# Software Engineering Project - Phase one

Yubo Sha 22207224
Yihan Yan 22207211
Tianrui Liu 2220721
Peiqi Tian 22207209

April 2024

## 1 Game introduction

Mahjong is a four-person game that originated in China, it is an entertainment that is generally made of bamboo or bone or plastic made of small rectangular pieces, engraved with patterns or words. It contains 136 tiles. The way mahjong is played varies depending on the region and cultural background.

Tile type: Mahjong tiles are usually divided into Dot, Bamboo, and Character suits, each suit has 1 to 9 nine numbers, and each tile has four. In addition, there are some special Honor tiles, such as east, south, west, and north tile.

The goal of the game: The player's goal is to form the tiles in his hand into specific tile types, such as pairs (two identical tiles), triplets (three identical tiles), barbs (four identical tiles), and sequences (three tiles in a row).

Winning rules: When playing, players need to have a pair of tiles (pairs) and three sets of sequences or triplets. Different mahjong plays may have different winning rules.

Special rules: Chow, Pong, Kong. The last house plays tiles, and the next house tiles just form a pair of sequences, he can Chow. Someone else plays a tile, and if he has two identical tiles in his hand that makeup exactly one set, he can Pong them. Bars are divided into exposed Kong and concealed Kong, other people play a tile, and their hand has three identical tiles, called exposed Kong. There are three identical tiles in the hand, and caught an identical tile, called the concealed Kong, others do not know what tile. When you have all the tiles in your hand together into a useful tile, just add the last one can be and tile, and you can enter the stage of ready hand. Win by others means that you play a tile that just lets the other side have a tile, which means that you let the other side win.

Scoring and winning: The winner of the game is usually determined by who is the first to win. In some games, the player who wins gains points, while the player who lets others win loses points.

# 2 Game Requirements

## 2.1 User requirements:

1. Allow users to check the rules of the game.

**System requirements:**

1.1 Contain one option in the main menu to show the rule.

2. Allow the user to start the game.

**System requirements:**

2.1 Contain one option in the main menu to start the game.

3. Allow the user to end the game.

**System requirements:**

3.1 Contain one option to end the game during the game.

3.2 The system can find out whether the game is over after one player finishes his round.

3.3 The system can find out whether the game is over after one player wins.

4. Display the pattern of each tile in the player's hand and the discard pile.

**System requirements:**

4.1 The system should connect which pattern represents which type of tiles.

4.2 The system should store where the tile is (In the wall, in the player's hand or the discard pile).

4.3 The system should show the tiles as the pattern in the player's hand, discard pile and some special places.

5. Allow the user to get one tile at the beginning of his round.

**System requirements:**

5.1 The system should have a queue to store all tiles in the game (represents the wall).

5.2 The system should connect which pattern represents which type of tiles.

5.3 The player should have a place to store 14 tiles (player's hand).

5.4 The system should remember which player can have action.

5.5 The system should give the head of the queue to the player.

5.6 The player can add the tiles into the player's hand.

5.7 The system should store where the tile is (In the wall, in the player's hand or in the discard pile).

6. Allow the user to throw one tile into the discard pile to end his round.

**System requirements:**

6.1 The system should connect which pattern represents which type of tiles.

6.2 The player should have a place to store 14 tiles (player's hand).

6.3 The system should remember which player can have action.

6.4 The player can throw the tiles from the player's hand.

6.5 The system should allow other players to respond to the tiles this player throws.

6.6 The system can store the tiles in the discard pile.

6.7 The system should store where the tile is (In the wall, in the player's hand or the discard pile).

7. Allow user Pong, Kong, Chow, Exposed or Win after another player throws a tile.

**System requirements:**

7.1 The system should connect which pattern represents which type of tiles.

7.2 The system should remember which player can have action.

7.3 The system should allow other players to respond to the tiles this player throws.

7.4 If legal, the system should allow the player to get the tiles and do further action (Pong, Kong, Chow, Exposed or Win).

