Project Report

Group 8
COMP2021 Object-Oriented Programming (Fall 2018)
17081996d Gao Haorui
17082705d Qin Yaxue
16098537d Yu Jing
17083686d Xia Jialu

1 Introduction

This document describes the design and implementation of the Jungle game by group 8. The project is part of the course COMP2021 Object-Oriented Programming at PolyU. The following sections describe the requirements that were implemented and the design decisions taken. The last section describes the available commands in the game.

2 The Jungle Game

The game is divided into two sections, with two rivers and bridges in between. The chess manual is of seven columns and nine rows with 8 different pieces of each player in the grid. The ranks of pieces is arranged: elephant > lion > tiger > leopard >wolf > dog > cat > rat. The higher ranking can eat the one with identical and lower ranking, but rats can eat elephants. Rats can swim across rivers, but rats cannot capture elephants or other rat on land directly from a water square. Similarly, animals on land cannot attack rats in the water. The rat may attack the opponent rat if both pieces are in the water or on the land. Lions and tigers can jump over rivers. But if rats are in rivers, lions and tigers can't jump over the river. Only the one who moves any piece into opponent's den or capture all of the opponent's pieces will be the winner.

2.1 Requirements

REQ01: When the program is launched, a user will be able to choose between starting a new game and opening a saved game;

Class **Console**, method **private void start()**: prompt user to choose options by entering 1 or 2.

Option 1: starting a new game

- Class *JungleGame*, method *public void openNew()***:** set turn (=0 for X go first), 2 players, 16 Pieces and set them in the board in corresponding positions.

Option 2: opening a saved game, and once it is chose, users will be prompt to input a file path to be opened.

- Class *Console*, method *public String open(String path)***:** return the string in the saved game file by the provided file path.
- Class *JungleGame*, method *public void openSaved(String savedStr)*: set turn, 2

players, 16 Pieces and set them in the board in corresponding positions using the String savedStr parameter.

REQ02: After starting a new game, two players are prompted to input their names. Then program prints the initial the board on the screen and ask the first player to input a command.
Software element:

- Class *Console, method public void start()*: to get the names of the two players

*Use 'Message.**MSG_NAMEX**.getMessageStr()' && 'scanner.nextLine()' to allow the two players to enter their names and also store.*

- Class *JungleGame*: to print the new broad

Use 'board[][].getpiece' to get the piece value and then use a correct format to print them out.

- Class *Console, method public void run()*: Ask the first player (X) to input a command:

*Use 'Message.**MSG_INPUT_COMMAND**.getMessageStr()' && 'scanner.nextLine()' to get the command and use "execute" to do this command.*

REQ03: There are total four types of command
Class Console, public enum Command:

CMD_SAVE("save"),
CMD_OPEN("open"),
CMD_MOVE("move"),
CMD_EXIT("exit");

Used in method: private boolean execute(String string) that will switch four commands case:

1: save command

- Users need to enter "save [filePath]", e.g. save C:\test.txt. If the file does not exist, a new file with specified filename will be created. Once saving successfully, a message will appear in command line. The program will catch exceptions for some conditions and prompt users to enter their command again.
- Method: public boolean save(String path, String toSave) that can save the String toSave into file path.

2: open command

- Users need to enter "open [filePath]", e.g. open C:\test.txt. The program will firstly check whether the current game has been saved by private filed *boolean saved*. If not, method *public boolean savePrompt()* will prompt users and provide two options: entering 1 for back to the current game to save, entering 2 for continue anyway [2 -> Class: *JungleGame*, method: *public void openSaved(String savedStr)* (see REQ01); 1 -> return true (directly return to the current game, and that user

can enter command again)].

- Method: *public String open(String path)* that can read the file and return in String type. If IOException is caught, users will be prompted to enter the command again.

