Sprint 3 Rationale

FIT 3077 Group 29
Tycl's inc. ©
Terry Truong 32466870
Cynthian Thai 31477119
Leon Ma 31449840
Yuhan Zhou 31468748

Revised Architecture

A key change that was implemented within our architecture was our revised implementation of our board setup. Revising our board design was deemed essential by our team after finishing our second sprint. The problems that arose in our design during our second sprint realised the inability to design neighbouring positions to tokens placed on the board without heavy violation of the single responsibility principle. This issue was given a heavy priority to fix in this sprint where we recreated the board's design to allow for specific token placed on the board to be given an attribute (int x, int y). The board class now creates a board of fixed size that contains coordinates that tokens can be placed on. This solves our neighbour problem as in our previous sprint, positions were stored in an array and each position's neighbours were hard coded in as the horizontal and vertical neighbours. In our new design the coordinates can either contain or not contain a token and the board will be updated when a token is placed within the specific coordinate.

Another significant modification made since sprint 2 was the use of the Model View Controller design pattern. Previously, all display logic was included in all classes; for example, the Board class will have related construct display methods in order to generate the board. The UI of the class required to be updated considerably more frequently than the underlying model while building the game rules, due to UI components also being tied to implementation logic, this breaks the design Principle "Separation of concerns" and thus the design was revised. The Model View Controller approach was used instead to segregate the user interface from the implementation code. GameView and PlayerView are new classes that were added during this sprint to reflect the concept of a view that manages display information. The Game classes now operate as controllers, interpreting user inputs such as clicking on a token or positions and informing the model about the move made and finally,, the controller will also inform the view to update accordingly. This design change also made isolating sources of error and bug detection much easier because it now adheres to the separation of concerns, where both the view and the controller rely on the model but the model does not. This solves the original problem because the implementation is now separated and the model can be built independently and tested.

Overall, this change we had made to the board class has proven to be much more efficient and viable moving forwards with our game design.

Quality attributes

Usability

9 Men Morris being a software that is going to be utilised by user's that may not be familiar with the game, its mechanics and rules, but most importantly it is also a software designed to provide an enjoyable experience for its users, so creating an easy to use and understandable interface will be critical to their user experience. The game's design manifests this non-functional requirement as it adheres to Norman's usability principles and heuristics.

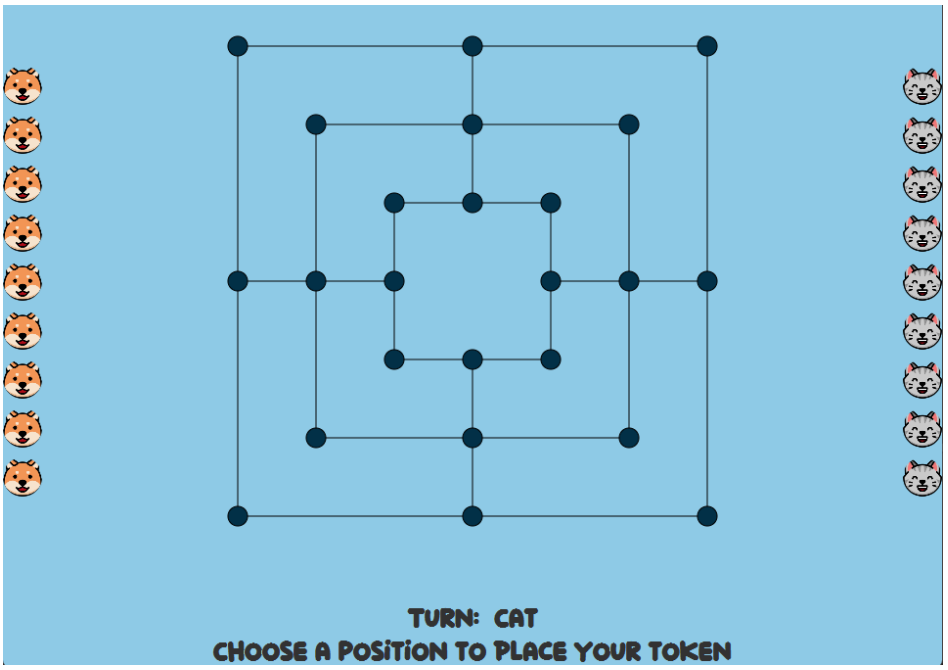


Figure 1: Start Game Initial Board View

The "Recognition rather than recall" heuristic of the 10 usability heuristics is adhered to by keeping track of whose turn it is and specifying the action they need to take to make their move at the bottom of the view (Figure 1). This intuitive design allows the user to have a more pleasurable experience while making the game smoother. The display shows how many tokens are yet to be placed on the side. This eliminates the need for the user to constantly remember how many tokens they have left and whose turn it is, drastically reducing the users need to recall or count while playing the game.

