G52GRP Interim Report

Draft

Group:

Yu Qu (6513261)

Han Bao (6514879)

Yuan Feng (6514885)

Guanqiao Hui (6514885)

Qiwei Sun (6514901)

Yuan Meng (6515015)

December 14, 2015

Word count: 3904

Catelogue

Chapter 1 Background information

1.1 Project introduction

The software engineer group project, Advanced Xiangqi, is to develop an application (app) to help Xiangqi players playing Xiangqi games with people in real world. The app is used to analyze the game and give recommendations. The player controls the Advanced Xiangqi system. The basic rules of Chinese Xiangqi are similar to international chess in some ways. But there are some differences in analyzing pattern comparing with international chess. In general, Xiangqi players, both experienced and in-experienced players would consider the whole situation across the chessboard, and how many steps players could predict depends on playing skills. Therefore, computer, and increase the probability of winning game through people and computer working together could calculate some faults, which may be ignored by people.

The one-year project was divided into four steps. Firstly, complete simple Advanced Xiangqi text-based game using java. Secondly, finish desktop game with chess engine and user interface design. Thirdly, mainly explanation of app design. Forth, optimize and debugging.

The offline app program mainly covers two parts that are interface and engine. The UI interface, chess board with chess pieces, would make sure the user could move pieces and use the four buttons which have searching functions straightly. There are four orders, show, check, search, undo, are designed into buttons in a lower place of the interface. Each function of orders is connected with engine. Then, the functions of the four buttons would be instructed. Firstly, button ‘show’ is for showing the movements of

one possible situation in three steps. For example, user chooses one piece of chess and click button ‘show’, and the computer will calculate the best way of this piece in the situation at start. Then, a predicted opponent’s move would be calculated. At last, the move, which could be used to against the predicted opponent’s move, would be calculated. Secondly, button ‘check’ is for checking 3 best moves of this piece. For example, user chooses one piece of chess and click button ‘check’, and the computer will calculate three best moving way of this piece in the situation. Thirdly, button ‘search’ is for searching the best movement across the chessboard. That means the user do not need to select one certain piece and computer would calculate every possible moves of each piece, and select the best move. Forth, button ‘undo’ is for canceling. Due to the fact that user need to simulate in the app interface and decide which move to be accepted, the undo function is of significant important in re-checking.

1.2 Xiangqi

Xiangqi, Chinese chess, is a sport. Xiangqi become a extremely popular chess activity because it is easy to learn and has high interest. It is one of 78 sport events that are launched in the Spring and Autumn period (770 years BC - 221 BC) in China.

In the ancient China, Xiangqi was regarded as the activity by Scholar-officials to cultivate moral character. And now, it is viewed as a beneficial activity, with millions of amateurs.

Two people play Xiangqi, and each person moves one piece at a time, the playing thoughts are followed by the warfare thoughts. According to the Master Sun (515 years BC), ‘those who render others' armies helpless without fighting are the best of all’, and players kill the opponent’s general in

order to win. The red player moves first, and each player moves one piece at a time, until the result of the game that is either a win, loss, or draw. In the game, people could exercise thinking abilities through several complex relationships which are attacked and defenced, real and imaginary, whole and parts (Sun, 515 years BC).

In conclusion, Xiangqi is simulating ancient war, straight-line war, and land war. Xiangqi, with centuries-old history, needs appropriate playing skills. To clarify, the clear basic rules and elegant various changes make the game could be suitable for most of people all over the world. And all words on both chessboard and chess pieces show Chinese culture.

