

Part 1: Project Description

For our final project, we decided on the board game Onitama. Onitama is a two player strategy game that uses many ideas from chess to form a completely new and unique experience. In Onitama, each player has 5 pieces total. There are 4 student pawns and 1 master pawn per player on a 5x5 square board. What is unique about this game is how your pieces move. Your movement is decided by 5 random cards drawn at the start of the game. Each card has a different moveset indicated on it which you can apply to any of your pieces. Each player always has 2 cards at a time, which rotate after you use one and switch with the other player. This adds a really unique strategy to the game where you know exactly what moves will be available to your opponent on your turn, and you can even prevent them from getting one move by holding it on your side.

There are two unique ways to win this simple but challenging game, The way of the Stone, and The way of the Stream. The first being very similar to chess, take their master pawn. You can take it with any piece, and there is no check, only capture. The second way however is a bit more unique. On the starting position of each player's master pawn, there is a special space. If you manage to get your master pawn onto the opponent's master space, you win the game. With two unique win conditions, and a slightly different experience every round with the different card moves, this game creates a quick but thoughtful game experience that we are all very excited to bring to life with our project.

Our main plan right now is a bit odd. Instead of creating individual pieces in our game, we plan on having all the movement and captures to happen inside of the board itself. The thought process behind this is to think of the game as less a board with pieces on it, but a 5x5 board made of 25 different squares with designs on them. Instead of moving a piece to a new square, we are simply switching the squares unless a capture happens, in which case you get rid of a square and replace it with an empty tile. This method of thinking is a bit weird in terms of the game, but should provide a really solid base for our game that will allow easy customization for future endeavors. Some fun things we have planned include pawn skins that you can select at the start of the game, and for-fun modes such as a 7x7 board or two master pawn mode. We cannot wait to bring this game to the digital world for this final project, and we hope you enjoy the experience as much as we do!

Part 2: Functional requirements

GUI	Board
Priority	High

Purpose	The Board GUI shall display the board for the user and return information via the buttons to the game object.
Inputs/ Needs	Need to understand Swing better.
Operators/ Actors	Board object, player1 and player 2
Outputs	Sets the values in the Board object.

Model	Board
Priority	High
Purpose	The Board shall compute valid moves and the location of pieces.
Inputs/ Needs	Need to commit more time/ finish the logic
Operators/ Actors	Board GUI object, player1 and player 2
Outputs	Contains the values used by the board GUI