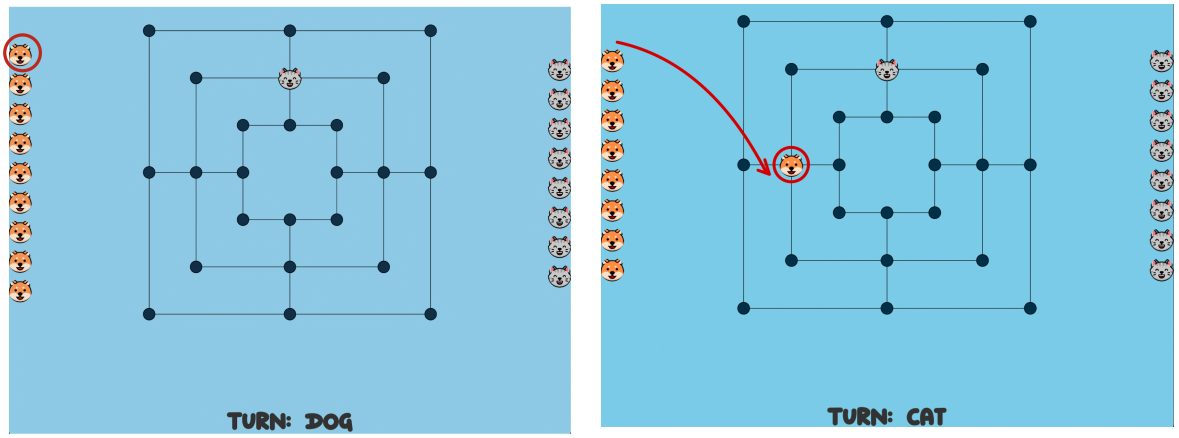


Figure 2: Example of player placing a token

As shown in Figure 2, by displaying the user's token on the board when a position is clicked, there is an indication that their action was successfully performed. This input also acts as an indication that the game is functioning as planned, and through predictable interactions creates a sense of trust in the software, resulting in a soothing experience.

Reliability

The reliability of a software system affects how well it can regularly execute a given function. Nine Men's Morris is a piece of software designed to provide its users with amusement. The game also adds competitiveness by evaluating how well players perform against a rigid set of rules defined by the software against other players. As a result, reliability is a crucial non-functional criterion since it reveals how consistently the software follows the rules it defines. By ensuring that the game follows the rules, it does not give any of the players an unfair advantage, which is likely to reduce the enjoyment of the game, but it also ensures that the software is always set up correctly, which may reduce frustration in its users and thus helps keep the player engaged and provide an enjoyable experience.

To guarantee that this non-functional need is met, the Tycl development team conducted extensive testing, including testing out edge scenarios, mincing illegal moves, as well as starting and restarting the game numerous times to ensure the game board is set up correctly. PlayTesting was performed to confirm that the software behaves correctly according to the given rules. A set of 5 testers were instructed to run and play the software 10 times and note any irregularities that were detected. Bugs and errors were detected and addressed during playtesting to assure the game's reliability.

