FOOP final report
—— team 12 豆腐金鋼隊

(1) The team members' names, school IDs, responsibility of our team member

B01902027 Yi-Chung Wu : have participate in all parts execpt AI.
B01902103 Ya-Yuan Cheng : report, UI and Animation
B01502036 Kuo-Wei Lai : report game design, AI
B01502130 Yu-Nien Huang : report game design, AI

(2) Game Overview and rule

This game is adapted from **"Romance of the Three Kingdoms"** which is originally designed as a 2~5 men board game. There are 5 kinds of cards in the original version, include role card, counsellor card, war card, tower card, and general card. In the begining, every player decides the role and set the amount of money. Afterwards, a player can take tax, draw cards, and set a war to attack another tower. At last, whoever survives in the war wins the game. In order to make this game more interesting on the computer, we then change some elements after several disscussions. Below are the guides and rules of our games.

1.To open a game
- a.Press "File">>"OpenGame">>set the number of player.
- b.Decide the Name and the Role for a player>>press "OK">>"start".

2.When a game is started
- a.Each player will be given $100, 50 soldiers, and a tower with 100 bloods.

3.In each round, a player can
- a.Press "Action" and decide to go to
  - i.Construction department (Repair tower).
  - ii.Munition factory (Buy soldiers).
  - iii.Bank (Get money).
  - iv.Explore (Try his/her luck to get more soldiers, money, and win rate).
- b.Press "Shop" to buy treasure.
- c.Press "Deploy" and set offense/defense soldier>>press "Ready" to start the war.

4.At the end of each round, the computer will count
- a. the soldiers and the tower blood a player lost in the war depends on win rate, the soldiers set to defense/offense, and some random mechanism. >>Press "Clear" to start next round.

5.To win the game by one of the condition below
- a.destroying the enemy's tower.
- b.Buyung three kinds of treasure.
- c.Surviving after 10 rounds and get more points than the enemy.(The point depends on a player's money, soldiers, and treasure.

(3) Overall structure of our game

```
|—— Game
        |—— ( Game Main Object )
                |—— Camp
                        |—— Tower
                        |—— Player
                                |—— Role
                                |—— Property

        |—— ( Game Building )
                |—— Functions
                        |—— Bank
                        |—— ConstructionDep
                        |—— Explore
                        |—— MunitionFactory
                        |—— Shop
                                |—— Stuff

        |—— ( Game Operation )
                |—— Commander
                |—— Computer
                |—— EnemyAI

        |—— ( UI Use )
                |—— Create
                |—— MainFrame
                |—— ActionFrame
                |—— EndFrame
                |—— FightFrame
                |—— HistoryFrame
                |—— NewFrame
                |—— NotifyFrame
                |—— OriginalFrame
                        |—— OriginalPane
                        |—— RulePanel
                |—— ShopFrame
                |—— StateControl
                |—— TotalPrint

                |—— ( Animation Use)
                        |—— AnimatedBoat
                                |—— AnimationPanel
                        |—— BankAnimation
                                |—— BankAnimationPanel
                        |—— ClickShake
                        |—— MyApplet
                                |—— TimerAnimation
                        |—— Toaster

        |—— ( Demo Use )
                |—— Constants
```

(4) Java file we design and their function

Game.java

This is the main class that mainly manipulates the game. There are two forms of the game. One is conducting on the terminal which is a testing version, while the other is UI based which contains our main works. **We put more emphasis on the UI version in this game and report**.
As the game starts, it initializes fundamental elements, and then starts the game. In each round, it first asks two teams' player to take some action, respectively. And then two teams engage in war based on the players' setting in the round. After war, judges if there's a team that achieves the winning conditions by the rule we described in "Game Overview and rule". If "Yes", ends the game and prints the winner team. Else, just go to next round.

Camp.java

This class is a super class which is extended by Player.java and Tower.java. In this class, we are able to define names of camps.

Player.java Property.java Role.java

These three classes determine a player's name, soldier amount, money, role...etc. When a game is begun, those properties will be set in these classed, and the role property will slightly influence the number of soldiers, tower blood, and money for a player.

Tower.java

This class extends from Camp which indicates one camp have one tower. Tower has its own resistance, an index of offense solider and and index of defense soldier. With these elements, Computer.java could calculate the result of the war and send them back to each camp's tower.