3: move command

- Class *JungleGame*: Execute movement

  Method:

  move (position1, position2) to judge whether the position1 has a piece on it, if true->

  move (piece, position2) which first judge whether it is a valid move -> isValidMove

  isValidMove (piece, position2) to judge 1. Whether it is your turn {we use 0 or 1 to judge whose turn} 2. The piece must move. 3. (The piece could only move horizontally or vertically with one grid && this piece can't jump river) || (piece can jump river && there is no rat block it)

  if isValid, go to move (piece, position2) to judge whether it could move to position2
  if notvalid, print the err message and allow the user to enter again except for not your turn.

4: exit command

- *Users can exit the game by entering "exit", then the program will run System.exit(0).*

REQ04: Only valid command would be executed. Invalid commands would not influence the game state.

- Class Console, method execute() will return boolean that indicating whether the input command is valid. If return false, users will be prompted to input again:

```
String s = scanner.nextLine();
    boolean e = execute(s);
    while (!e) {
        System.out.println(Message.MSG_INVALID_INPUT.getMessageStr());
        s = scanner.nextLine();
        e = execute(s);
    }
```

- Class *JungleGame*: To judge if the input is a valid movement

  Method:

  If it is a valid move (Boolean isvalidmove (piece, position) && Boolean move (piece, position)), piece.setposition() to both the old and new position will update the board and also change the turn (turn=1-turn) of the game.

  If it is not valid, just return false.

REQ05: After each valid move, the updated game board will be printed. Then the

program would check whether the goal is achieved or not, if achieved then game is over and print the name of winning player; otherwise current player's turn would be terminated, and the other player would be prompted to input next command.

- Class *Console*, method *run() (after //step2)*: to check if any player has achieved the goal after each move command input, If not, exchange the turn and continue the game. If yes, print out the name of winner.

- Class JungleGame, method *public int check()*: to check if any player has achieved the goal. If yes, to find the player who is the winner (if player X is winner, return 1; else return 2). If no, return 0.
  Method: int check();
  If all piece. getstatus() ==false { return 1, if x win, 2, if y win} ||
  If xden.getpiece!=null, return 2; else if yden.getpiece!=null, return 1.

REQ06: Upon an invalid command then error messages should be shown and the same player would be prompted to input another move. If the invalid command is a move command, the current game board should also be printed.

- (See REQ04) error messages: *Message.MSG_INVALID_INPUT in Class Message*
- (See REQ04) use a while loop to let users to input again if the command is invalid.
- In method private boolean execute(String string), case CMD_MOVE: run print() method after running the method move() in class JungleGame:

```
case CMD_MOVE:

    boolean b = game.move(str[1],str[2]);

    game.print();

    return b;
```

Bon01: the game should have a full-fleged GUI mode

1. Method 1: actionPerformed

Get action command. If the string is new, open a new game; if the string is saved, open the saved file; if the string is filepath, select save path.

2. Method 2: initGui

Draw the panel and set menu bar, use frame, Toolkit and JMenuBar to construct the panel

```
frame.setSize(1000, 1000);
  Toolkit kit = Toolkit.getDefaultToolkit(); //define the tool kit
```

frame.setJMenuBar(*menuBar*);

Method 3: openNew

Open a new game:

- set all the pieces, traps and dens to their initial position
- add mouse listener to get the input of click by mouse
  first, click the targeted piece user wants to move. Then the program gets first click. While the click is of no piece, show the message dialog that "first-click must be one animal piece!" After getting the first valid click, the program needs to capture the second position that the user wants to move piece to. Use function isValid to ensure the move is valid
- input the names of player in the dialog box to remind which players should input command

3. Method 4: openSavePath
   - Open an old game
     Get the path of the old game and then open it.

```
openFile = new FileDialog(frame, "open file", FileDialog.LOAD);
openFile.setVisible(true);
String dirName = openFile.getDirectory();
String fileName = openFile.getFile();
File file = new File(dirName + fileName);
```

- read the content of the old game

```
FileInputStream fis = new FileInputStream(file);
int len = fis.available();
byte[] data = new byte[len];
fis.read(data);
```