7.5 The system can move the tiles this player throws to the other player's hand.

7.6 System should store where the tile is (In wall, player's hand or discard pile).

8. Allow the user to see how many tiles are left in the queue.

**System requirements:**

8.1 The system should have a queue to store all tiles in the game (represents the wall).

8.2 The system should store where the tile is (In the wall, in the player's hand or in the discard pile).

8.3 The system should remember how many tiles are left in the queue (the wall).

9. Allow the user to see how many tiles are left for a given tile.

**System requirements:** 9.1 The system should connect which pattern represents which type of tiles.

9.2 The system should store where the tile is (In the wall, in the player's hand or the discard pile).

9.3 System should count for a given tile how many it doesn't show out.

10. Allow the user to see which type of tile is Hun (which can represent every type of tile).

**System requirement:**

10.1 The system should connect which pattern represents which type of tiles.

10.2 The system should store which type of tile is Hun.

10.3 The system should show which type of tile is Hun.

11. Allow users to see which tiles they need to win the game.

**System requirement:**

11.1 The system should connect which pattern represents which type of tiles.

11.2 The player should have a place to store 14 tiles (player's hand).

11.3 The system should have a function to calculate which tiles they need to win the game.

11.4 The system should show the player which tiles they need to win the game.

12. Allow users to see how many points they get or lose.

**System requirement:** 12.1 The system should connect which pattern represents which type of tiles.

12.2 The player should have a place to store 14 tiles (player's hand).

12.3 The system should have a table to record points for all forms of the player's hand.

12.4 The system should calculate the result.

12.5 The system should move points from one player to another player.

12.6 The system should find out whether the game is over.

13. Allow 4 users to join in the game on 4 different computers.

**System requirements:**

13.1 The system should contain a protocol for multi-computer interaction.

13.2 The system should allow data transfer from one player to another.

Data: such as tiles from the queue, tiles throw, whose turn . . .

13.3 One computer serves to work as the host another computer serves to access the host to get data.

13.4 Every player should see the same discard pile and tiles in their hands.

## 2.2 Functional Requirements:

1. The user shall be able to do an action (throw tiles, Pong, Kong, Chow, Exposed or Win) by clicking the mouse.

2. The system should display a game scream where the player playing the game.

3. The system should respond to user requests within an acceptable time.

4. The system should make sure all players have data synchronization.

5. The system should be robust.

## 2.3 Non-Functional Requirements:

1. Usability requirements:

User interface design should be simple.

Users can do the action easily.

2. Performance requirements:

The system should response time should be as fast as it can.

The system should maintain high resource utilization.

3. Reliability requirements:

The system should run according to plan no matter what the player clicks on the game scream.

4. Security requirements:

The code in the system should not be changed by the user (ensure fairness).

5. Scalability requirements:

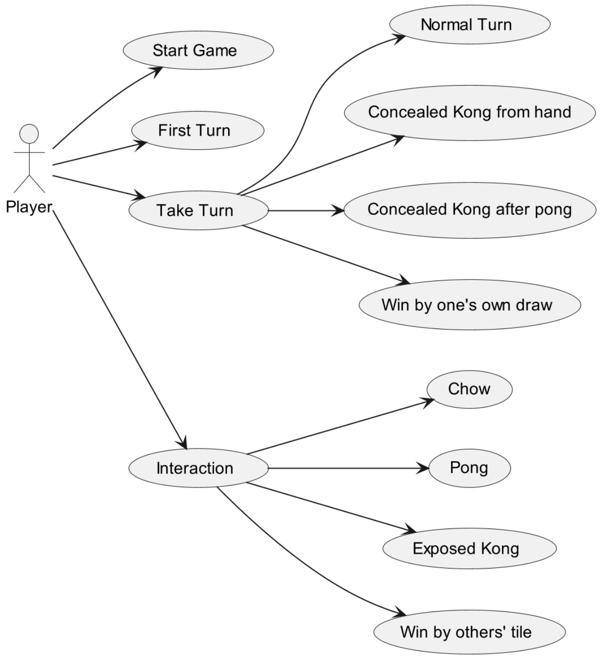
The system should allow 4 users to access the system.

