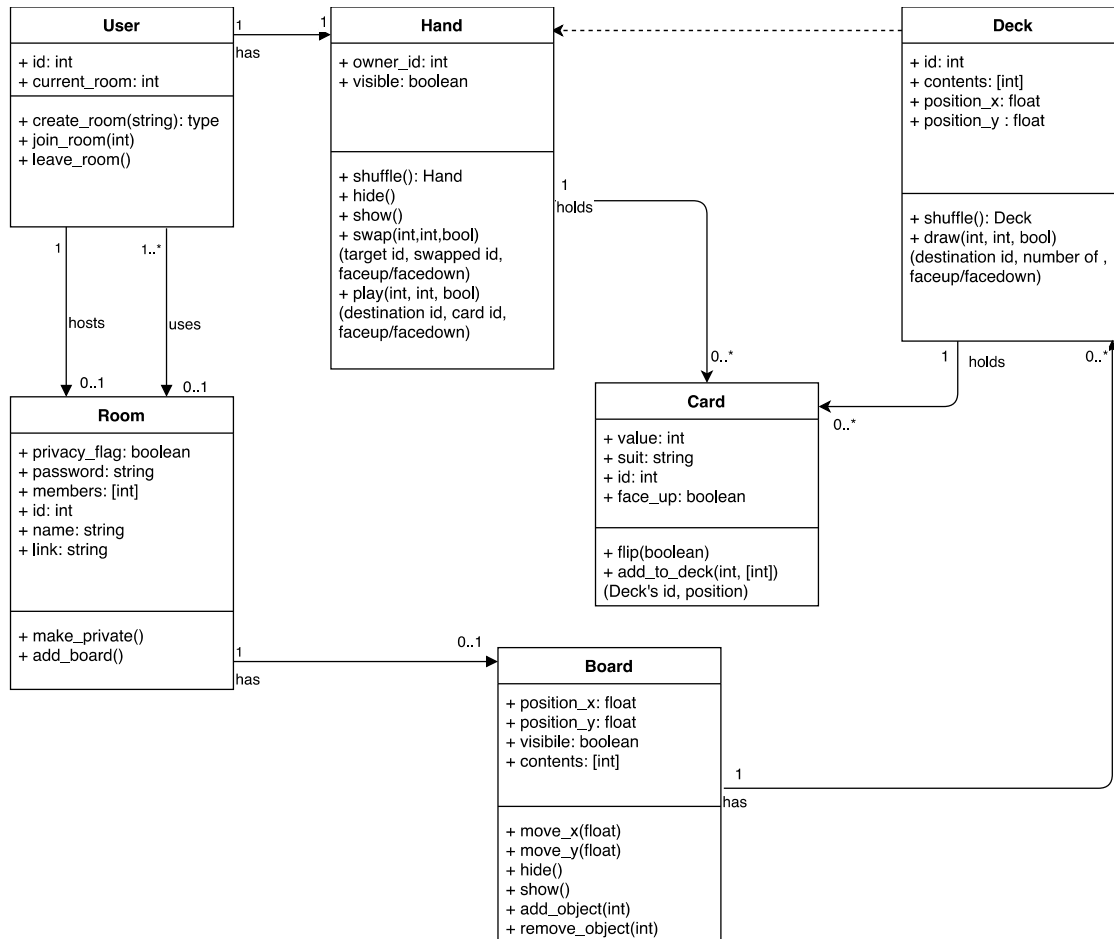


Figure 7: Card Game Class Structure UML Diagram Version 1

After the creation of figure 7 it became clear that the work spent on creating these diagrams was starting to deviate from HCI design and into implementation. To avoid the design of the application being negatively influenced by implementation, focus was moved to the creation of low fidelity paper/whiteboard sketches where individual team members would start with their own mental model by creating paper sketches, an example of which can be seen in figure 8. Collaboratively these sketches were reviewed and broken down into design choices which were joined together to create the initial design as seen in Appendix A.

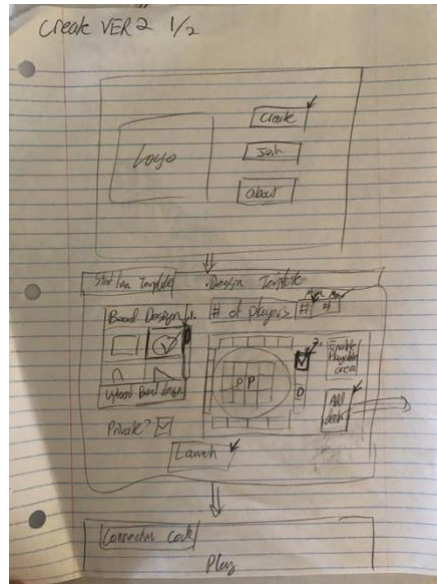


Figure 8: Example paper sketch of game creation

Further design considerations were made following the creation of use case storyboards for different card games which drove the design of new features to improve the user experience. An example of the storyboards created can be seen in figure 9.