- Then, initialize the chess manual with old pieces position

4. Method 6: selectSavePath
   - Select path to save the game. It's a little like to get the path of the old game.
   - Record each piece position and each piece status in the chess manual

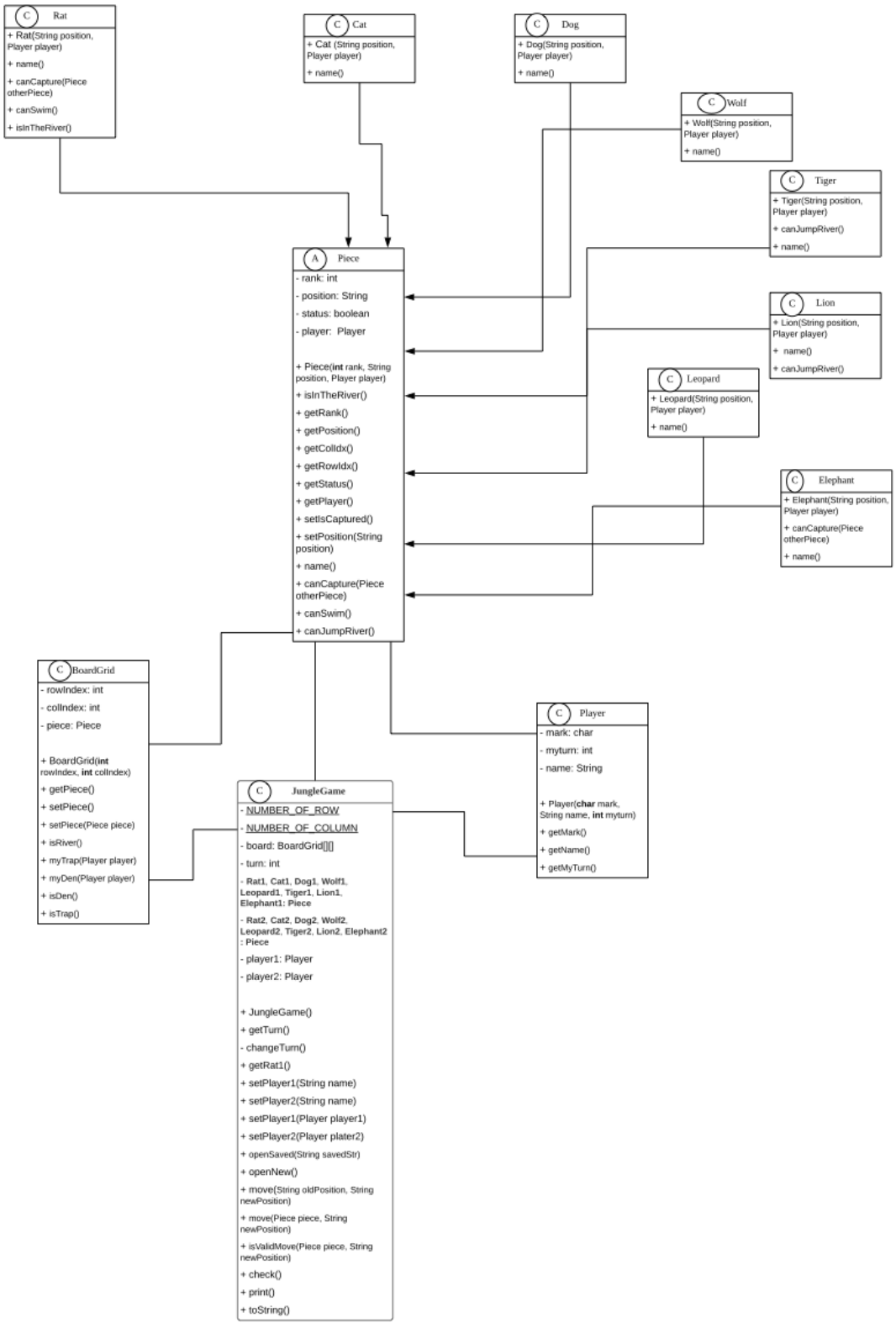
```
stringBuilder.append(board[i][j].getPiece().getRank()+","+board[i][j].getPiece().getStatus()
  + "," + board[i][j].getPiece().getPosition() + "," + board[i][j].getPiece().getPlayer().getMark()
+","+board[i][j].getPiece().getPlayer().getName()+","+board[i][j].getPiece().getPlayer().getM
yTurn());
```

- Write into file

2.2 Design

Our design is overall a Model-View-Control pattern, and for the view part, we could operate both GUI and Command-line to let the player play the game.