6. Compatibility requirements:

Ensure that the software can run properly on different hardware, operating systems, and browsers.

# 3 UML design

## 3.1 Use-case diagram:



### 3.1.1 Use Cases

1. Start Game

2. First Turn

3. Take Turn

3a. Normal Turn

3b. Concealed Kong from hand

3c. Concealed Kong after pong

3d. Win by one's own draw

4. Interaction

4a. Chow

4b. Pong

4c. Exposed Kong

4d. Win by others' tile

Use Case 1 – Start Game:

The user starts the game

The system shows the game table with 13 tiles in front of 3 players and 14 tiles in front of one player. A tile will be shown to point out which tile is the Hun (which can represent every type of tile). The Hun is the next type of tile shown.

Use Case 2 – First Turn:

If the player has 14 tiles, they choose one to place face up on the game table

The system removes the tile from the player's hand and moves it to the game table facing up.

Use Case 3 – Take Turn:

When it is the player's turn, the system puts a tile into the tiles in front of them.

Then they will have four options:

The user performs use case 3a

The user performs use case 3b

The user performs use case 3c

The user performs use case 3d

Use Case 3a – Normal Turn:

The user selects one of the tiles in their pile to discard

The system removes the tile from the player's hand and moves it to the game table facing up.

Use Case 3b – Concealed Kong from hand:

The user selects four tiles in front of him. The system shows the selected tiles three face down and one face up for all of the players to see. (On the right side of this player)

Back to use case 3.

Use Case 3c – Concealed Kong after pong:

The user selects a tile in his hand and this tile is already pong. (Four in total)

The system puts this card together with the triplet

Back to use case 3.

Use Case 3d – Win by one's own draw

The tile the system puts in gives the player a win. (four sets of three plus a pair)

The system shows all of the user's tiles face up to the other users and has a hint to show the user is win.

The system shows all of the users that this round of the game has ended.

Use Case 4 – Interaction

After a player plays the tile, other players will have four options:

The user performs use case 4a

The user performs use case 4b

The user performs use case 4c

The user performs use case 4d

Use Case 4a – Chow:

The user selects the face-up tile that was discarded by the previous player which can make a sequence with the hand tiles.

If there is more than one sequence to chow, the system will display them on the screen for the user to choose

The system shows the selected tile together with the tiles in their place that match with it (these are shown face up for all of the players to see) (On the right side of this player)

The user selects one of the tiles in their pile to discard

The system removes the tile from the player's hand and moves it to the game table facing up.

Use Case 4b – Pong:

The user selects the face-up tile that was discarded by other players which can make a triplet with the hand tiles. The system shows the selected tile together with the tiles in their place the same as the face-up tile and shows face-up for all of the players to see. (On the right side of this player)

The user selects one of the tiles in their pile to discard

The system removes the tile from the player's hand and moves it to the game table facing up.

Use Case 4c – Exposed Kong:

The user selects the face-up tile that was discarded by another player and the user has three tiles that are the same as the tiles discarded in hand.

The system shows the selected tile together with the tiles in their place the same as the face-up tile and shows face-up for all of the players to see. (On the right side of this player)

Back to use case 3.

Use Case 4d – Win by others' tile:

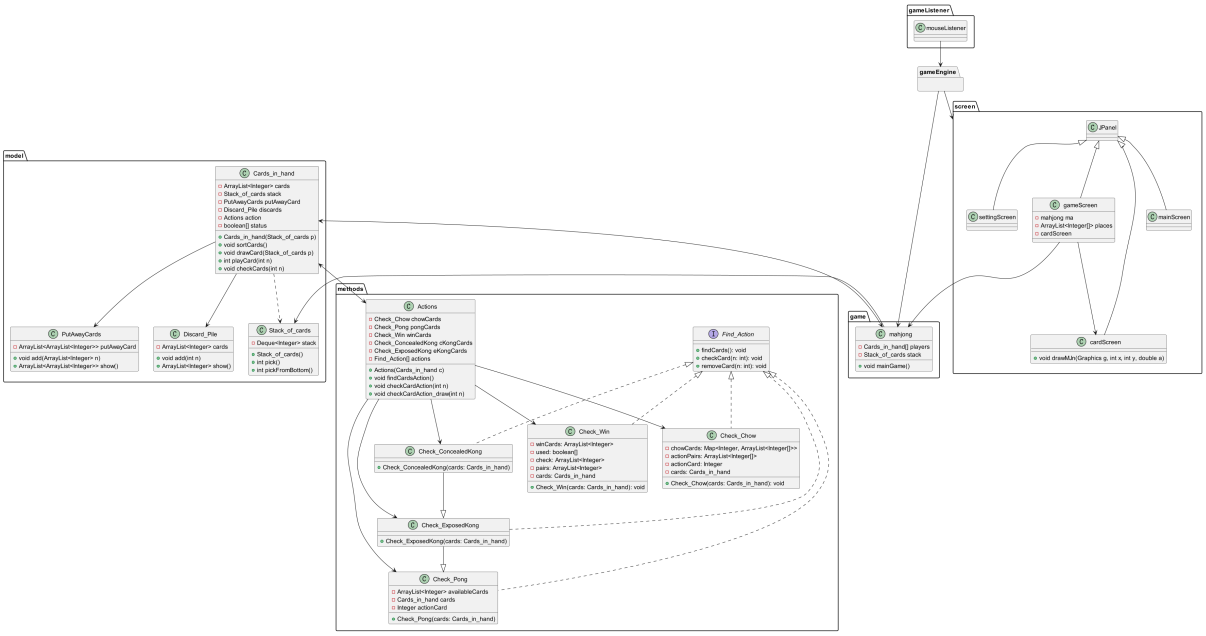
The user can only select this use case when they have never done chow, pong and exposed Kong

The user selects the face-up tile that was discarded by another player and the tile gives the user a win. (four sets of three plus a pair)

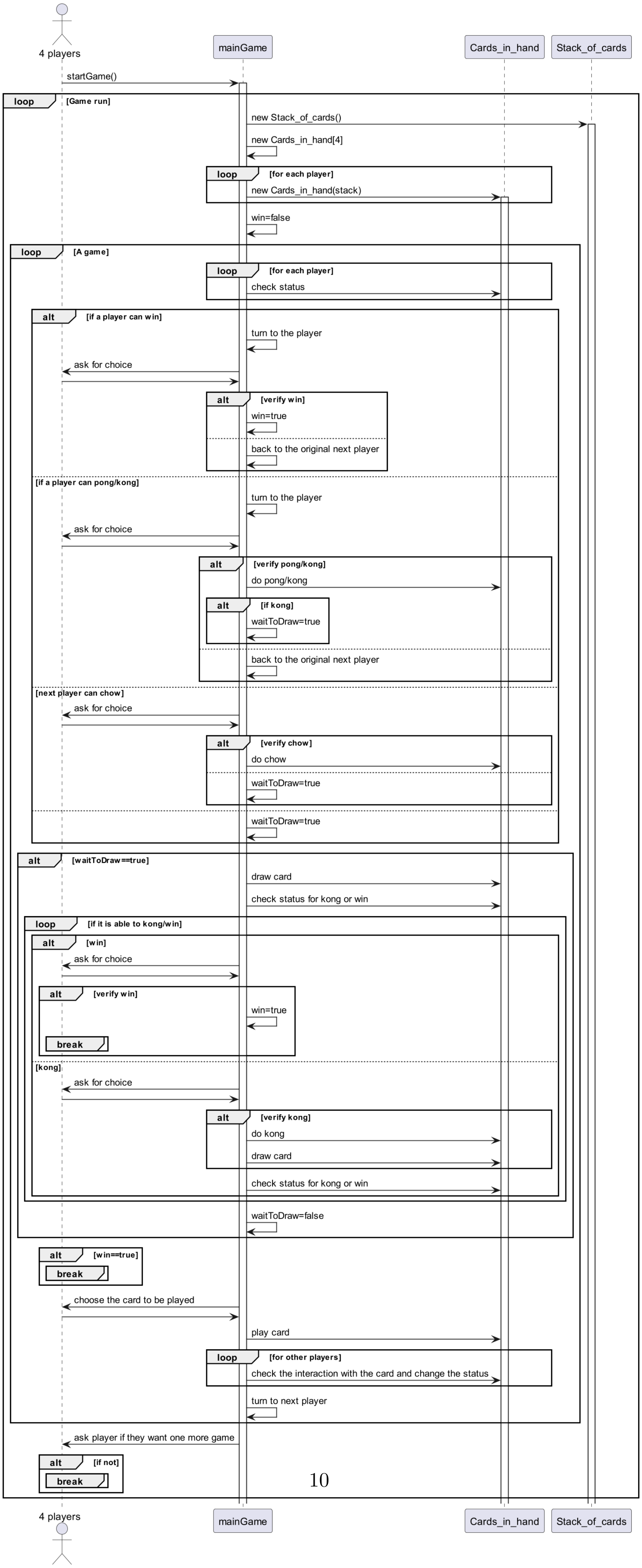
The system shows all of the user's tiles face up to the other users and has a hint to show the user is winning.