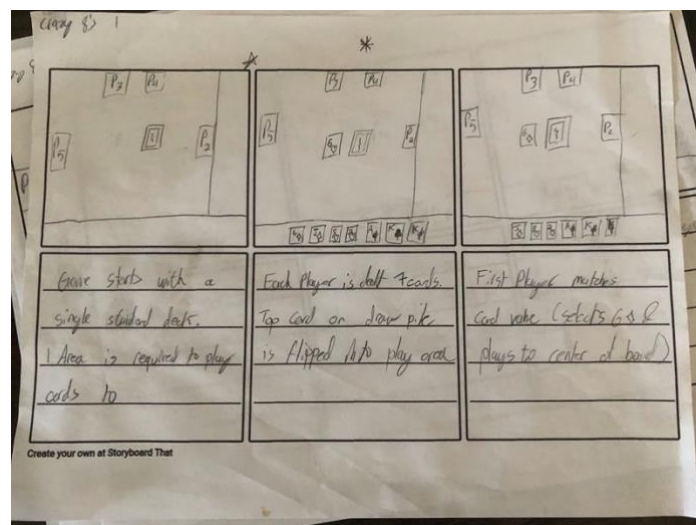


Figure 9: Example storyboard for Crazy 8's

In maintaining a novel approach to online virtual board games while also providing an improved user experience, the following key design decisions were made:

- Board manipulation through selecting options: Currently existing solutions primarily rely on “drag and drop” to manipulate their virtual objects. For a novel

- approach that would support streamlined and intuitive interactions the use of selecting from a list of options was chosen as the main method of board manipulation.
- Grid based gameplay: Objects on the gameboard are placed along a grid. This design decision was made with future improvements to the accessibility of the application in mind and to allow for a fully working minimal viable product prototype.
 - Context sensitive options based on selection: With the primary board manipulation using options it became apparent that displaying all possible options would lead to a cluttered and unintuitive UI. To avoid this problem and to fully benefit from using option selection instead of “drag and drop,” options are displayed based on the context of the selected objects. Only the available options based on the objects selected are provided to the user leading to a cleaner and more intuitive UI.
 - Automation of repetitive tasks through “Dealer” macro: Through the storyboarding process the number of repetitive tasks in many different card games was highlighted. In discussion with Dr. Miguel Nacenta the use of some tool to automate repetitive tasks was decided upon. The design of a dealer entity on the gameboard was made as an extension of the context sensitive option displays. When selected the dealer would provide access to automated actions to reduce the need for repetition.
 - Symbolic turns: Initially the design included allowing players only to interact with the game board on their turn. This decision was initially made due to synchronization concerns. This constraint of gameplay was analyzed against possible games and deemed an unnecessary complication. Games such as UNO include many transitions between players with players able to interrupt turns. The design was then modified to make turns a symbolic optional feature to be used only for the benefit of the user.
 - Hand reordering: It is known that many players like to reorder the cards in their hands while they play. As a continuation of simulating real life card games the ability to reorder the hands was included in the design.
 - Game template play area modifications: The design of game templates allows for selecting which areas of the game board are used for gameplay. This allows for templates to be made that are more intuitive to use as the playable areas are shown.
 - Supporting multiple decks and deck customization: With the goal of providing a flexible interface that supports as many card games as possible, the ability to use non-standard card decks was required for the design.
 - Including the setup of the “dealer” in the game template design: Highlighted by the storyboarding process was the setup required for many card games leading to a series of repetitive tasks. To further the benefits of automating repetitive tasks the use of the dealer was included in the design of the template creator to allow users to further automate their game setup.

VII Final Design Details

The prototype implemented was reduced in scope to ensure a working minimal viable product could be presented. Screenshots of the final implementation can be found in Appendix B. The activity diagrams for the currently existing implementation can be observed in figures 10 and 11. The automating dealer and customizable deck features were not implemented but are still included in the design. While some key features are still marked as future implementation work the majority of the objectives for the project were met:

- The application is accessible by anyone with an internet connected device. It is possible as the prototype was successfully built in a Docker Container and while it currently is not hosted online, it could be easily.
- The application does accommodate 1 to 8 players.
- The application is mostly flexible to allow people to play niche games and house rules of common games. Through unconstrained card moving, games that use a standard 52 deck of cards are playable. Once the custom card decks are implemented any niche card game would be possible.
- The application is intuitive, avoiding too large of a learning curve. Through the use of context sensitive options and grid-based templates manipulating the gameboard to play card games follows a logical flow.
- The application is not currently scalable. While the structure of the database that stores user and game data would scale, the prototype would require a refactor of the backend that handles maintaining a shared state between players before it could be considered scalable. The current implementation was done due to time constraints and the desire to have a fully working prototype to display the other features and objectives.

