LOL: Win you Runeterra
By: “N&A”

Team members:

- The Anh Nguyen – H292126 – anh.t.nguyen@tuni.fi
- Thuy Phuong Nhi Tran – H291937 – thuyphuongnhi.tran@tuni.fi

1. Introduction

LOL: Win you Runeterra (original name “Wolf Hall: a game to win the Kingdom”) is the game project given by the course Programming 3: Techniques (TIE-0240x). The game is implemented based on the base code given by the course side which the interface for card and action classes provided. Therefore, our team responsibilities are to implement our own game rules, game logic, GUI, and addition action that players can do with specific location.

Our team has implemented the minimum, intermediate requirements and some bonus/ additional features that may qualifies bonus points.

GameStory:

Once upon the time, under the ruler of The King Aurelion Sol, The nine Realms (Freljord, Demacia, Noxus, Shurima, Targon Mount, Bilgewater, Ionia, Shadow Isles) of Runeterra peacefully coexist.

One day, in order to remain the peaceful of Runeterra, The King Aurelion Sol has a fight with mysterious creature from the Void and made a heroic sacrifice. The Runeterra now in chaos as the councilor in The Realms starts preparing army to invade the other Realms.

Witnessing people misleading, The Creator of Runeterra, Riot, informs to everyone in the Runeterra: "If anyone can unify The nine Realms without war, this person can gain unlimited power and be the new The King of Runeterra."

The players, you are the savers for The Runeterra, however each of you have different approach. Therefore, you are fighting with each other to gain unity to the Realms. Each of you will begin with one assistant. The assistant acted as agent can be sent to one location that you can gain more influence the for one kingdom. The agent can gain more connection to the councilor, however when the assistant is sent to another kingdom, you will lose the connection to the previous kingdom. If lucky enough, you can find more assistant when visiting one kingdom. When you gain enough influence and connection to the councilor and the people in one kingdom, this kingdom will be by our side to help you in the way the be the new The new King of Runeterra.

BE CAREFUL! You cannot know the others influence and connection as this is there secret plan, so call the assistant back to you if you think that you cannot gain enough influence or connection to this kingdom. If someone has one kingdom by them side, your assistant will be captured, and you cannot do any action in this kingdom.

When the time is running out or the players have no assistants, except one person, this person will be The new King of Runeterra and gain unlimited power from God Riot.

Good luck all to you!

2. Software architecture

The game is implemented by course staffs and our team members. The implementation by the course staff are with namespace Interface, and our implementation use namespace: CardModel, Ui, Action, Utils, and ReadData.

Project Structure of classes implemented by the course staff:

- Card:
  - Councilor: each location has its own councilor. When one player owns this card, this player owns the whole location.
  - Influence: one influence card can only belong to one location. The player can draw randomly the influence card in their turn the gain more influence on one location.
- Location:
  - Location: each location has one councilor card and its deck which containing mixed cards. One location can only belong to one player when player can enough influence and connection.
  - Deck: each deck belongs to one location. Deck contains mixtures of one agent card and thirteen influence cards.
- Action:
  - Withdrawaction: the player can withdraw their sent agent card.
- Game:
  - Location (as mentioned before)
  - Player: each player is distinguishable by the id. The player can do actions to win a location after sent their agent card.
- Error:
  - ControlException: exception when there is action goes wrong.
  - Formatexception: exception when something is not in the correct format.
  - Gameexception: exception when something goes wrong during the game.
  - Ioeexception: exception when input or output are wrong.
  - Keyexception: exception for unknown search key usage.
  - RangeException: exception when over-indexing occurs.
  - StateException: exception class for object used whilst in illegal state.

Project structure of class implemented by our team:

- Card:
  - Agent: Each player has one initial agent card when game start. When this card is sent to one location, player is able to do further action to own one location.
- Action:
  - SendAgentCard: Player is able to send an agent to any location that is not belonging to anyone to do other actions.
  - ShuffleAndDrawAction: player is only able to do this action when their agent card is sent to one location. Player can randomly have one influence card or luckily a new agent card.
  - StrengthenConnectionAction: player is only able to do this action when their agent card. After this action, the sent agent card gain more connection to the location.
  - OwnLocationAction: player is only able to do this action when their agent card when has engough influence point and connection in this location.
- Offline Reader: read the card and location information from JSON file.