The system shows all of the users that this round of the game has ended.

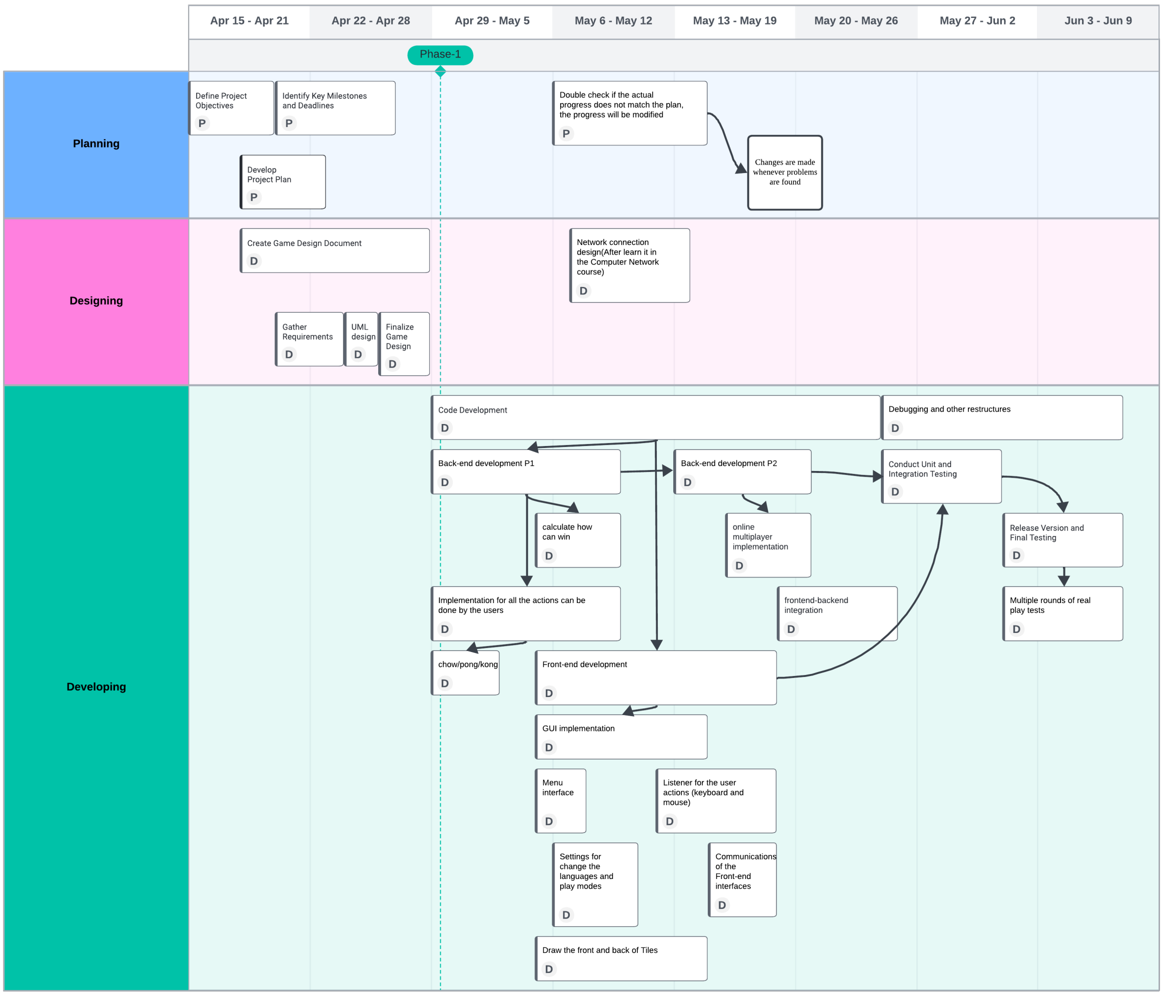
## 3.2 Class diagram



## 3.3 Sequence diagram


4 players
mainGame
Cards_in_hand
Stack_of_cards
startGame()
loop [Game run]
new Stack_of_cards()
new Cards_in_hand[4]
loop [for each player]
new Cards_in_hand(stack)
win=false
loop [A game]
loop [for each player]
check status
alt [if a player can win]
turn to the player
ask for choice
alt [verify win]
win=true
back to the original next player
[if a player can pong/kong]
turn to the player
ask for choice
alt [verify pong/kong]
do pong/kong
alt [if kong]
waitToDraw=true
back to the original next player
[next player can chow]
ask for choice
alt [verify chow]
do chow
waitToDraw=true
waitToDraw=true
alt [waitToDraw==true]
draw card
check status for kong or win
loop [if it is able to kong/win]
alt [win]
ask for choice
alt [verify win]
win=true
break
[kong]
ask for choice
alt [verify kong]
do kong
draw card
check status for kong or win
waitToDraw=false
alt [win==true]
break
choose the card to be played
play card
loop [for other players]
check the interaction with the card and change the status
turn to next player
ask player if they want one more game
alt [if not]
break
4 players
mainGame
Cards_in_hand
Stack_of_cards

# 4 Project completion plan


Apr 15 - Apr 21
Apr 22 - Apr 28
Apr 29 - May 5
May 6 - May 12
May 13 - May 19
May 20 - May 26
May 27 - Jun 2
Jun 3 - Jun 9
Phase-1
Planning
Define Project Objectives P
Identify Key Milestones and Deadlines P
Develop Project Plan P