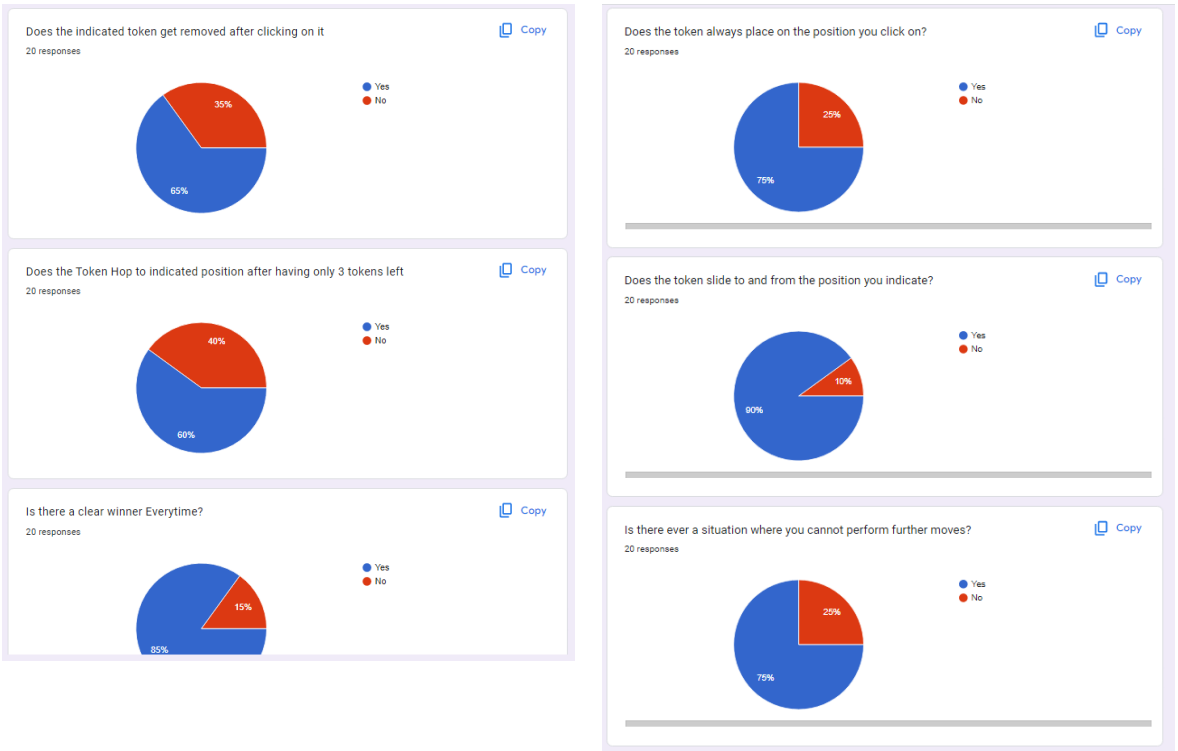


Figure 3: Results of testing the software while the game was in development

Human value (Schwartz's theory)

Pleasure was a main value from Schwartz's theory that we endorsed through the implementation of the game as a game that couldn't be enjoyed wouldn't be a very good game at all. We did this by bringing lighthearted UI design into the game so that it would change a more bland UI into something that would encourage lighthearted gameplay. This is done with our font size and also our token characters. *Dilo world* font (Figure 4) gives its characters rounded edges which generates a more friendly and delightful environment, inviting its players to continue playing.

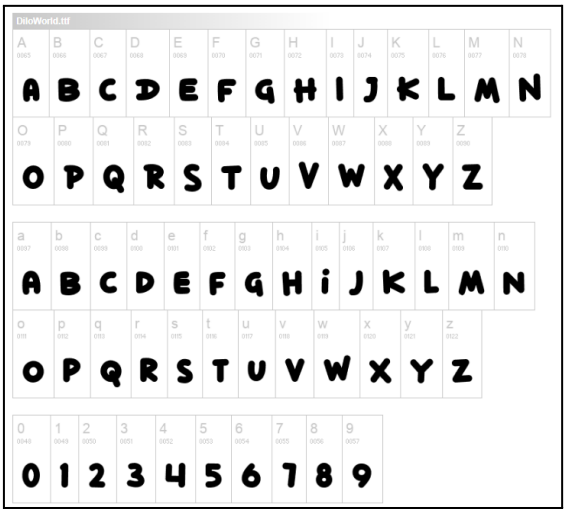


Figure 4: Dilo world Font Alphabet and Numbers

Figure 5: Example of Dilo world Font in game

Token pieces are represented by cute animals of cats and dogs (Figure 6), which reinforces this value. By using cute animal tokens and light pastel colours, it creates a more lighthearted and colour mood in comparison to the original black and white colour scheme, and thus creates a sense of happy emotions, allowing its players to have a more enjoyable experience.