Resource

- Game intro text
- Card data
- Card and map image source

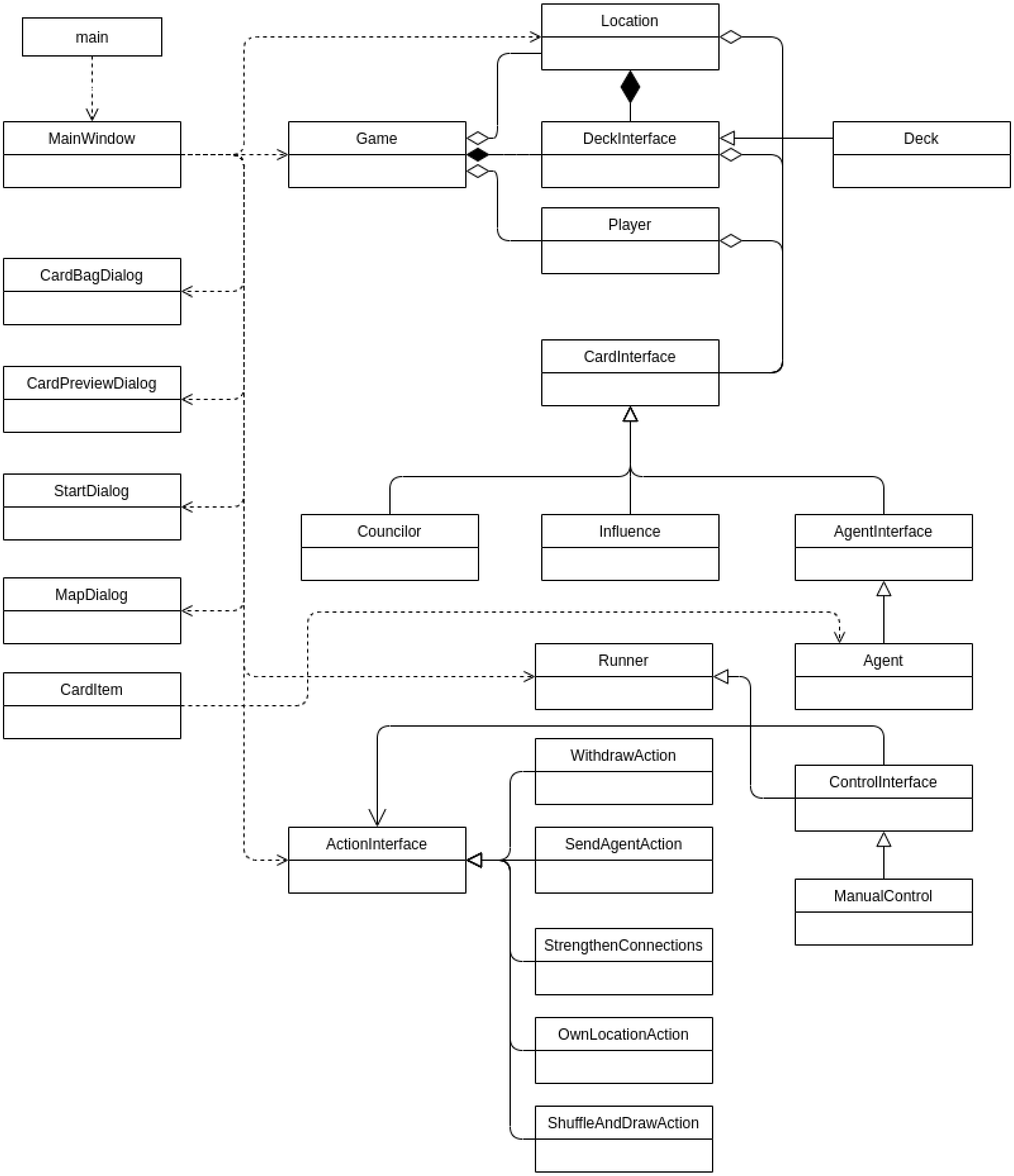


Figure 1: Class Diagram

3. Extra Feature

In addition to the minimum and intermediate requirements, we have implemented these following extra feature:

- Card as data: The cards used in the game are not hard coded, but their data is read at the beginning of the game from a JSON file.
- Location as data: The locations used in the game are not hard coded, but their data is read at the beginning of the game from a JSON file.
- Statistics of the player in turn is real-time updated. When the player does any action that affects the status (influence in chosen locations, total score, the owner of the location/councilor…), the status of the player is update automatically.
- Game setting: at the start dialog, in this setting dialog, players can choose how many locations they want to play (minimum 4 locations), game duration (minimum 1 and maximum 100 minutes), number of players (minimum 2 and maximum 4). The default player is two and there are line that player can add their own name (if the name is not given the player's name is set as default name). The maximum name length is 10 characters. The text input field in appears automatically according to the chosen number of players. After pressing the OK button, the game is set according to the player's setting.
- Card detail preview: When player hover on any card drawn in the game play area, a tooltip will appear to give detail descriptions of cards.
- Card appearance: Each card type has a specific looking that player easily distinguish them just by a look.
- Attack operation: When one player owns an area that has agent cards which are belonging to other players, these cards will be disable, other players can not withdraw or do any actions related to that agent card, which means that they lose their card. However, the owner can withdraw their agent card and add all the left cards into their own bag (not the other agent cards).
- Drawn card detail review: When the player draws a card, there will be a pop-up dialog that show the larger image of the card and the detail information about the card.
- Location background: The background of the deck game changes when the location changes.
- Game story has been created logically. There are image and figures that makes the game less boring but interesting with various location view and different card appearance.

Navigation Diagram:

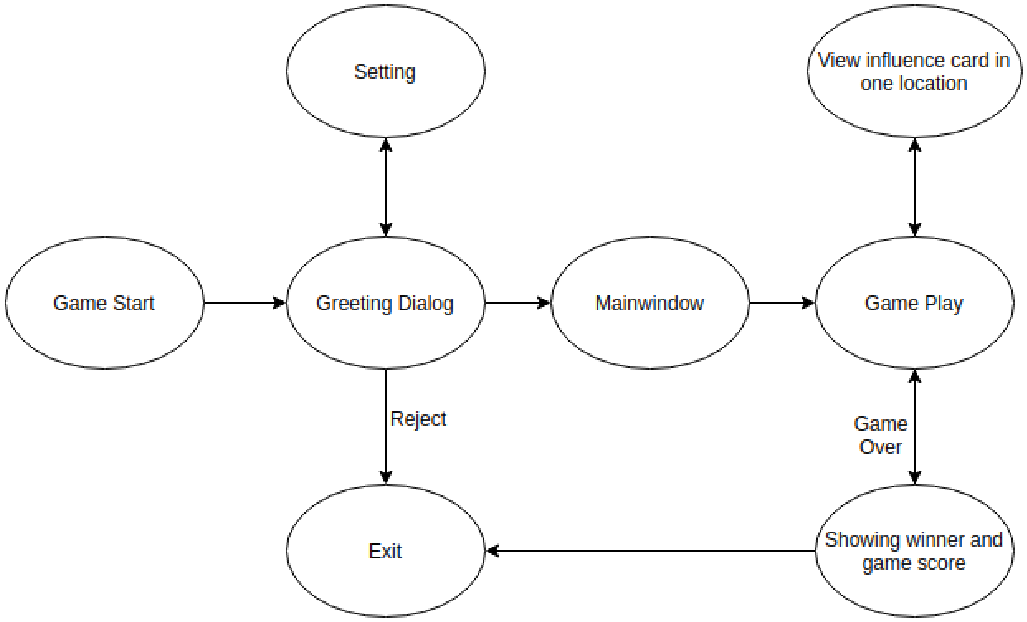


Figure 2: Navigation diagram

4. Word Division

In the beginning, before doing the project, we have many meeting to first decide the game rules and mechanism. Also, it takes time to understand the implementation code written by the course side. Then we have the basic sketching for the user interface base on the game rules by us. Mostly, the workload is divided based on the functionalities of the game.

The Anh Nguyen:

- Init the basic mainwindow UI, connecting signals and slots for classes.
- Modify the implementation of agent card and write unit test for it.
- Display the game deck on the game view area with card on it.
- Connection the action.
- Initialize the card appearance.
- Add tooltip when the user hover over the card in the game area.