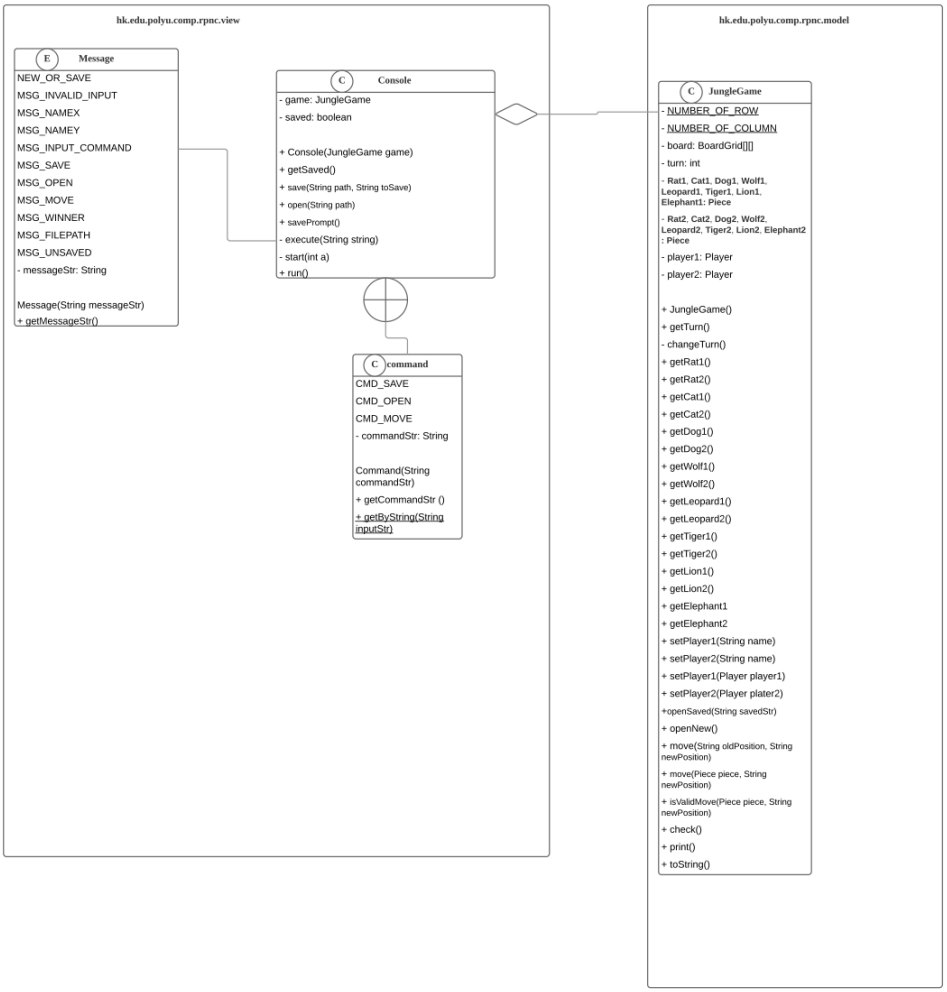
Here is our Diagram for the Model part.



We have 4 main classes, and the ‘Piece’ has 8 sub-classes which extends to ‘Piece’(e.g. Cat extends Piece) ‘How to play’ part mainly gathers in ‘JungleGame’,

JungleGame should have two players and several pieces and a GridBoard. Also the Pieces are got/set from/to the GridBoard, and each piece has an owner player. In the above logic, we combine them together.