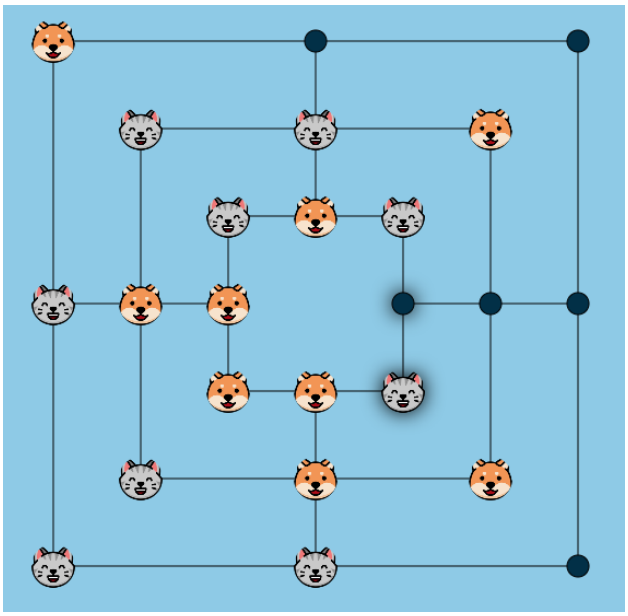


Figure 6: Tokens on the Game Board

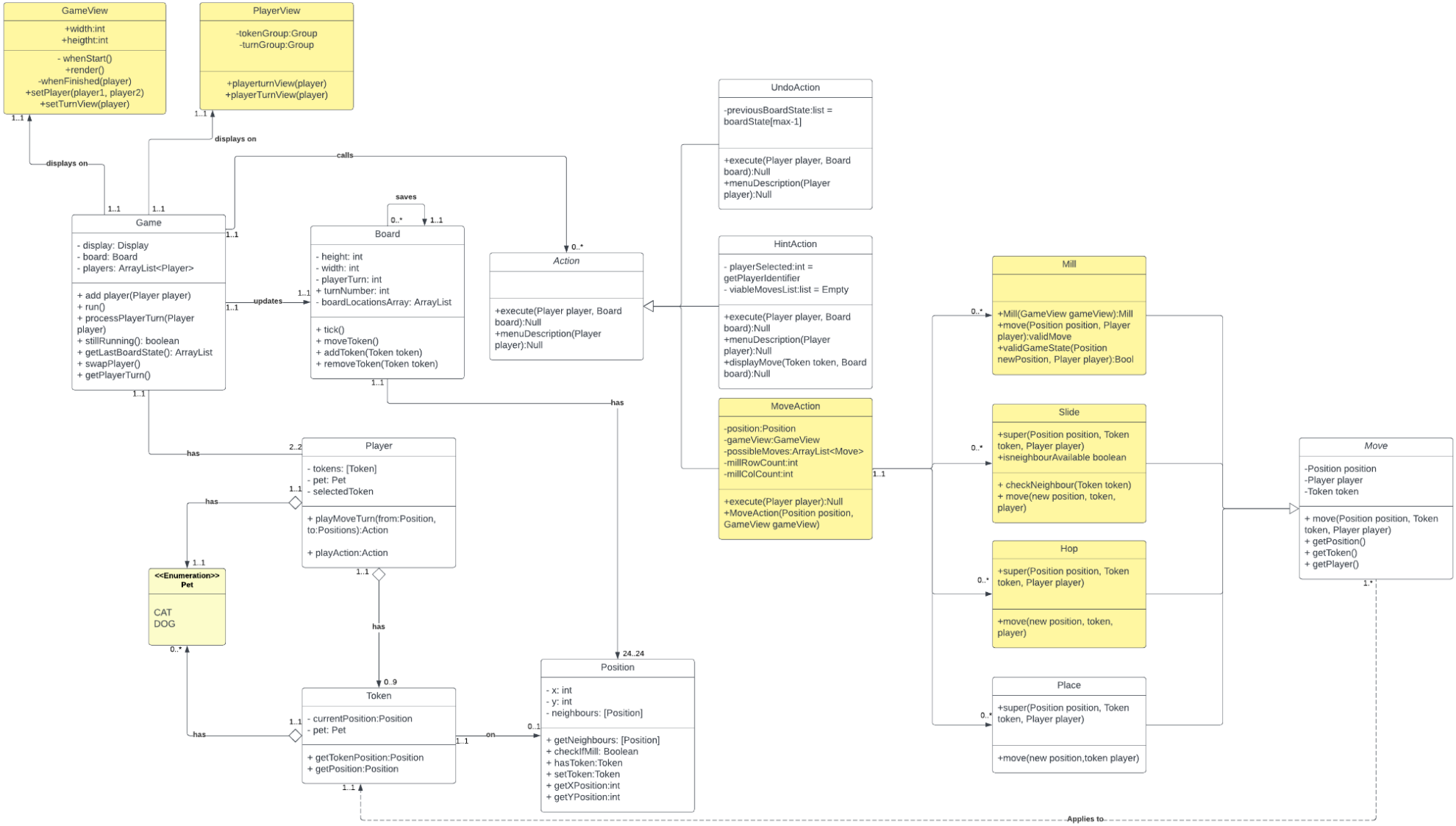
The helpful value from Schwartz's theory was a main value that we endorsed throughout our coding functionality. This is represented through the special requirements implemented 'hints'. This functionality was designed to assist in helping newer players or players who are not as familiar with the game rules be able to play the game without running too many invalid moves. This is important as it allows all players of different experience levels to enjoy and play the game following its rules.