Thuy Phuong Nhi Tran:

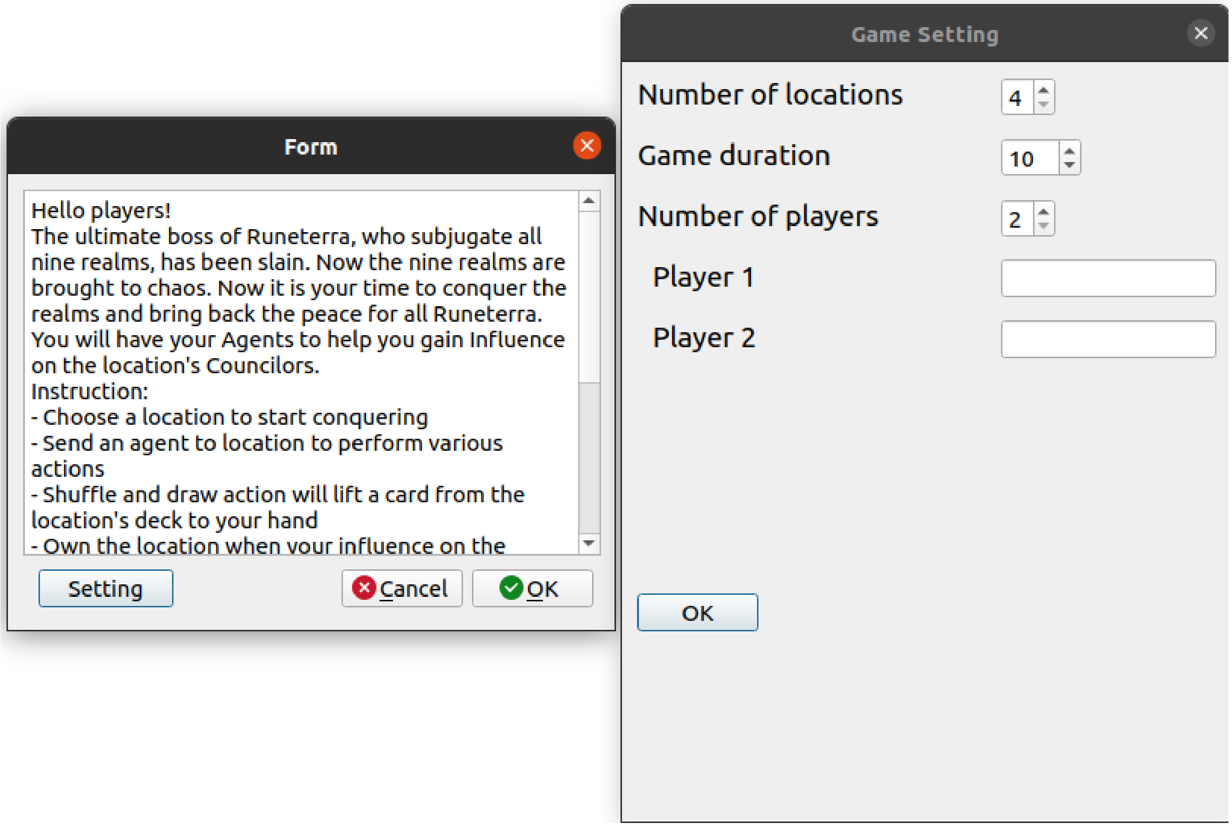
- Make the mainwindow UI in detail, connecting signals and slots for classes.
- Make the starting game dialog.
- Implementing the game game setting.
- Implementing the agent class.
- Write JSON file and implementing reading offline data reader.
- Initialize the game status based on the setting.

Both:

- Design logic and game flow.
- Handle exception.
- Make suitable data structure for storing game statistics.
- Fixing bugs and crash.
- Make the action classes and exception.
- Write documentation.

5. Game Manual

When execute the program, players are shown a greeting dialog with a brief backstory and basic instructions. In this dialog, the players are able to change the game setting.