For our View diagram.



We use the console to connect to the game and also allow the I/O.

2.3Quick Start Guide

2.3.1 Game using command line

Step1: Run the game and you will see the Welcome page

```
Welcome to Jungle game.
- Start a new game: Enter 1
- Open a saved game: Enter 2

```

Step2: Input 1 to start a new game; Input 2 to open a saved game

If choose 1 - start a new game: GO TO **Step3**
If choose 2 – open a saved game: GO TO **Step7**

Step3: Input two players' name for new game

```
Please input player X's mame
player1
Please input player Y's mame
player2
```

Step4: The game board will be printed out. Player1 to input command:

```
                          X
LionX             Trap    Den     Trap              TigerX
         DogX             Trap             CatX
RatX              LeopardX          WolfX             ElephantX
         River    River             River    River
         River    River             River    River
         River    River             River    River
ElephantY         WolfY             LeopardY          RatY
         CatY             Trap             DogY
TigerY            Trap    Den     Trap              LionY
                          Y
player1, please input a command:

```

Step5.1: Move Command type must be "move Capital Letter + number Capital Letter + number"

```
player1, please input a command:
move A9 A8
```

If the command is valid, program return true; else return false. Then Player2 input a command

```
player1, please input a command:
move A9 A8
                          X
                Trap      Den       Trap              TigerX
LionX     DogX            Trap              CatX
RatX              LeopardX          WolfX             ElephantX
          River     River             River     River
          River     River             River     River
          River     River             River     River
ElephantY         WolfY             LeopardY          RatY
          CatY              Trap              DogY
TigerY            Trap      Den       Trap              LionY
                           Y
player2, please input a command:
```

Step5.2: Command “save [filePath]”

```
                          X
                Trap      Den       Trap              TigerX
LionX     DogX            Trap              CatX
RatX              LeopardX          WolfX             ElephantX
          River     River             River     River
          River     River             River     River
          River     River             River     River
ElephantY         WolfY             LeopardY          RatY
          CatY              Trap              DogY
TigerY            Trap      Den       Trap              LionY
                           Y
gao, please input a command:
save C:\Code\Java\test.txt
C:\Code\Java\test.txt File Created
Save successfully!
gao, please input a command:
```

Step5.3: Command “open [filePath]”

If entering open command when not saving the current game:

Option 1:

```
Y, please input a command:
OPEN C:\Code\Java\text3.txt
Your current game has not been saved. Are you sure to unsave and open another one?
- No, back to save: Enter 1
- Yes, continue anyway: Enter 2

1
                          X
LionX             Trap    Den     Trap              TigerX
                          Trap            CatX
RatX      DogX    LeopardX          WolfX             ElephantX
          River   River             River   River
          River   River             River   River
          River   River             River   River
ElephantY         WolfY             LeopardY          RatY
          CatY              Trap            DogY
TigerY            Trap    Den     Trap              LionY
                          Y
Y, please input a command:
```

Option 2:

```
LionX             Trap    Den     Trap              TigerX
          DogX            Trap            CatX
RatX                      LeopardX  WolfX             ElephantX
          River   River             River   River
          River   River             River   River
          River   River             River   River
ElephantY         WolfY             LeopardY          RatY
          CatY              Trap            DogY
TigerY            Trap    Den     Trap              LionY
                           Y
GAO, please input a command:
OPEN C:\Code\Java\test2.txt
Your current game has not been saved. Are you sure to unsave and open another one?
- No, back to save: Enter 1
- Yes, continue anyway: Enter 2

2
                           X
LionX             Trap    Den     Trap              TigerX
          DogX            Trap            CatX
RatX              LeopardX          WolfX             ElephantX
          River   River             River   River
          River   River             River   River
          River   River             River   River
ElephantY         WolfY             LeopardY          RatY
          CatY              Trap            DogY
TigerY            Trap    Den     Trap              LionY
                           Y
P1, please input a command:
```