1.3 Research about Xiangqi engine

According to Chinese Xiangqi annual 2004-2009 (2010, cited in research group of Xiangqi White Paper, 2014), since Professor Shunqin Xu, known as the father of Xiangqi software, began the first study of artificial intelligence on Xiangqi software in 1985, the software of human-computer game is gradually developed. In 2004, xqbase (2004) formulated four protocols to standard the Xiangqi software, which are Move Presentation, Board Record Rules (Forsyth-Edwards Notation), Chess Book Rules (Portable Game Notation) and Universal Chinese Chess Protocol (UCCI). The details of protocols are available at http://www.xqbase.com/protocol.htm. Then, from 2006, several league matches are holed to compare and enhance the Xiangqi engine, and the champion of National Computer Games Tournament in 2015 is Xiang Qi Ming Shou (XQMS). Currently, most formal Xiangqi engines are written in C++ (information from source code of several engines, download address: http://www.xqbase.com/league/enginelist.htm), and the engines written by Java could only find from the individual project, which might be not as good as

famous open source engines and commercial engines, such as Elephant Eye and XQMS. In addition, there is no Advanced Xiangqi engine currently.

In this project, we use an engine written by Acerge (2007). The reason of choosing this engine and how does the engine work is explained in section 4.3.1. To achieve the Advanced Xiangqi, the engine of this project is based on Acerge's engine with some modification. Details will be shown in section 4.3.2.

Chapter 2 Requirement specification

2.1 Scope

This requirement specification identifies the requirements for the Advanced Xiangqi game in desktop version and android application version in the software engineering group project.

2.2 Functional requirements

2.2.1 Updating move

1. The game must allow users to update moves from both the user side and the opponent side through moving the chess on the interface.
2. The game must allow users to undo a move if the piece is moved by mistake.

2.2.2 Prediction and analysis

1. The game must recommend at least three moves based on the move of opponent side.
2. The game must allow users to try the moves except from the ones recommended and predict at least three possible moves of opponent side.
3. The game should analyze the potential to win the game and provide recommendations.
4. The game should analyze the potential dangerous of pieces capture and lose.

5. The game should predict the recommend moves three steps ahead.

2.2.3 Record and save

The game should record every move in the same round of game and save every round of game.

2.3 Non-functional requirements

2.3.1 Interruption

The game must allow users to interrupt the system providing recommendations during the game if users think it is not needed.

2.3.2 Branch factors

The game should select the pick the most possible moves to analyze and provide recommendations.

2.3.3 Highlights

1. The interface must contain highlights for recommendations and allow users to click them to get further information and explanation
2. The interface must contain warnings for the piece would be captured or the potential of losing the game.

2.3.4 Un-lock during using

The device must keep un-lock during the time that user playing the game.

2.3.5 Compatibility

The interface should be used on computers and different android devices (smart phones and tablets) with various sizes and ratio of screens.