Hello players!
The ultimate boss of Runeterra, who subjugate all nine realms, has been slain. Now the nine realms are brought to chaos. Now it is your time to conquer the realms and bring back the peace for all Runeterra. You will have your Agents to help you gain Influence on the location's Councilors.
Instruction:
- Choose a location to start conquering
- Send an agent to location to perform various actions
- Shuffle and draw action will lift a card from the location's deck to your hand
- Own the location when your influence on the

Number of locations 4

Game duration 10

Number of players 2

Player 1

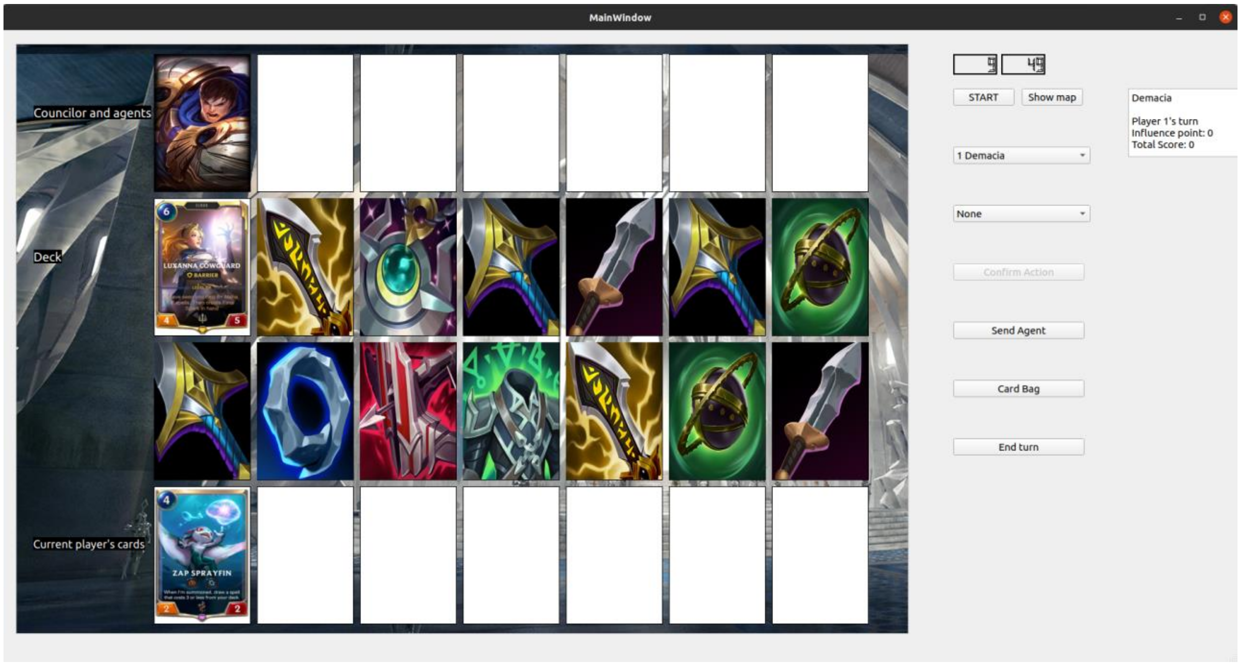
Player 2

Figure 3: game greeting and settings

If the players do not want to user default game setting, they can press the push button “Setting”, another dialog is opened and they can choose the game setting including number of locations, game duration, number of players, and players’ name. If the players do not give the name, the name of players will be set to the default name.

After accepting the settings and game instruction,, the players would be navigated to the mainwindow.

The game start when the player press START button. The game board is initialized with 4 rows and 7 columns of slots. The first row is reserved for the councilors and agents of the current location. The next two rows are reseved for the cards from deck of the current location. The last row is reserved for the agent cards that the current player has.