Step6.1: Two players take turns. If one player is winner, the game will show the name of winner and over.

```
                            X
                  Trap      WolfY     Trap                TigerX
LionX     DogX                        Trap                CatX
                  LeopardX            WolfX               ElephantX
          River   River               River     River
          River   River               River     River
          River   River               River     River
RatX                                  LeopardY            RatY
          CatY                        Trap                DogY
TigerY            Trap      Den       Trap                LionY
                            Y
The winner is yuki
```

Step6.2: Exit the game when the game is not over.

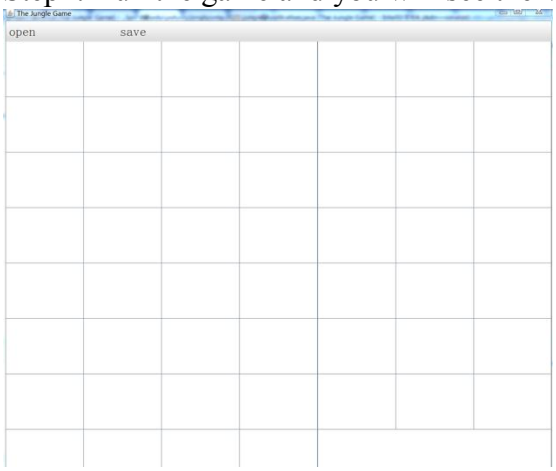
```
Welcome to Jungle game.
- Start a new game: Enter 1
- Open a saved game: Enter 2

2
Please input the path of the file:
C:\Code\Java\test.txt
                            X
LionX             Trap      Den       Trap                TigerX
          DogX                        Trap                CatX
RatX              LeopardX            WolfX               ElephantX
          River   River               River     River
          River   River               River     River
          River   River               River     River
ElephantY         WolfY               LeopardY            RatY
          CatY                        Trap                DogY
TigerY            Trap      Den       Trap                LionY
                            Y
YU, please input a command:
exit

Process finished with exit code 0
```

Step7: Open a saved file by entering the file path

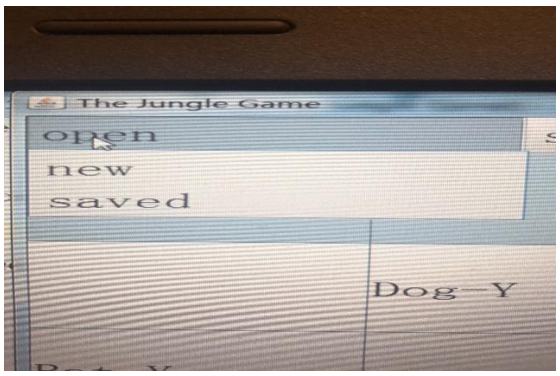
```
Welcome to Jungle game.
- Start a new game: Enter 1
- Open a saved game: Enter 2

2
Please input the path of the file:
C:\Code\Java\test.txt
                                X
LionX               Trap        Den         Trap                TigerX
          DogX                  Trap                  CatX
RatX                LeopardX              WolfX                 ElephantX
          River     River                 River       River
          River     River                 River       River
          River     River                 River       River
ElephantY           WolfY                 LeopardY              RatY
          CatY                  Trap                  DogY
TigerY              Trap        Den         Trap                LionY
                                Y
YU, please input a command:
MOVE C7 D7
                                X
LionX               Trap        Den         Trap                TigerX
          DogX                  Trap                  CatX
RatX                                        WolfX                 ElephantX
          River     River                 River       River
          River     River                 River       River
          River     River                 River       River
ElephantY           WolfY                 LeopardY              RatY
          CatY                  Trap                  DogY
TigerY              Trap        Den         Trap                LionY
```