Bank.java ConstructionDep.java MunitionsFactory.java Explore.java

These four classes provide players in the game four different "Actions" to take. In each class, there are some methods to complete each different action. i.e. Bank.java provides a method for gaining money. ConstructionDep.java contains a method for repairing tower. And, in Explore.java we design random events to provide players chances to win big rewards.

Commander.java

In each rounds, it asks player to allocate the defensive and offensive soldiers they want for the game use in the terminal version. In UI version, it receives the information of FightFrame.java and distributes army.

Computer.java

This class compute the remaining soldiers and tower blood for player at the end of a round. The computation depends on the number of offense/defense soldiers a player, and the win normalized win rate (Includes some random mechanism).

EnemyAI.java

This is a class of an simple AI that can compete with a real player. Our AI would distribute his money wisely depends on how much money he has. In order to fight against with a real player, the AI would equally deploy his army on the offense side and defense side. On the other hand, with this AI, we could test the stability of our game effectively.

ActivityMenu.java

This class is for the original use of the terminal test version. This class helps player to choose which Actions he wants to take.

Functions.java
An abstract class which contains nothing. Maybe for future works.

TotalPrint.java
This class is the speaker of our program. It is responsible for almost all output indications and it also determines which text area to show these informations according to which one’s turn.

*Frame.java

NewFrame.java
This frame provides a friendly condition for player to enter his/her name, make decision with the player number, and which role to play as.

MainFrame.java
This frame is core of this game. It is set with several button such as “Action”, “Deploy”, “Shop”, and “WAR”. Also, there is an area shows each player’s status below. The history bar and file bar are at the top of the frame. Whenever a button is clicked, StateControl will judge and make the game proceeds.

ActionFrame.java
This frame contains some combobox for players to choose different action type to take. After choosing, there are still some options for players to decide which level of services is suitable. (ie. when the player chooses “go to MunitionFactory” then the player needs to decide how many soldiers does he want and pay the price accordingly) If you are not affordable of the action you pick up, you just would be blocked and have to choose another time for not being broken. After you are totally done, this frame would transmit the information to the right methods of classes and deliver what you want to do. (ie. if you choose repair the tower about 20 blood for $40, you would be charged and also your tower would be repaired.)

FightFrame.java
This frame is for player to deploy their soldiers. Players can determine how many soldiers are responsible for defense and how many soldiers are going to attack. If your input number is not correct(i.e. negative numbers or the combination of offense and defense warriors exceeds the amount you have), the frame would not let you march your army! After a corrective input is done, this frame would call Commander.java to help player to arrange the army.

Shop.java Stuff.java ShopFrame.java
These classes is built for creating a shop in our game. Player could buy treasures in shop and Shop.java contains some stuffs in it which are objects from Stuff.java. Like other frames, ShopFrame would block you down if you don’t have enough money but still choosing something.

StateControl.java
This class is the brain of the program. All most 90% of the button will be judge what to do next and what procedures will it take when pressed.

Toaster.java
We know that a toast is a view containing a quick little message for the user in Android . We rewrite it in java with JWindow. We can insert words and pictures in JWindow and set the position of the Toast.

MyApplet.java
In this class we handle the positions of each object showing in the frame.We use a timer and once it is triggered , the characters will start to attack the towel of the opponent .

Constants.java
In this class, we are able to tune the parameters in the game. It provides us to adjust the balancing of the game quickly and easily.

ClickShake.java
This class will be used when a player press “War” at the end of a round. After clicked, this class will first let the left hand side role picture move to the right, wait for 4 sec (Thread.sleep), and let the right hand side role picture move to the left.

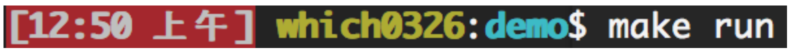
Create.java
When a game is begun, this class gets the decision from a player of choosing a role, and set for a player.

EndFrame.java
This frame will pop up when the game is over. Based on different ending situations, the text area would show different informations about the game results and players could understand who is the winner of this game.

HistoryFrame.java
This class would record what players do and the game results in the history, so player could look back and see what’s the key point and also make a decision according to the past.