Model	Game
Priority	High

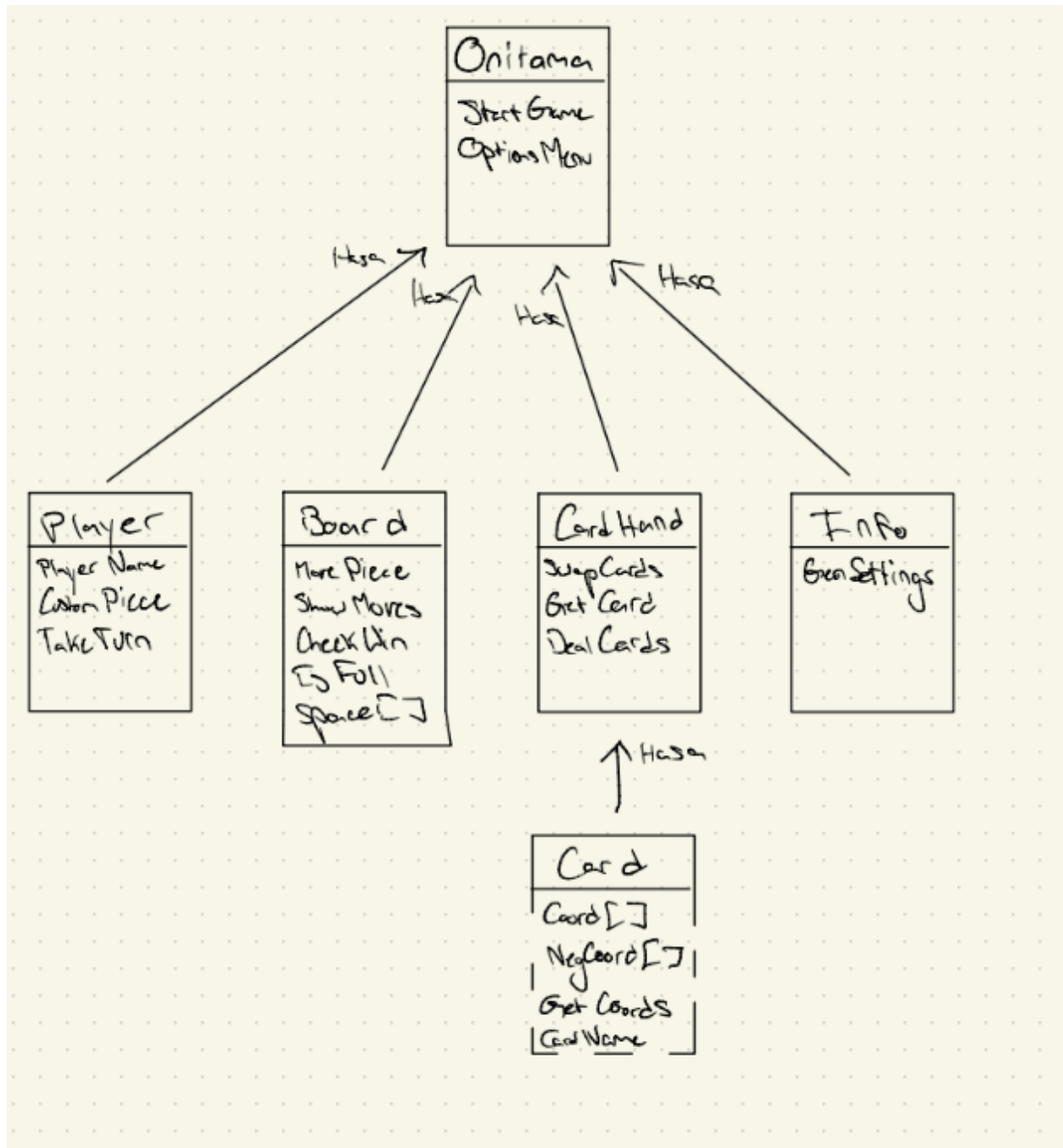
Purpose	The game shall run the game and manage turns.
Inputs/ Needs	Need to commit more time
Operators/ Actors	Interacts with all objects
Outputs	Holds lots of objects

GUI	Splash screen
Priority	medium
Purpose	The splash screen shall be able to change game settings and start a game.
inputs/needs	Need to assign someone to this job.
Operators/ Actors	This is the first interaction the player has with the game so it is important that the layout is clean and simple.
Outputs	Sets certain values in the info object such as player names, game modes, and player icons