Due to the current front-end design there is a boundary condition on the number of cards supported in a user's hand before the options are no longer visible. The boundary condition is reliant on the size of the user's screen and browser window with larger windows supporting larger hands.

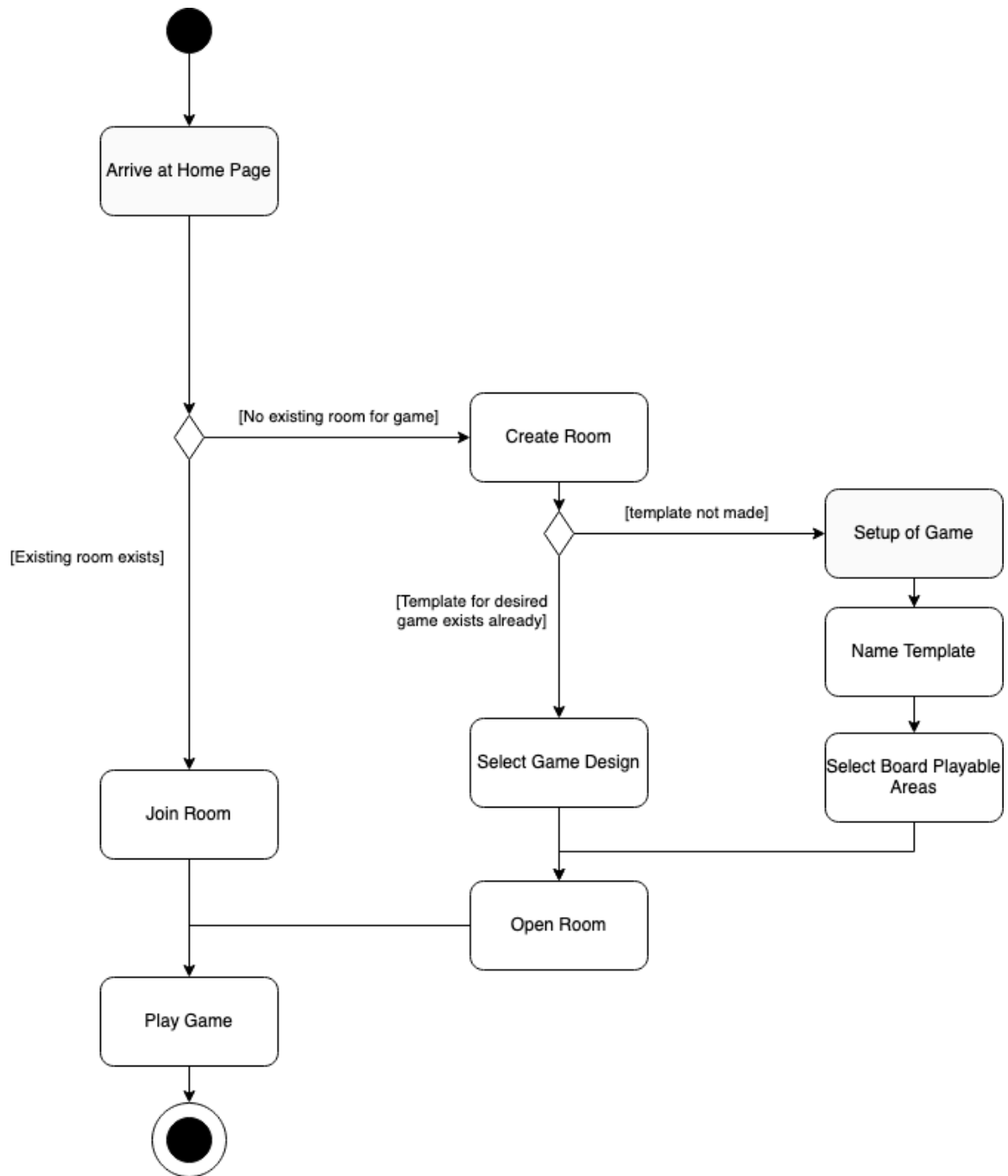


Figure 10: Overall Activity Behaviour UML Diagram Version 2

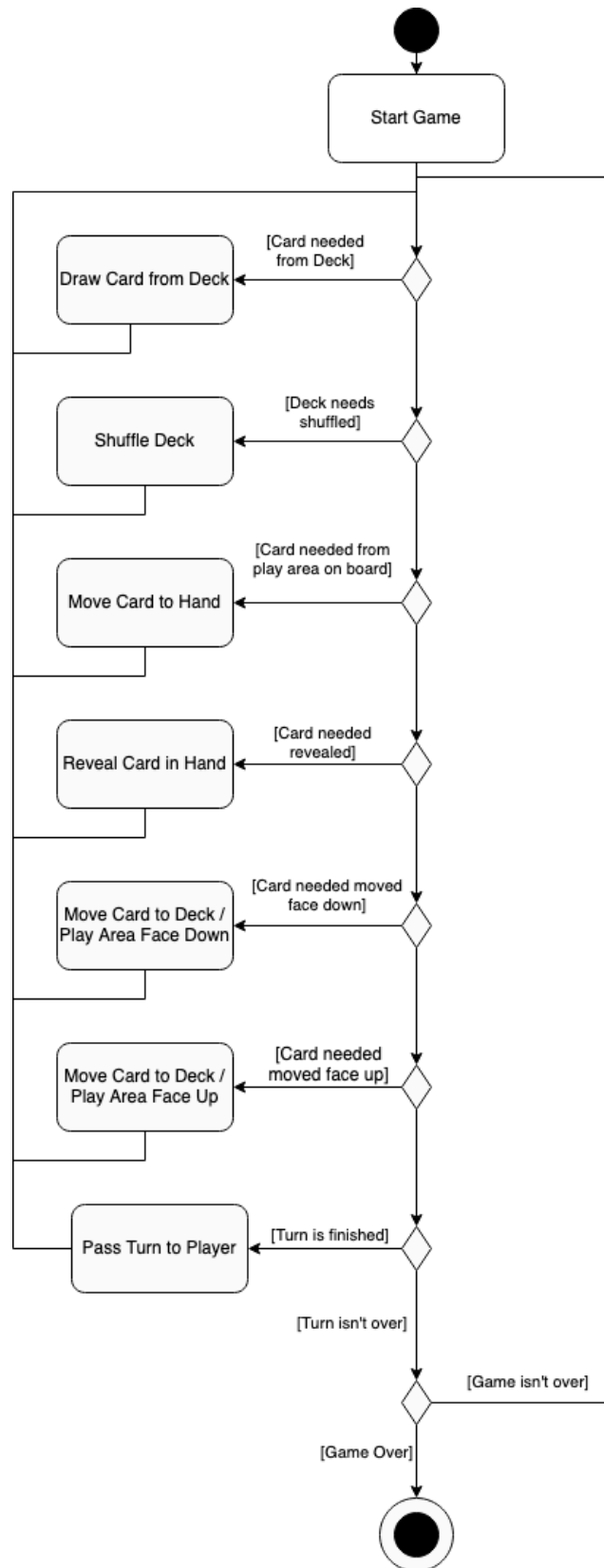


Figure 11: Gameplay Activity Behavior UML Diagram Version 2

VIII Testing & Validation

The testing of our platform is done in three stages: storyboards to test designs, usability tests to test implementation, and code tests to test basic functionality. The storyboards and code tests were primarily used to ensure that development was on the right track while usability testing was done to verify the results of our work.

VIII.I Storyboards

Before we began the implementation of our platform, we had to test out our mocks by creating storyboards. Since there are four of us, we picked 4 games to test our initial mock designs (found in appendix A). The four games were Texas hold ‘em, crazy eights, Durak, and Uno. Each of us went through a single game and checked that the games could be built with the current UI designs.

During this process we found several issues in the mock designs. We were missing a way to quickly set up a game and deal out cards, so we added dealer capability. We also needed tokens for games like Texas hold ‘em and deck design page for games with custom decks such as bridge.

VIII.II Usability Testing

Usability testing was done primarily by executing our story boards on the platform after implementation was complete. Notable changes we scheduled after usability testing were to add additional functionality for the dealer to do more than set up the game but being able to execute macros for a game like Texas hold ‘em that needs the dealer to deal out the flop, river and turn cards onto the board. We also found that players wanted to be able to manipulate cards like they do with physical cards such as reordering their hand or placing cards onto the board splayed out. Ideally, we would implement more changes based on this stage of the design and revisit usability testing as we increase the scope of the product. However due to the time constraints on the project we were only able to go through this phase once near the end of the term, this is mostly because we took a long time on the design phase to ensure that we did not dive into implementation too quickly without a good design. Our end product reflects this by giving a polished experience with limited functionality rather than overloading the user with options that they may not need.

VIII.III Code tests

Code testing was done via circle CI, a continuous integration tool. This tool tested the database functionality and the Django platform, as well as making sure that our project structure remains consistent. No major changes were caused by this however it was helpful to immediately see issues in commits made, since this runs automatically when a commit to the version control was made.