1 Demacia

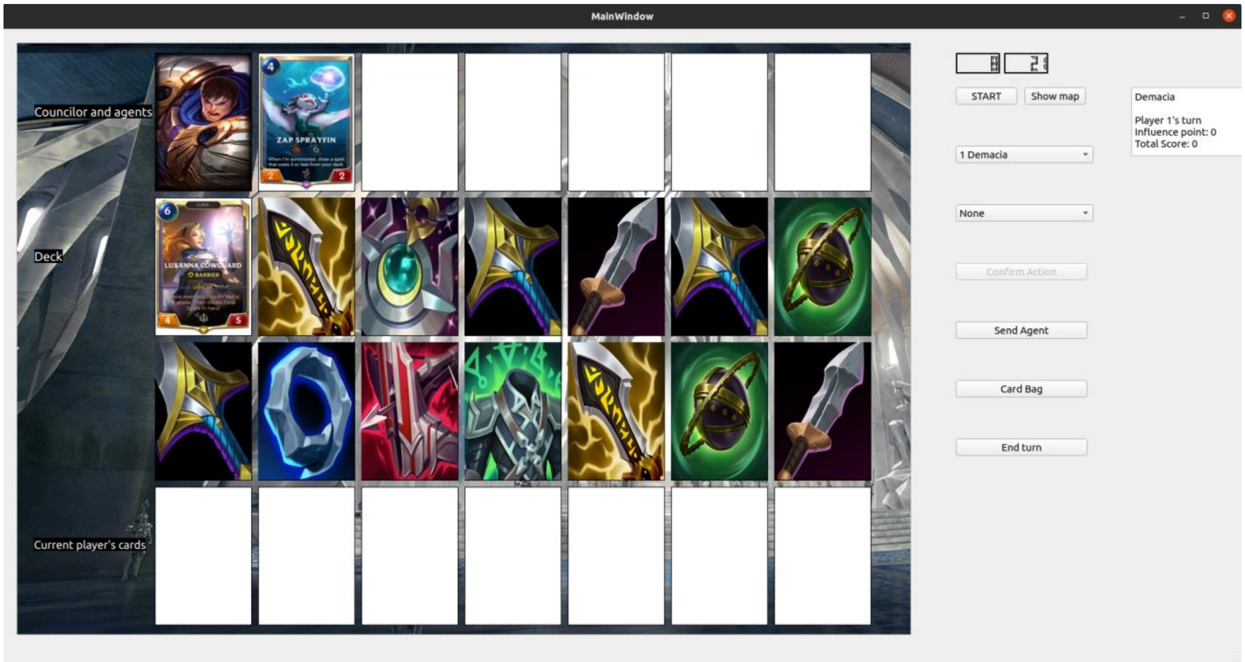
None

Demacia

Player 1's turn
Influence point: 0
Total Score: 0

Figure 4: Game play area

When an agent card is sent



1 Demacia

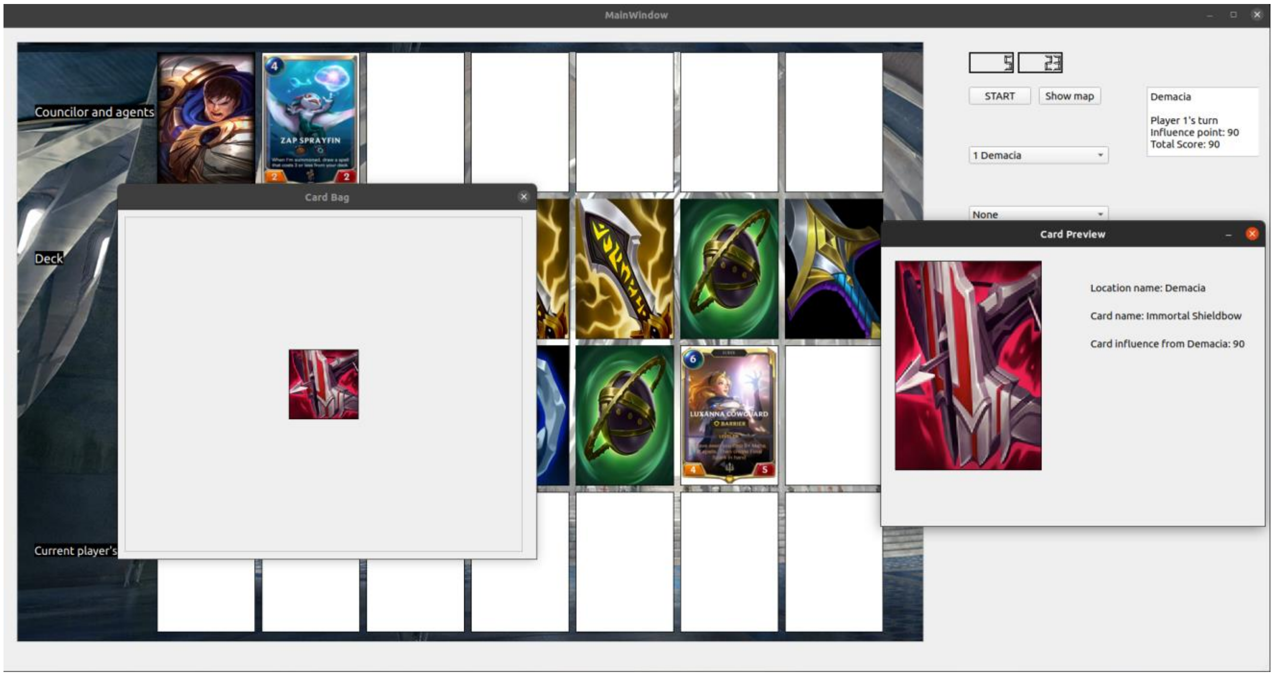
None

Demacia

Player 1's turn
Influence point: 0
Total Score: 0

Figure 5: Game area when agent card is sent.