2.3.2 Game with GUI

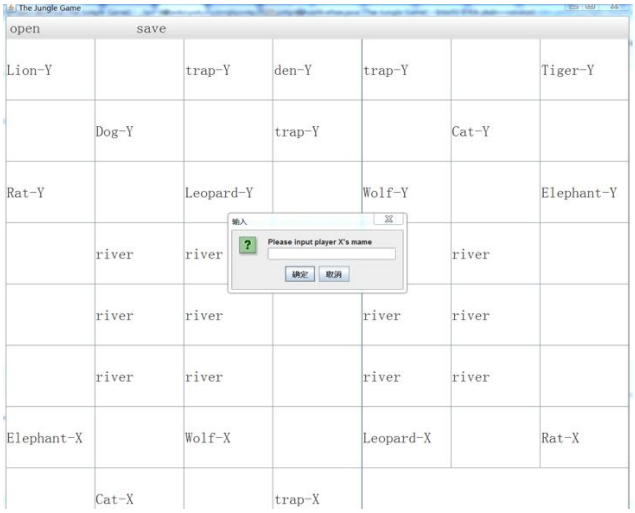
Step1: Run the game and you will see the Welcome page



Step2: click Open



Step3: Select new game

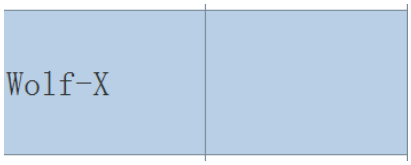


Step4: Player1 input name

Step5: Player2 input name

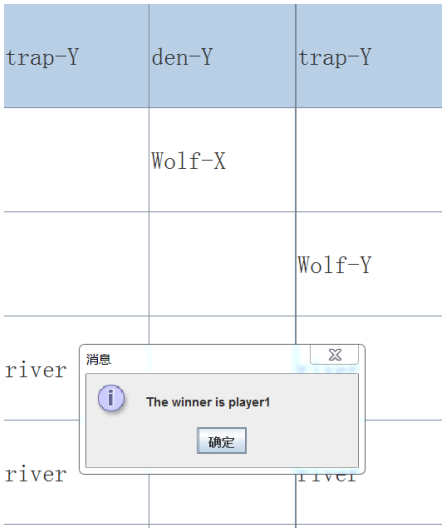
Step6: Player1 to input command

Step7: click animal piece and then click the neighbor block for movement

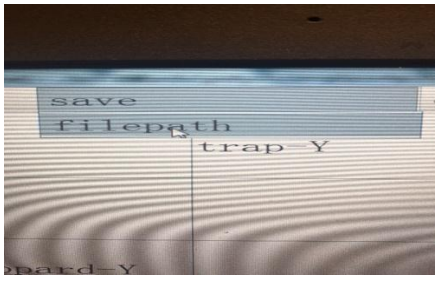


Step8: Player2 input a command

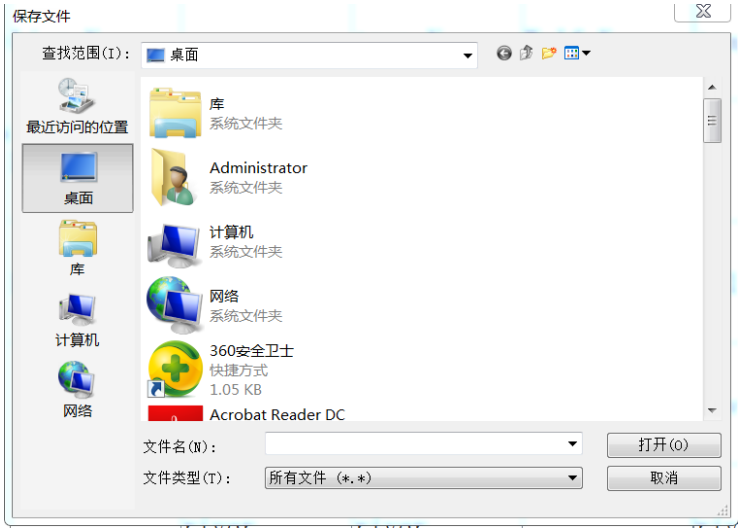
Step9: Two players take turns. If one player is winner, the game will show the name of winner.