VIII.IV Test Plan

The current testing done is sufficient for our platform as it is not a safety critical application and failures of systems like these are good for improving the design iteratively. Before a release is done, we would like to implement our changes from the first testing cycle and also add a feature to report bugs and suggest additions to make the testing user based. additional code testing with circle CI would also be good to set up but because our

development was fairly fast paced, we have limited testing done every commit. This testing could be expanded to test each page and each database function with test data.

IX Discussion & Recommendations

The storyboarding and usability testing conducted on the prototype were successful in generating a list of possible improvements to the prototype to fully meet the objectives of the project. Included as future work to our final design is:

- Make dealing more easier by not resetting selection after dealing or providing a dealer. Objects are currently deselected after an option is used but in certain cases like drawing from a deck it would provide a better user experience if the selection was maintained and allowed for the draw/deal option to be clicked multiple times in a row. Alternatively, the proposed dealer function would also handle this and other UI annoyances with repetitive tasks.
- Growing spaces dynamically for cards to be placed on sides. After demonstrating our prototype to Dr. Miguel Nacenta, we were informed of the possibility of certain card games requiring dynamically increasing board space as some games do not use specified play areas but instead change depending on gameplay.
- Create deck designer pages. Including customizable decks was in the design but not included in the prototype. Through usability testing it is clear that high on the priority list for future work is adding an interface to customize decks.
- Add tokens. The standard gambling card games require not just cards but other entities to represent score. While these could be kept track of by players off of the application it became clear through demonstrating the prototype that it would benefit the immersion of the simulation and improve user experience to include the ability to have some representation of tokens and score.
- Add UI fixes and quality of life changes for ease of use. Certain use cases such as resetting the game board are not handled well by the design. Further usability testing would allow for these problems to be listed out and fixed with minor redesigns.
- Add new boards to use. As the title of this project is virtual board game engine and not virtual card game engine it would stand to reason that the platform could be expanded to include support for other board games such as - chess, backgammon and monopoly.
- Hand reordering. This allows the user to adapt their hand to their preference as done in an in-person game. Dr. Miguel Nacenta made it clear that this is a must have feature because it allows users to play the way they like, and hand reordering can be helpful in particular games.
- Discovered in researching currently existing solutions was the programming engine boardgames.io. A possible variation on this project could include using that engine as a framework and include extensions to that open source project instead.

X Conclusion

In this project we set out to provide an accessible platform for family and friends to stay connected while also providing useful tools in playing a variety of unique house rules. We examined 4 possible approaches and set on using a combination of all of them for our initial prototype and included 4 main features: grid-based placement, logs, macros, and designing custom games. Through an iterative design process, we created design objectives and improved the user experience and intuitiveness of card management and achieved a fully implemented prototype version of our engine that can be played locally. The limitations of the current prototype are that assets cannot be modified, games with tokens are not currently playable, and that many actions are repetitive and can be improved through macros. In the future we hope to improve on the usability, and customizability of our solution and then consider the possibility of scaling this product into a more global solution through open source contributions.