Chapter 3 Initial prototype

3.1 Mid- fidelity prototype of the game interface of android system

Please click the link to view the design

https://modao.cc/app/MGr6bGnS6kbjTQMS0vl8

3.2 Description of the design

3.2.1 The welcome page

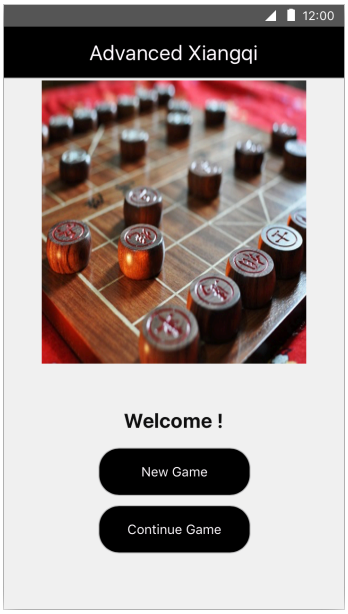


Figure 3.2.1

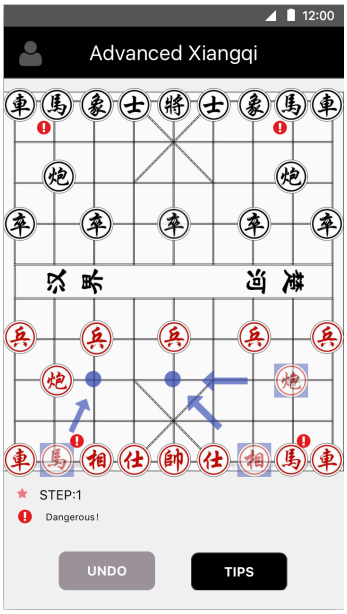


Figure 3.2.2

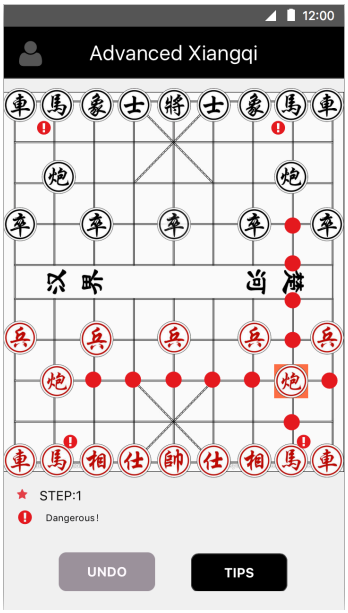


Figure 3.2.3

Figure 3.2.1 is welcome page. It shows two buttons, one of which is to start a new game and the other of which is to continue the game before.

3.2.2 The structure of game interface

In the game interface, there are only two buttons named ‘Undo’ and ‘Tips’ (figure 3.2.2). ‘Undo’ is grey now because the number of step is 1 and it can’t undo. The step counter is displayed at the bottom of chessboard. The star next to the counter is a prompt of whose turn. If it is the red’s turn the star is red and shining. If opponent’s turn it is grey and still. On the chessboard the chessman with a ‘!’ is dangerous that might be captured. ‘Tips’ button includes two parts, ‘check’ and ‘show’. It will be shown in detail later. The interface will always show the searching result of the best choices of movement that user can take.

3.2.3 The movement of chessmen

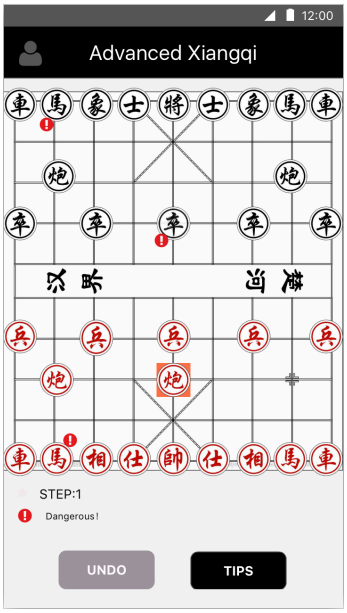


Figure 3.2.4

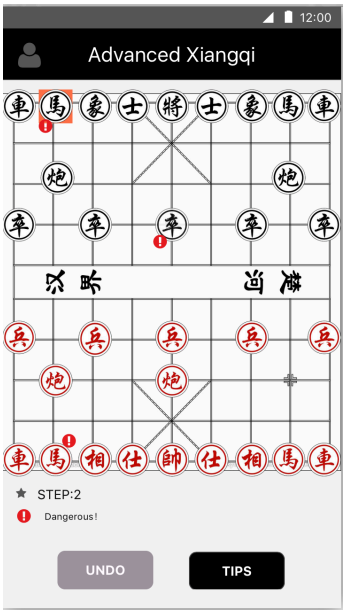


Figure 3.2.5

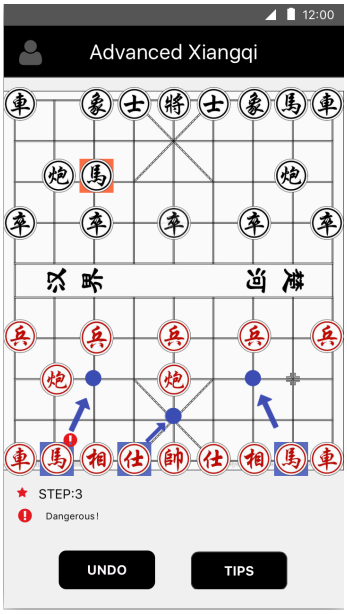


Figure 3.2.6

When pressing the chessman, it will be highlighted and the possible destination will be marked by red point (figure 3.2.3, figure 3.2.4). The

opponent will move directly (figure 3.2.5, figure 3.2.6). If user presses 'Undo' the interface will be figure 3.2.2.