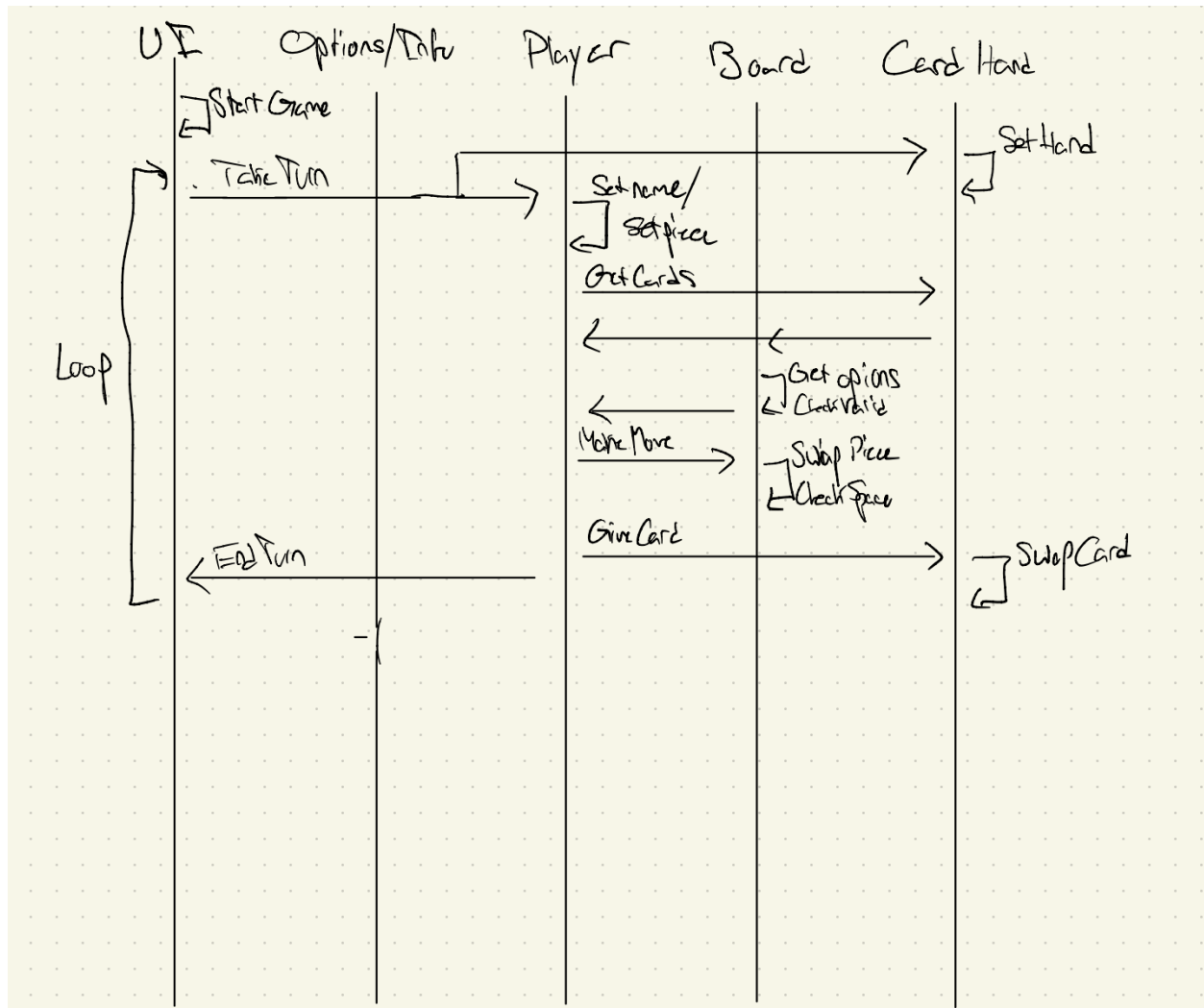
GUI	End screen
Priority	medium/low
Purpose	The end screen shall communicate the winner of the game and ask if the users would like to play again.

inputs/needs	Need to assign someone to this job.
Operators/ Actors	This is the final screen the players see so the priority is not as high as the other requirements.
Outputs	Does not have many outputs but it can start a new game.