References

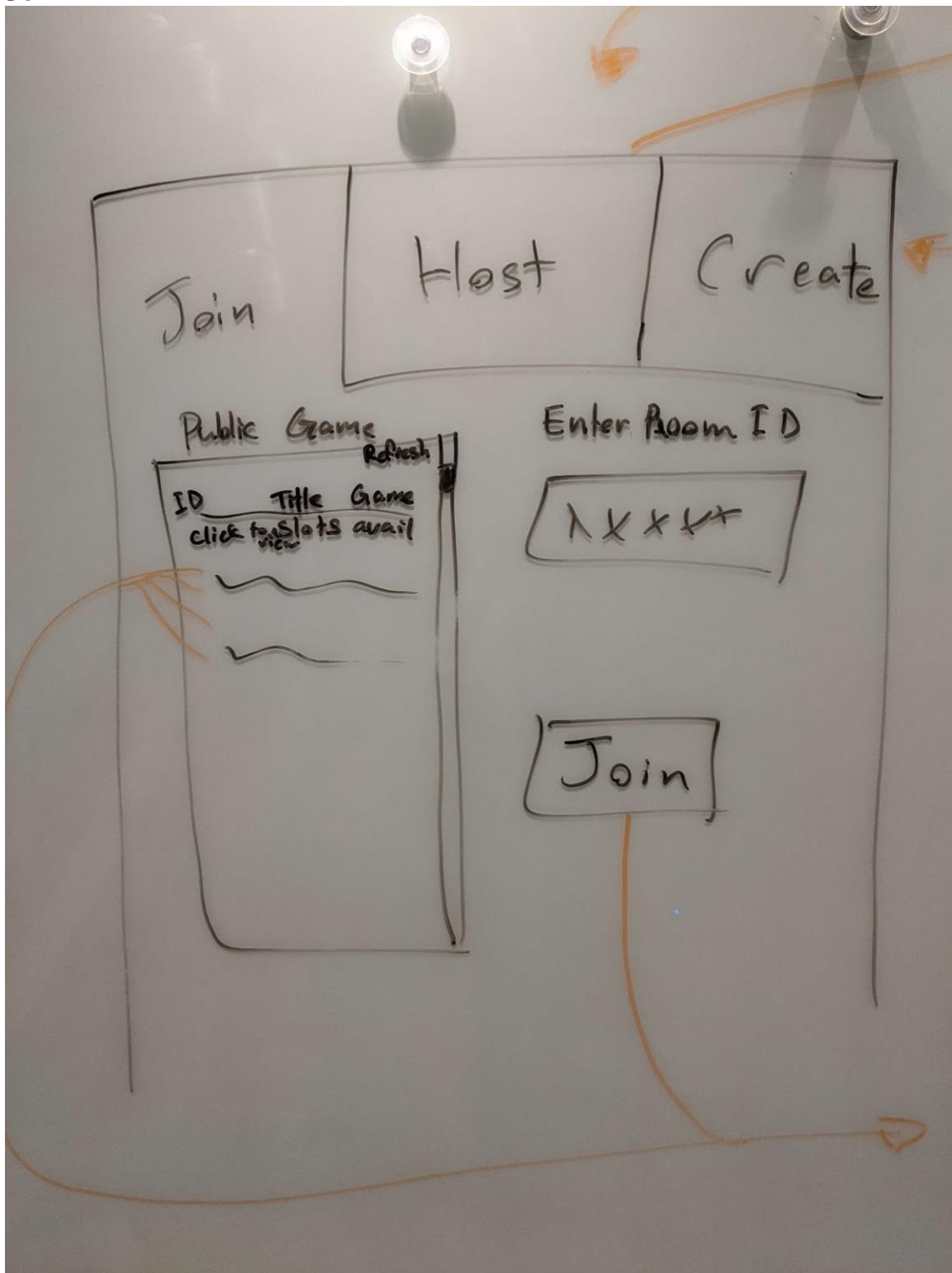
- [1] "cardgames.io", *Cardgames.io*, 2020. [Online]. Available: <https://cardgames.io/chess/>. [Accessed: 28- Jul- 2020].
- [2] B. Games, "About | Tabletop Simulator", *Tabletopsimulator.com*, 2019. [Online]. Available: <https://www.tabletopsimulator.com/about>. [Accessed: 29- Jul- 2020].
- [3] "Advanced controls - Tabletop Simulator Knowledge Base", *Kb.tabletopsimulator.com*, 2020. [Online]. Available: <https://kb.tabletopsimulator.com/player-guides/advanced-controls/>. [Accessed: 28- Jul- 2020].

Appendix A: Design Sketches

Start



Join



Host

Join	Host	Create
ROOM NAME		
list of template		* If you don't see one you want go here and create a new one
<input type="text"/>		
Preview		
<div>Description image of board</div>		
Num players		
Private/Public		
<input type="button" value="OPEN ROOM"/>		
<div>Deck Creation Current <input checked="" type="checkbox"/></div>		