3.2.4 The left menu

Figure 3.2.7

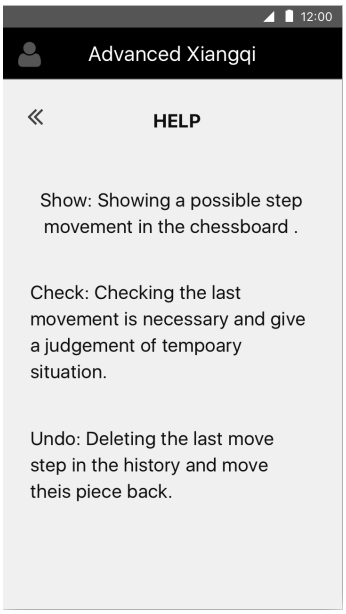


Figure 3.2.8

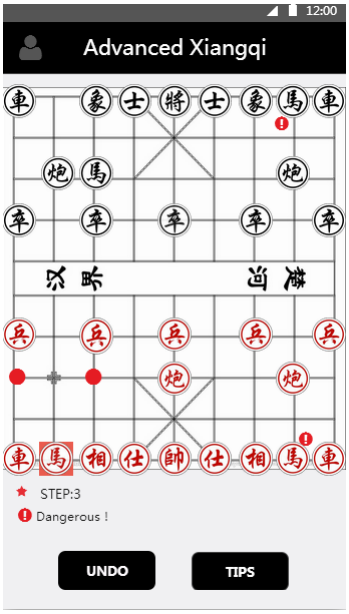


Figure 3.2.9

It is a menu (figure 3.2.7) that allows users to restart (figure 3.2.2) and quit to the welcome page (figure 3.2.1). 'HELP' button is linked to a page showing the explanation of function (figure 3.2.8). Continue game to go back to the situation before (figure 3.2.2).