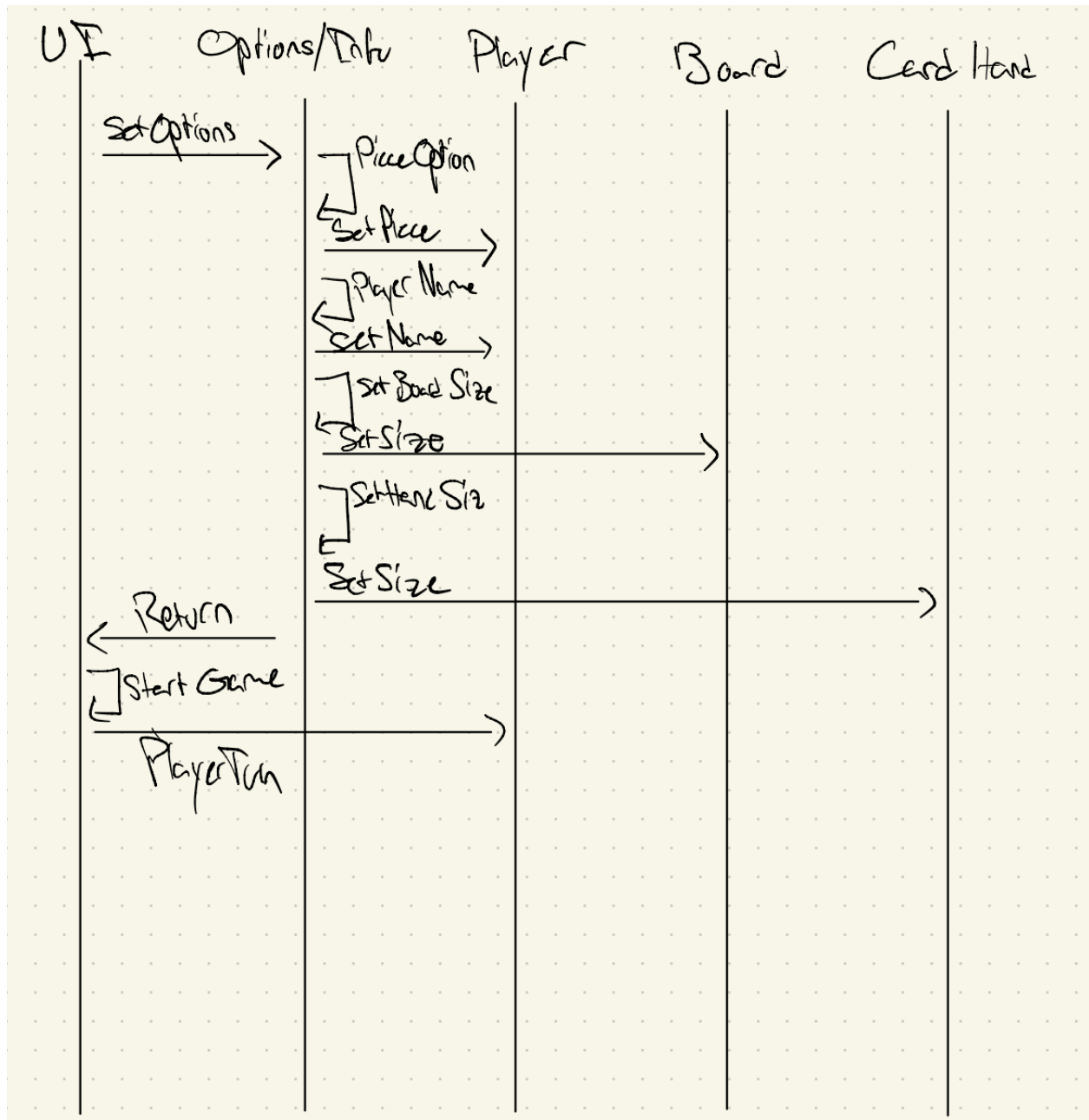
Part 3a: UML Diagram



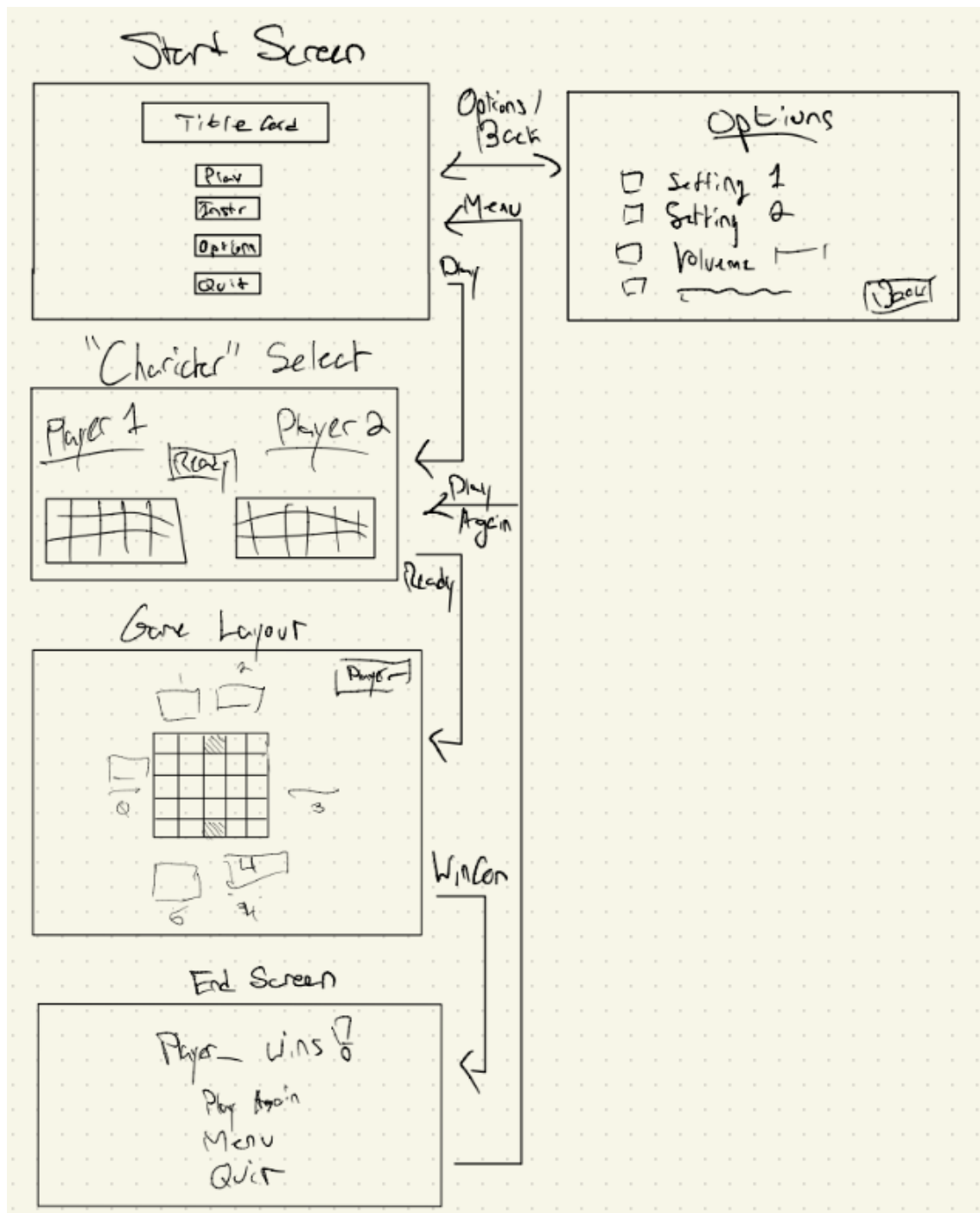
Part 3b: Use Case Diagrams



4/5/2023



Part 4: Storyboard



The implementation of Onitama is largely divided into three categories which may be divided into further subcategories: **GUI, Game Implementation, Testing**

The following represents a tentative plan that may be subject to change.

GUI:

Design: The first step in implementing the game GUI is translating the UI sketches into a non-functional prototype in swing. The game consists of five screens: Start, options, character selection, main gameplay (view of board), and end screen. Each of these screens must be made into an acceptable design prior to complete integration with the game.

The non-functional swing design will ideally be complete by April 18th.

Colin is currently beginning work on the prototype GUI build. The estimated total time to complete is estimated ranges from 4-8 hours or more as necessary.

The next major part of the GUI implementation involves integration with the code. Prior to this point the design will be complete including all components that the player will interact with. The functional game will be complete as well.

The due date for GUI code integration is estimated April 22nd.

All team members will contribute to this portion of the project as necessary.

Game Implementation

The process of game implementation began with the decision to decompose the game into several classes (Onitama, Player, Board, CardHand, Info, Card). The group has begun to write code for the game.

The non-GUI implementation of the game will ideally be complete by April 15th.

Caleb and Alex are currently working on implementing the game in code. The estimated total time to complete is estimated 5-12 hours or more as necessary.

As indicated above in the GUI section, following implementation of the non-GUI based game the next step in integration with the GUI.

The due date for GUI code integration is estimated April 22nd.

All team members will contribute to this portion of the project as necessary.

Testing

A significant portion of time will be devoted to testing the game to ensure it works as expected. At the lowest level individual behaviors within the various classes will be conducted as the code is being written. Every class will undergo thorough testing to ensure each of its behaviors works as expected. The entire system will be tested by each member along with a collection of users.

All team members will contribute to this portion of the project.

Individual class unit testing will ideally be complete April 25th.

System testing ideally be complete May 5th.

The week before the project due date has been reserved for fine tuning any aspects of the game as necessary.