Join Host Create

game board

+	+	+	+	+
+	+	+	+	+
+	<div>0 c5b5</div>	+	<div>5c c5b5 edit</div>	+
+	+	+	+	+

help box

edit and move btn

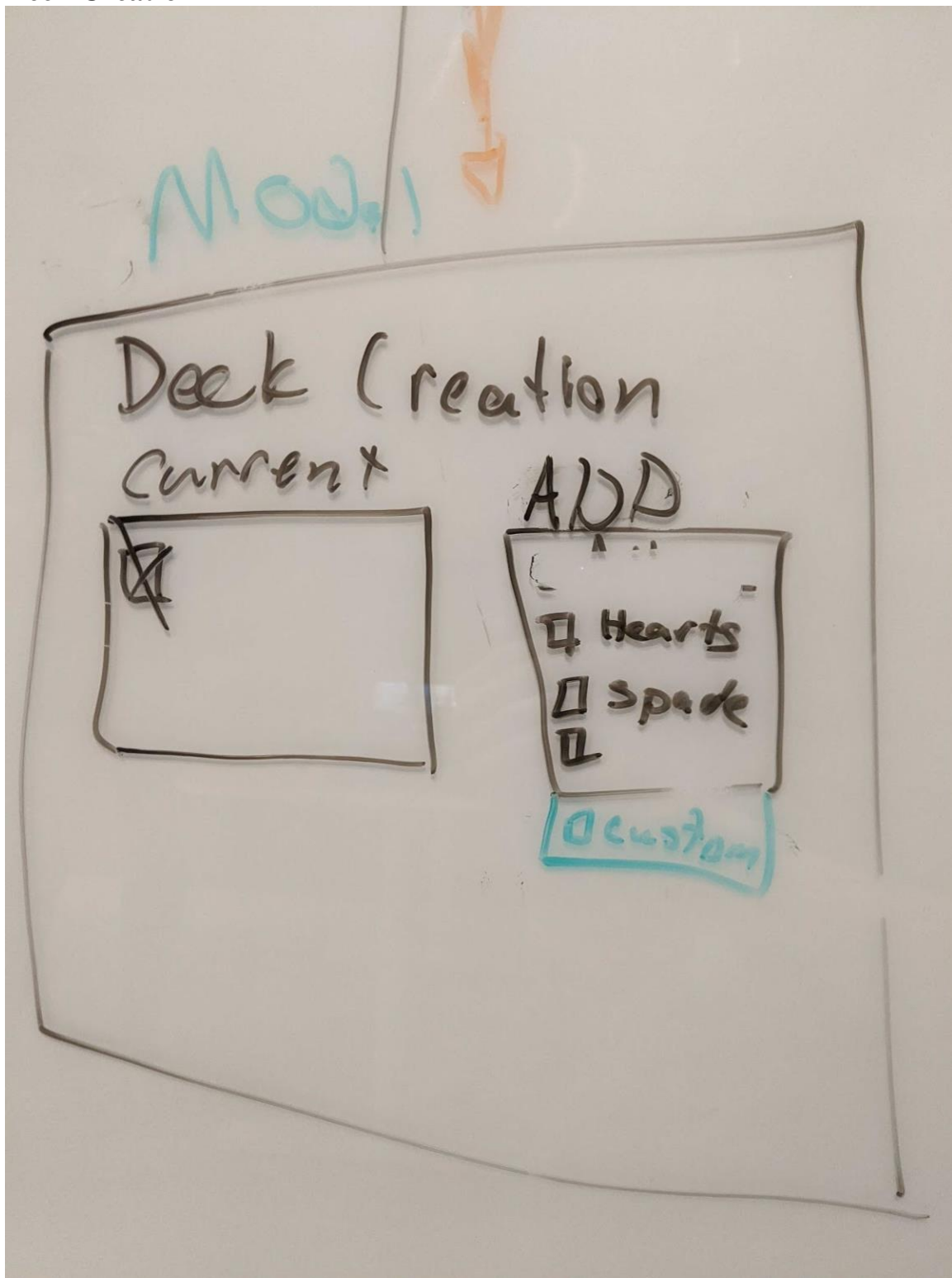
Click to add empty play a

Players range
turn direction cw/ccw

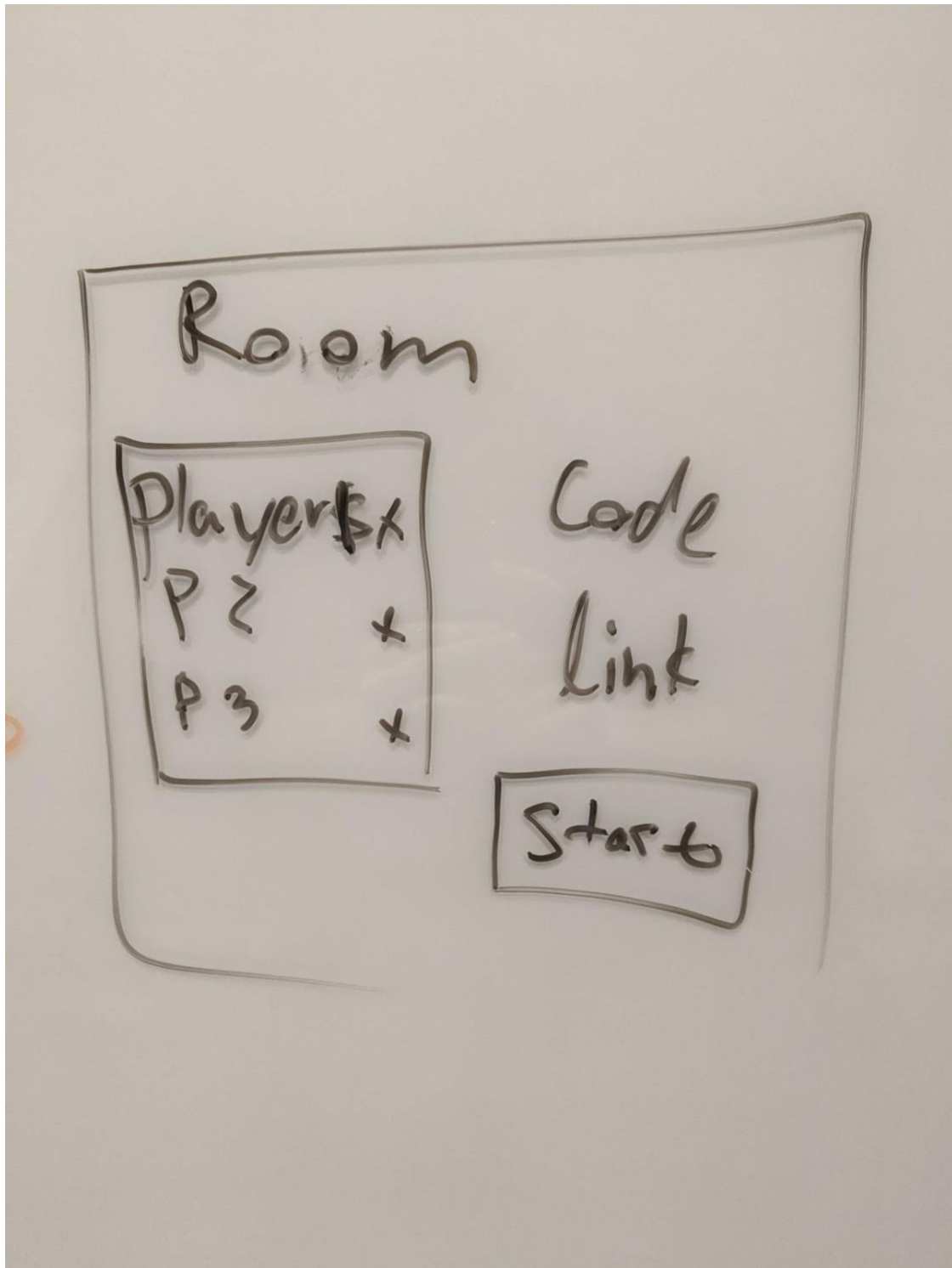
Game Description

save

Deck Creation



Room



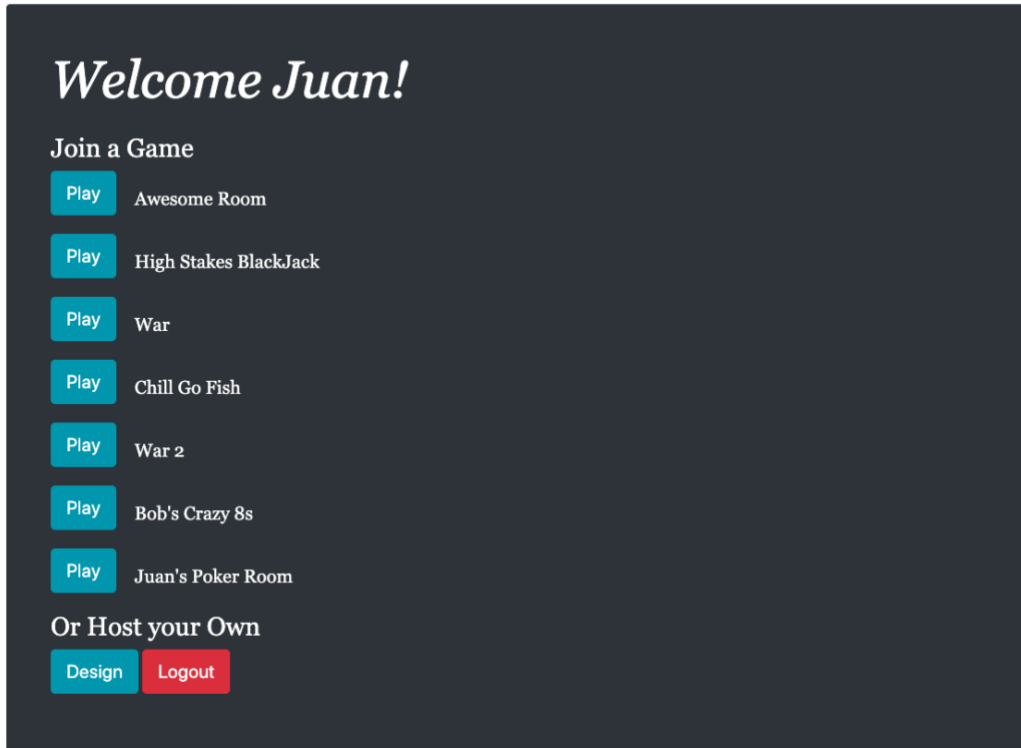
Play



Appendix B: App Screenshots

Home Page

Emoj



Host a game!

Design your game play below.

Room Name

Select a
template

Or create your
own



Template Name

Design Board

Select positions
to add to the
board.



+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

Open Room

Play Page

Home

Play

Juan

Details

Oleg

Details

Emily

Details

Options

Game Log

Hand

Appendix C: Full Overview UI Sketch