3.2.5 Check

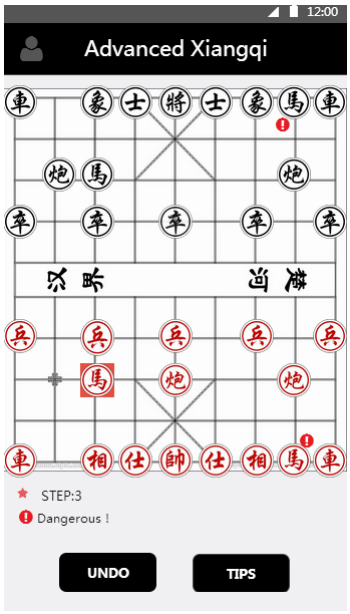


Figure 3.2.10

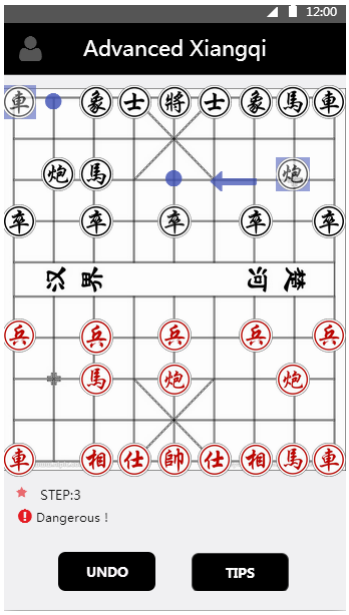


Figure 3.2.11

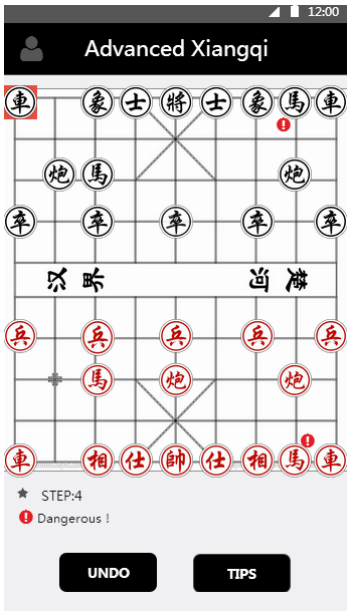
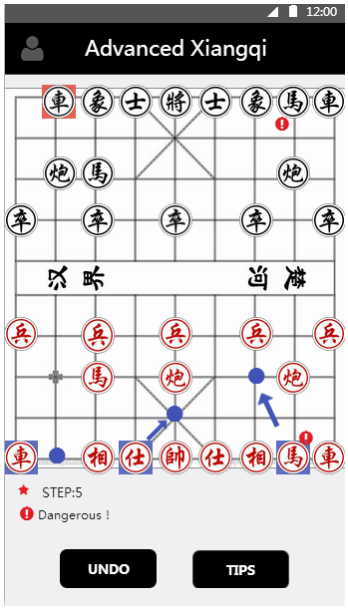
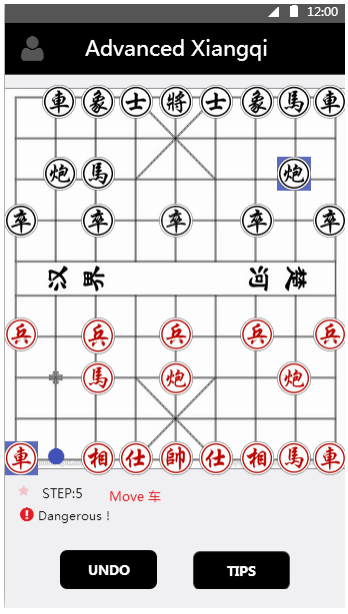


Figure 3.2.12

Press 'TIPS'->'Check' to judge the possible movement of opponent when user finishes moving (figure 3.2.11). It must be after user's movement (figure 3.2.9, figure 3.2.10) and before opponent's movement.

3.2.6 Show





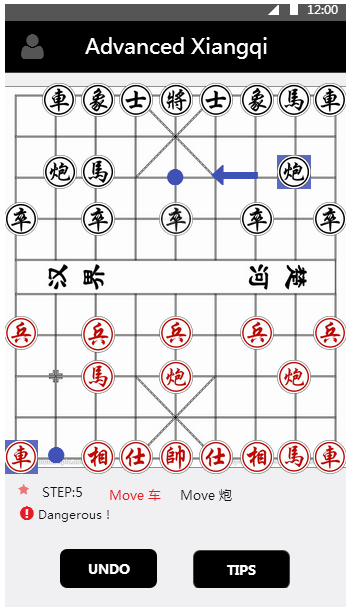


Figure 3.2.13 Figure 3.2.14 Figure 3.2.15

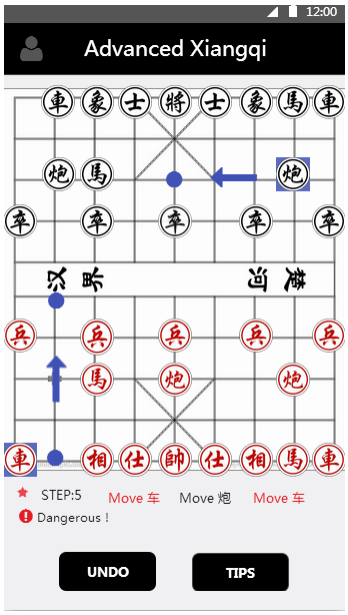


Figure 3.2.16

Press ‘TIPS’->’SHOW’ to view a possible situation of next several steps by highlighting chessmen and written in the space (figure 3.2.14, figure 3.2.15, figure 3.2.16). It can implement ‘show’ not only before but also after you move.

Chapter 4 Design and implementation

4.1 Website function description, content description

The website mainly introduce the structure and rule of advanced Xiangqi game and members of group. It consists of four pages: homepage, introduction of group members, theory of the game and the rule of that as well.

There is a navigation bar and a footer fixed in each page. Besides these, the homepage contains a welcome scene and below that are introductions of group, links to each part of the website and contact information.