(5) How to play our game

step1 : In terminal, type $make run (Should be under the environment with supporting Java Awt and Swing)



```
[12:50 上午] which0326:demo$ make run
```

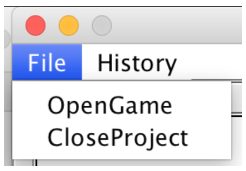
step2 : After step1, an animation will show. After that, it will jump to rule frame automatically. Read rule on the frame carefully and **press OK**.

step3 : After step2, it will close rule frame and show another main frame. **Hit File -> OpenGame** in the left top side of the frame. It will jump out a frame to set a new game.
***** One can press QuickStart button on the bottom of the frame for 2 player's default setting without doing step3.**

(1) Select the number of players

Choose 1 : Play with AI

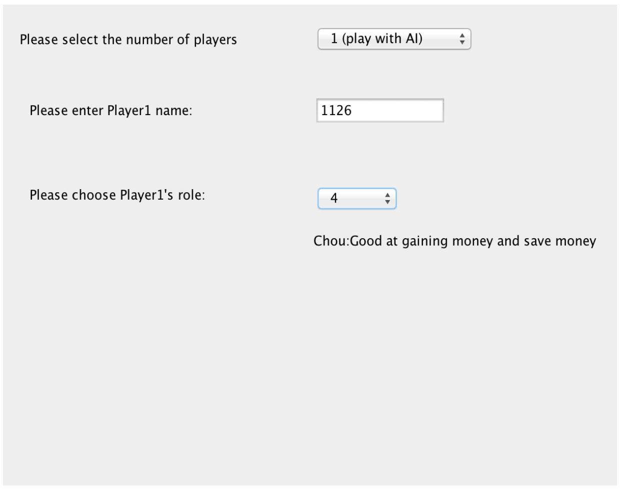
Choose 2 : Play with other people



(2) Type player name
(3) Choose role

0 for Liu, who is good at tower repair
1 for Sun, who is good at gaining money
2 for Cao, who is good at building army
3 for Zhao, who is good at repair and building army
4 for Chou, who is good at gaining and saving money
5 for Chang, who is good at building army and gaining money

After finishing all these settings like the graph below, **press OK**

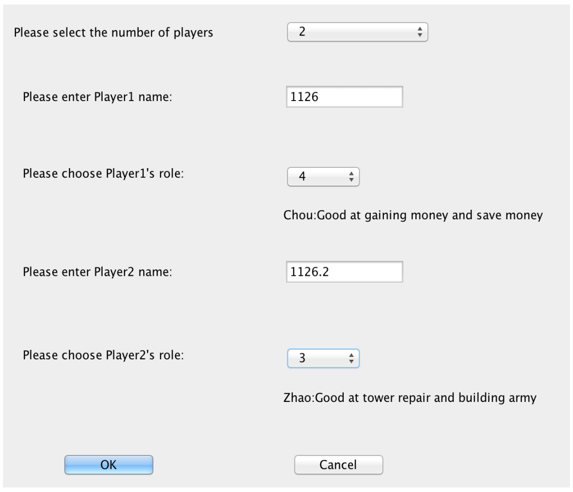


Please select the number of players 1 (play with AI)

Please enter Player1 name: 1126

Please choose Player1's role: 4

Chou:Good at gaining money and save money



Please select the number of players 2

Please enter Player1 name: 1126

Please choose Player1's role: 4

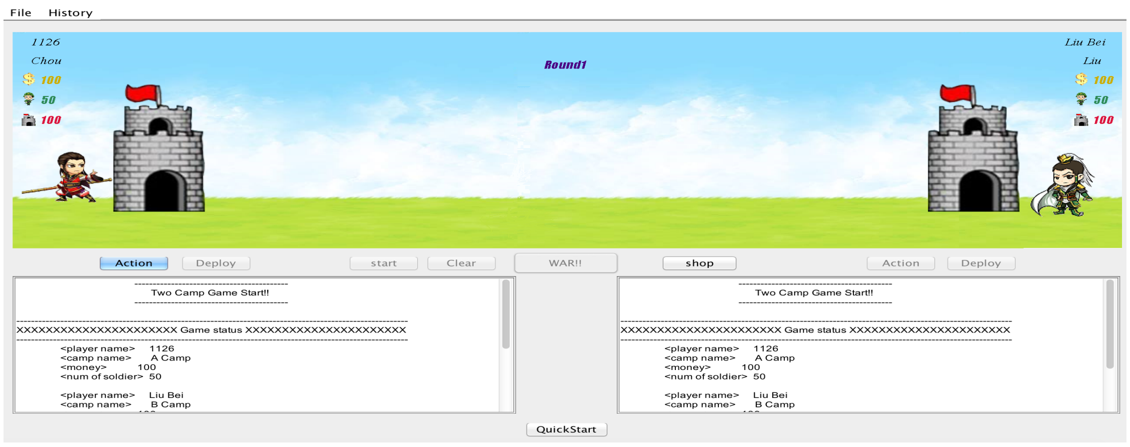
Chou:Good at gaining money and save money