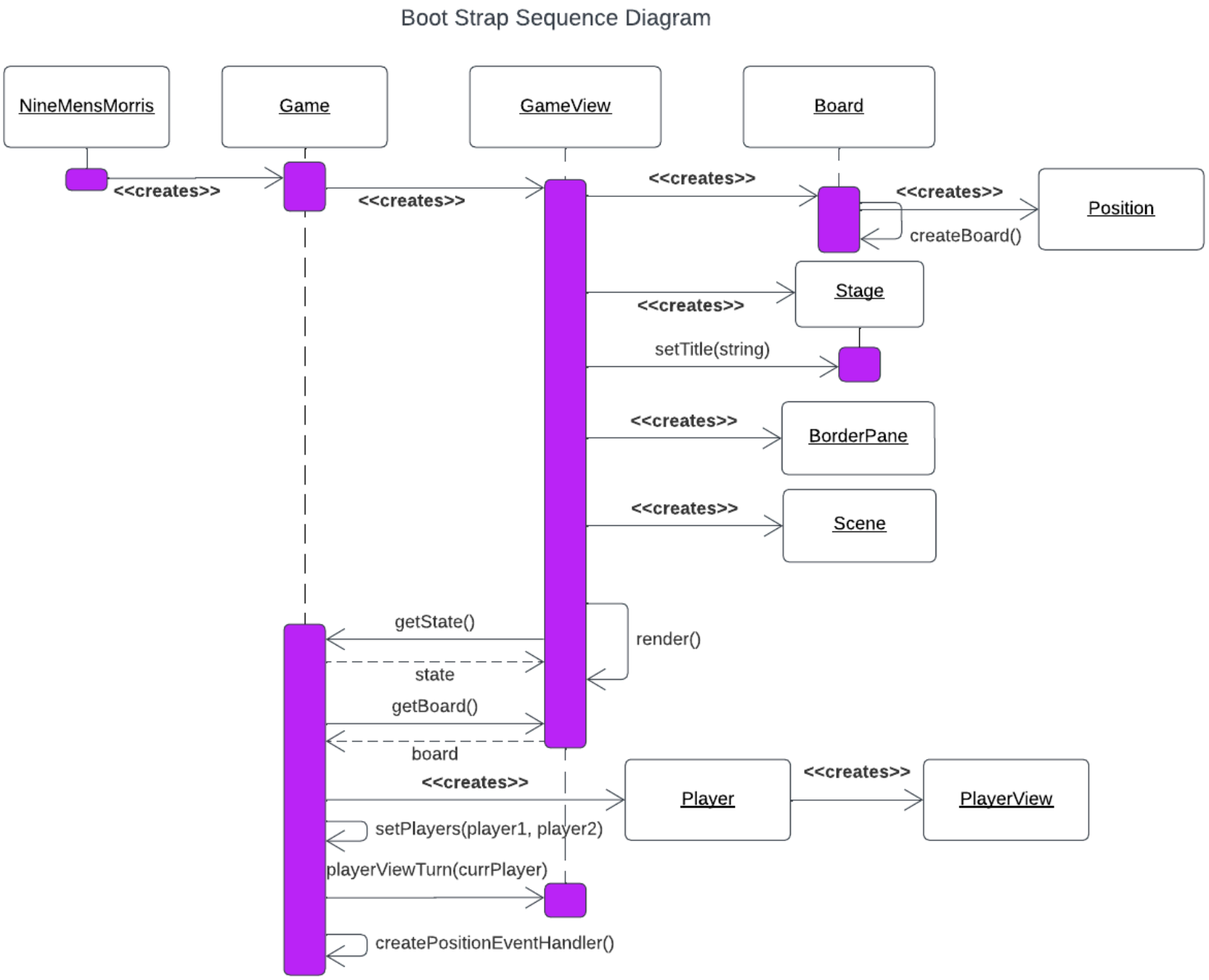
9 Men Morris Demonstration Video:

https://youtu.be/029FZ4jtkBQ

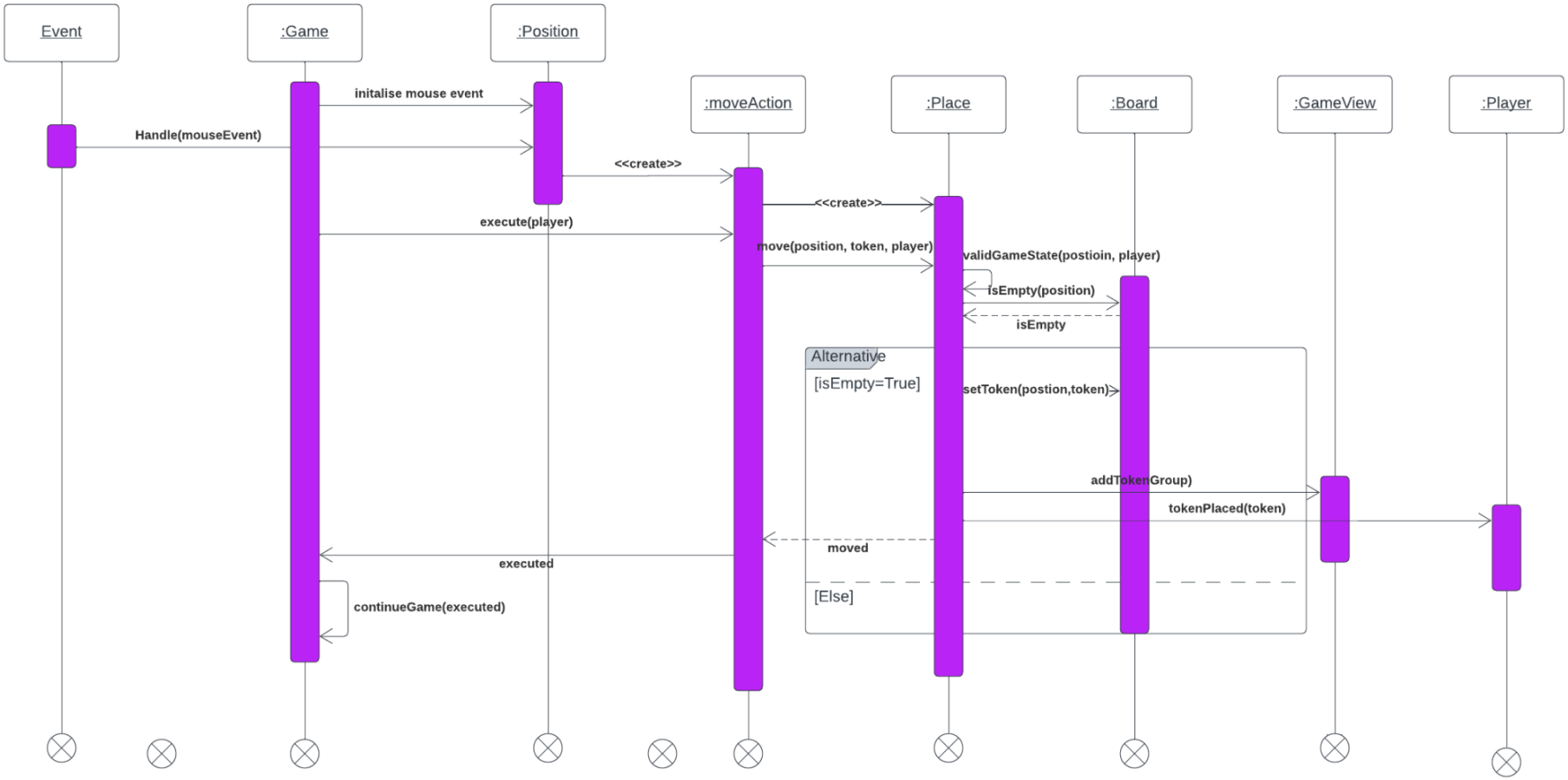
Architecture

9 Man Morris Class Diagram

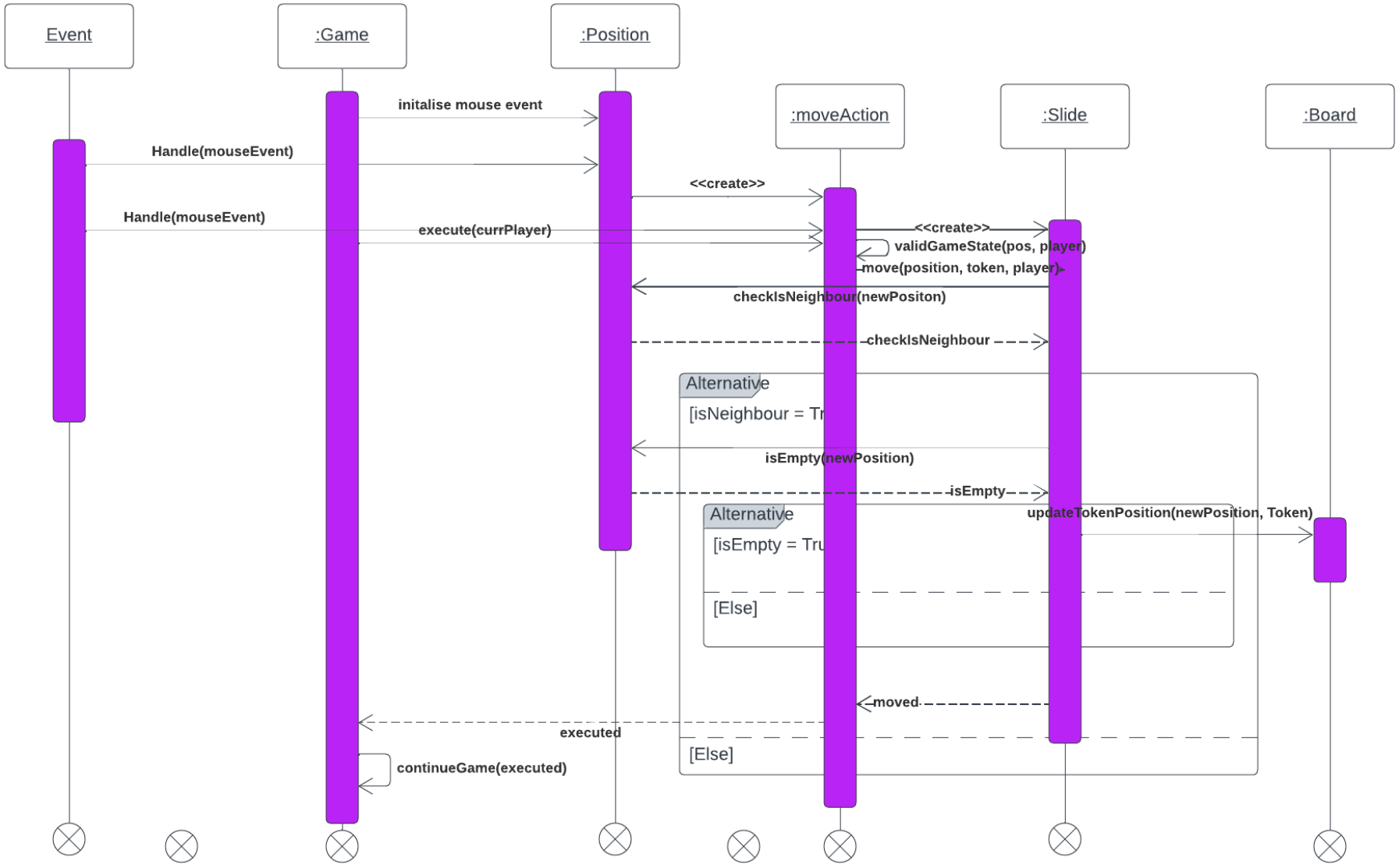



Boot Strap Sequence Diagram
NineMensMorris
Game
GameView
Board
<<creates>>
<<creates>>
<<creates>>
<<creates>>
Position
createBoard()
Stage
<<creates>>
setTitle(string)
<<creates>>
BorderPane
<<creates>>
Scene
getState()
render()
state
getBoard()
board
<<creates>>
Player
<<creates>>
PlayerView
setPlayers(player1, player2)
playerViewTurn(currPlayer)
createPositionEventHandler()