When one player draws a card from a location:



1 Demacia

None

Demacia

Player 1's turn
Influence point: 90
Total Score: 90

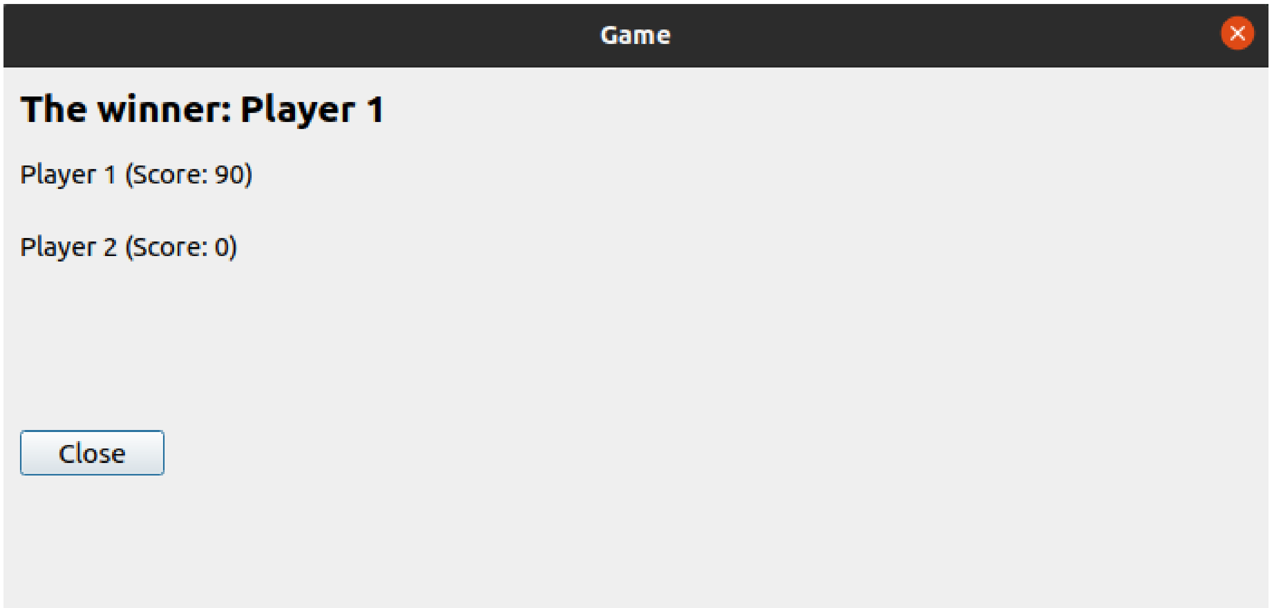
Location name: Demacia

Card name: Immortal Shieldbow

Card influence from Demacia: 90

Figure 6: Game are when one card is drawn and the current card bag

When the game end:



The winner: Player 1

Player 1 (Score: 90)

Player 2 (Score: 0)

Figure 7: Dialog to show the winner when the game ends.

6. Issues and Notes

- Hovering might take 2 to 3 seconds for the tooltip to appears. Stay till the tooltip appears.
- The text font is varied depending on different screen resolutions and scaling. This is because Qt does not automatically rescale. Therefore, it may cause missing characters.

7. Conclusion

At the moment, there is no game bugs, and all the exception can occur during the game play are handle. However, if the game duration is too long, testing the game may reveal bugs if there is any way.

Furthermore, the game can be developed in many ways. There may be more card types to increase the difficulty of the games. Each location can have different features and requirement (mini games). There can be items that agent can use the strengthen its power. However, due to the limits of time and logic provided by the course, these are not implemented. With this game idea, there are lots of room for improvement, but it takes time and work.

The game project using Design by contract techniques that course provides interface and our team use it to build our own game.