Please enter Player2 name: 1126.2

Please choose Player2's role: 3

Zhao:Good at tower repair and building army

step4 : **Press Start button**. It will show some information and role picture in the top panel of the frame like the graph below.



```
---------------------------------------------
        Two Camp Game Start!!
---------------------------------------------

------------------------------------------------------------------------------------------------
XXXXXXXXXXXXXXXXXXXXXX Game status XXXXXXXXXXXXXXXXXXXXXX
------------------------------------------------------------------------------------------------
        <player name>     1126
        <camp name>       A Camp
        <money>           100
        <num of soldier>  50

        <player name>     Liu Bei
        <camp name>       B Camp
```

```
---------------------------------------------
        Two Camp Game Start!!
---------------------------------------------

------------------------------------------------------------------------------------------------
XXXXXXXXXXXXXXXXXXXXXX Game status XXXXXXXXXXXXXXXXXXXXXX
------------------------------------------------------------------------------------------------
        <player name>     1126
        <camp name>       A Camp
        <money>           100
        <num of soldier>  50

        <player name>     Liu Bei
        <camp name>       B Camp
```

step5 : In each round, player 1 first take at least 2 action and 1 optional action.

(1) **Press Action Button**, it will show Action Frame. Choose the action you want and after that, press OK.

* This action includes

Construction Department : Tower repair, add blood of tower

Go To Munition Factory : Add number of soldiers.

Go To Bank : Gain money

Go To Explore : which have another 3 choice

a. **expeditionary** : have possibility to get more soldiers.

b. **steal money** : have possibility to get much money.

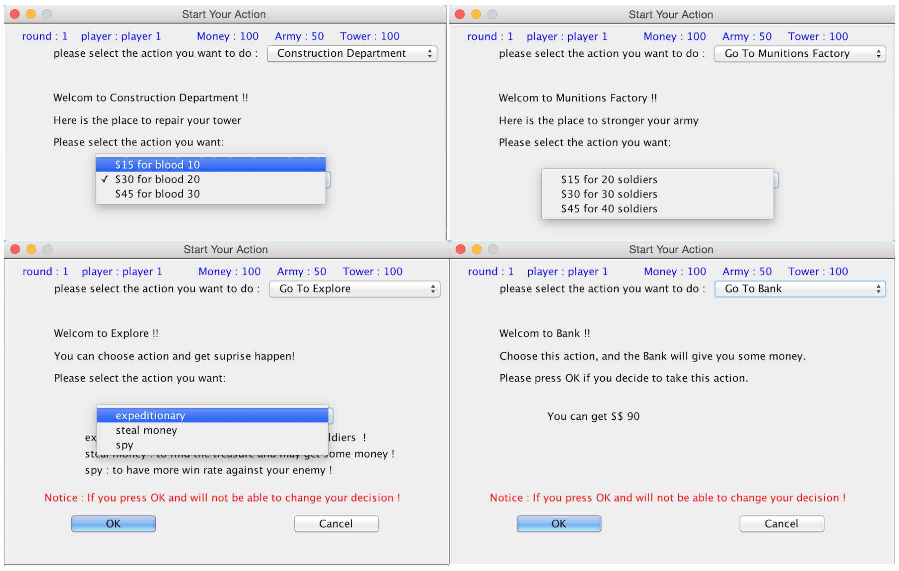
c. **spy** : have possibility to get more win rate over opponent

Abandon this round : don't give up your life

* Each role has some advantage over some choice.

* After pressing OK, the choice can't be changed.

* Example below.



Start Your Action

round : 1 player : player 1 Money : 100 Army : 50 Tower : 100

please select the action you want to do : Construction Department

Welcom to Construction Department !!