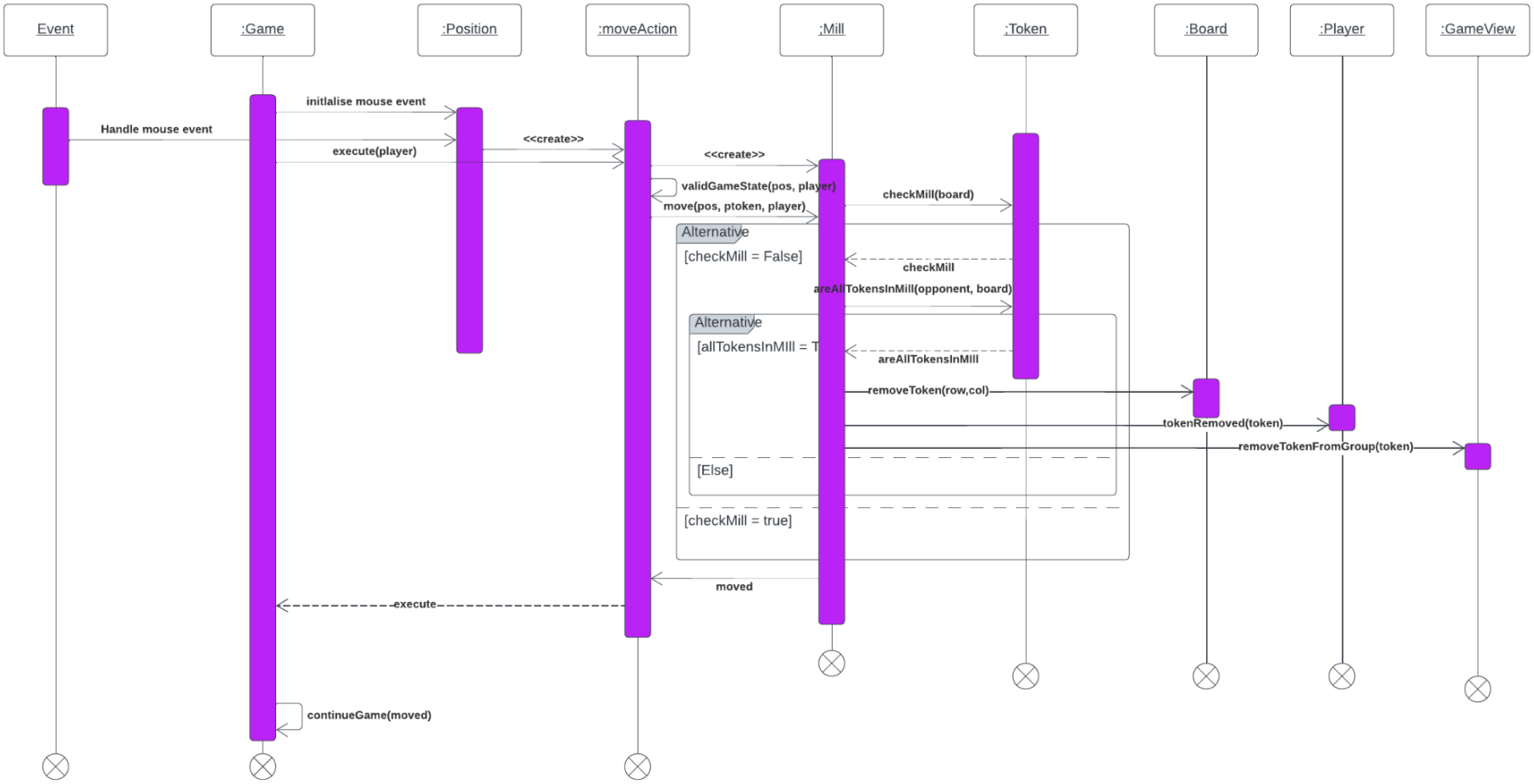
Place Sequence Diagram



Slide Sequence Diagram



Mill Sequence Diagram



Hop Sequence Diagram