Introduction page is divided into two fragments. The first is brief introduction of the usage of this application in Xiangqi game playing. The second part is the detailed introduction of the group members and positions in the project.

In theory page, it introduces the origins of Xiangqi. It displays several assumptions of the origin, which are accepted by public. Additionally, it presents the background, related stories and historical record of these origins. It aims to raise the interests of players to this game.

The rule page primarily introduces the detailed rule of how to move chess and the common way to name these principals of moving chess in China. The icon of corresponding chess is provided beside the description of the rule as well, which make players remember the rule easier.

As for design, all the icons of chess, navigation bar, footer and homepage background are created by Photoshop rather simply adopted from Internet in order to make the appearance of the whole website more unified.

Moreover, except welcome page, footer and navigation bar are written in English only, all the content are written in bilingual form. This can provide better interaction with people both native Chinese and foreigners who are interested in China's Xiangqi culture.

4.2 User interface

In this semester, our group focus much more on desktop version. We use Eclipse as tool and use Java language with limited knowledge and limited time. Swing and Awt packages in Java constitute the whole GUI for desktop part. Desktop version GUI mainly has four parts, menu, chessboard, buttons and history list, as showing in 4.2.1.1.

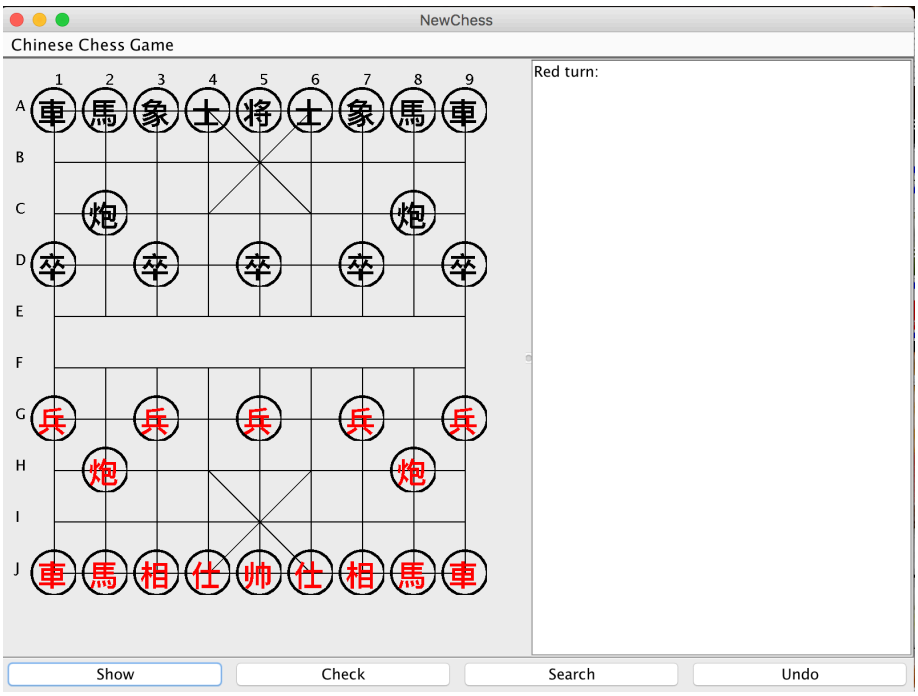


Figure 4.2.1.1

4.2.1 Menu

Only has a restart function.

4.2.2 Chessboard (figure 4.2.1.2)

After a lot of research, most chessboards use numbers or alphabets (Bodlaender, H. and Duniho, F., 2014). Some prefer to lowercase letters and some prefer uppercase letter. Use 'A' to 'J' and '1' to '9' to give a certain and clear point in the board when the history points are listed. A game of Advanced Xiangqi has two sides, one is black and one is red. As the rule shows, red goes first and user could choose any side. If user moves piece invalid or chooses wrong side, there is an alert window and then give a warning on it. When we click one piece, the color will change to blue to tell users that they have got a clicked piece  .