Here is the place to repair your tower

Please select the action you want:

$15 for blood 10
✓ $30 for blood 20
$45 for blood 30

Start Your Action

round : 1 player : player 1 Money : 100 Army : 50 Tower : 100

please select the action you want to do : Go To Munitions Factory

Welcom to Munitions Factory !!

Here is the place to stronger your army

Please select the action you want:

$15 for 20 soldiers
$30 for 30 soldiers
$45 for 40 soldiers

Start Your Action

round : 1 player : player 1 Money : 100 Army : 50 Tower : 100

please select the action you want to do : Go To Explore

Welcom to Explore !!

You can choose action and get suprise happen!

Please select the action you want:

expeditionary
steal money
spy

ex ldiers !
st ome money !
spy : to have more win rate against your enemy !

Notice : If you press OK and will not be able to change your decision !

Start Your Action

round : 1 player : player 1 Money : 100 Army : 50 Tower : 100

please select the action you want to do : Go To Bank

Welcom to Bank !!

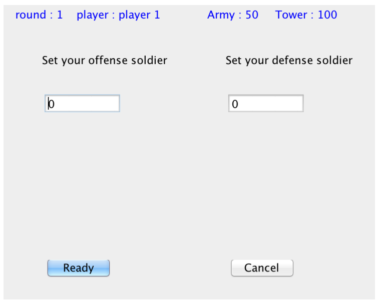
Choose this action, and the Bank will give you some money.

Please press OK if you decide to take this action.

You can get $$ 90

Notice : If you press OK and will not be able to change your decision !

(2) **Press Deploy Button**, to deploy both number of offensive army and number of defensive army. Deploy your army smartly will gain more win rate.

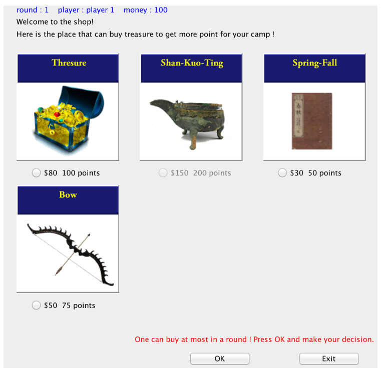


round : 1 player : player 1 Army : 50 Tower : 100

Set your offense soldier: 0

Set your defense soldier: 0

deploy frame



round : 1 player : player 1 money : 100

Welcome to the shop!

Here is the place that can buy treasure to get more point for your camp !

Thresure: $80 100 points

Shan-Kuo-Ting: $150 200 points

Spring-Fall: $30 50 points

Bow: $50 75 points

One can buy at most in a round ! Press OK and make your decision.

shop frame

(option choice) **press Shop button**, you can buy at most one thing here to get more points in the end.

step6 : Switch to player2 to do the same operation.

step7 : **Press war button**, run animation and count the win rate, remaining army, tower of the blood according to the settings of the deployment. It will show up the result on the panel.

step8 : After checking the war information, **press clear** and go to the next round. This will clean the information of this round.

* one can check his former round information by pressing history -> history(left) or history -> history(right)

step9 : repeat step 5 ~ step 9 until game ends.

(6) Answer the question

a. The relation between the classes that we design

1. StateControl: The class Implements ActionListener of Java Swing objects in UI part, so we can handle different states of the game easily in this class. For example, we won't let the

player do two actions in a single round. So we disable the Jbutton “Action” after the player makes an action .

2. Toaster: It is an ancestor of other classes. Other classes can inherit it and use the feature of toaster.
3. MyApplet: It is a class for animation of the characters. Each object of the character has to use this class to do attack animation.
4. MainFrame: In our program , we will create many Jframes due to the action one player performs , all new frames are newed in MainFrame.

b. The advantage of our design

1. Easy to maintain and add new utility: Each utility has its own class , it is easy to find the caller and callee and modify.
2. Good state control: We disable buttons to lower the percentage of making errors .
3. Good communication of player and game : We have a space for output statistics and data. Player will know what to do next by looking for hints in the output area.
4. History part: We will memorize all decisions by player. One can see the data when finishing the game.

c. The disadvantage of our design

1. Not readable enough.
2. Could have better structure of the program
3. We don’t use good algorithm to do computation, so the cost of each computation is very high.
4. Players can see the stats of the opponent during the game .

d. Other packages that we you have used