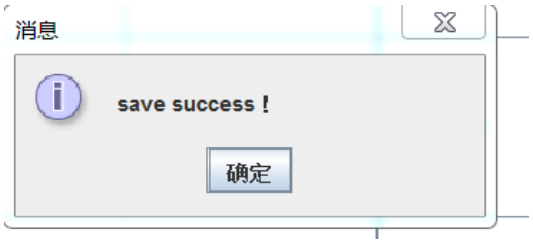
Step10: Save the game



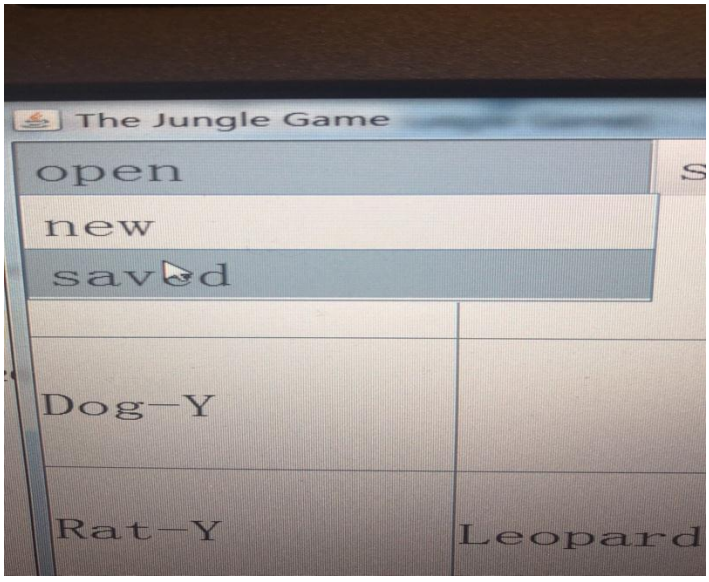
Step11: Select the save path and input the file name



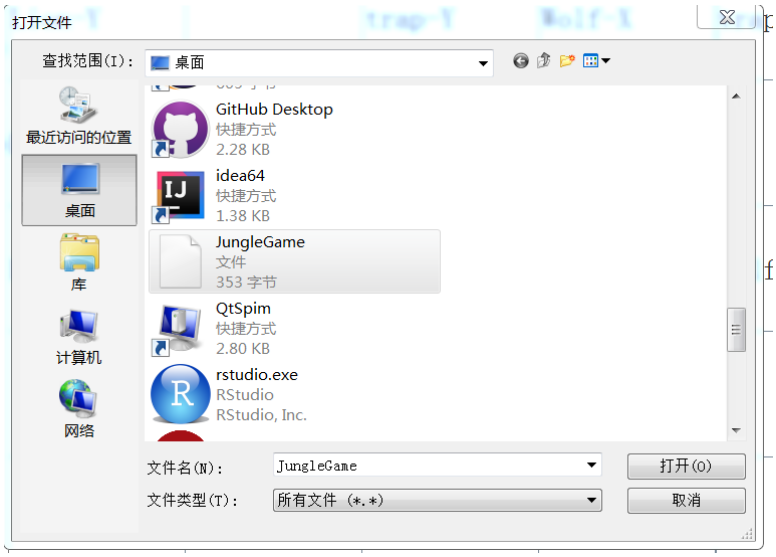
Step12: After save, you will see “save success!” message



Step13: Open a saved game, click “Saved”



Step14: Find the file you want to open



Step15: You will open your saved game

Lion-Y		trap-Y	Wolf-X	trap-Y		Tiger-Y
Dog-Y			trap-Y		Cat-Y	
Rat-Y	Leopard-Y			Wolf-Y		Elephant-Y
	river	river		river	river	
	river	river		river	river	
	river	river		river	river	
Elephant-X				Leopard-X		Rat-X
	Cat-X		trap-X		Dog-X	