Double check if the actual progress does not match the plan, the progress will be modified P
Changes are made whenever problems are found
Designing
Create Game Design Document D
Network connection design(After learn it in the Computer Network course) D
Gather Requirements D
UML design D
Finalize Game Design D
Developing
Code Development D
Debugging and other restructures D
Back-end development P1 D
Back-end development P2 D
Conduct Unit and Integration Testing D
calculate how can win D
online multiplayer implementation D
Release Version and Final Testing D
Implementation for all the actions can be done by the users D
frontend-backend integration D
Multiple rounds of real play tests D
chow/pong/kong D
Front-end development D
GUI implementation D
Menu interface D
Listener for the user actions (keyboard and mouse) D
Settings for change the languages and play modes D
Communications of the Front-end interfaces D
Draw the front and back of Tiles D

# 5 Member contributions & divide the work

Game front end: Tian Peiqi and Yan Yihan

Game back end: Sha Yubo and Liu Tianrui

Tian Peiqi:

Making the front and back of mahjong tiles

The background design of the GUI.

Yan Yihan:

Do the project management.

Document compile.

Game front-end design and implement.

ShaYubo:

Write the main program that controls the game. Implement the player actions like Pong, Chow, Kong and so on. Achieve the consolidation of the game codes.

LiuTianrui

Write the codes for some basic functions. Summarize the requirements. Implement the judge of the tiles the user needs to win the game. Implement the arithmetic to count the score.