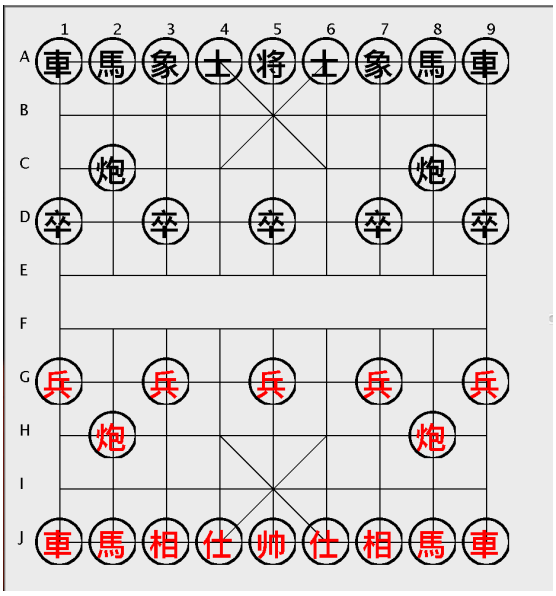
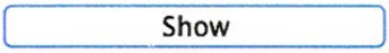


Figure 4.2.1.2

4.2.3 Buttons

There are four functional buttons.



would foresee the next three steps of this game.

Check would check selected piece possibility and give a feedback.

Search would give the next step advice from the temporary chessboard situation.

Undo would delete the last move step in the history and move this piece back.

All results would show in the history board.

History board: Remind users that color turn and show the move step and give some notice. In addition, results of the functional buttons are showing in the history part.

To explain better, we play a game of Advanced Xiangqi on this program.

Example:

1. Suppose that red is our side, showing in figure 4.2.1.3.
2. Move red piece "馬", from "J2" to "H3" showing in figure 4.2.1.4.
3. Move black piece "炮", from "C8" to "C5" showing in figure 4.2.1.5.
4. User does not how to play next, so he clicks "Search", it gives serveral answers(genernal three answers) showing in figure 4.2.1.6.
5. User also wants to guess the following steps, so he clicks "Show", it also gives three steps. Two are red side and one is black side. If there are some dangerous signals such as losing pieces or game, user can see them in advance, showing in 4.2.1.7.
6. According to the "Show" and "Search" result, user wants use none of them. User chooses a piece is Red "炮", click "Check" and get a result of possibilities, showing in 4.2.1.8.
7. Also, user can click "Undo" to cancel this step.

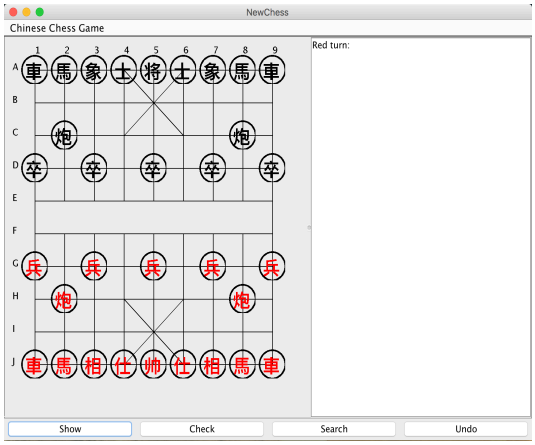


Figure 4.2.1.3

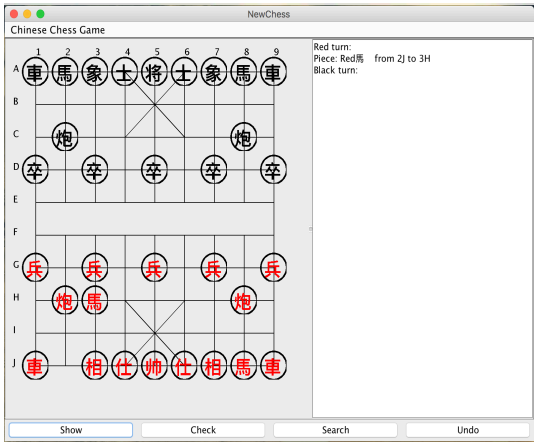


Figure 4.2.1.4

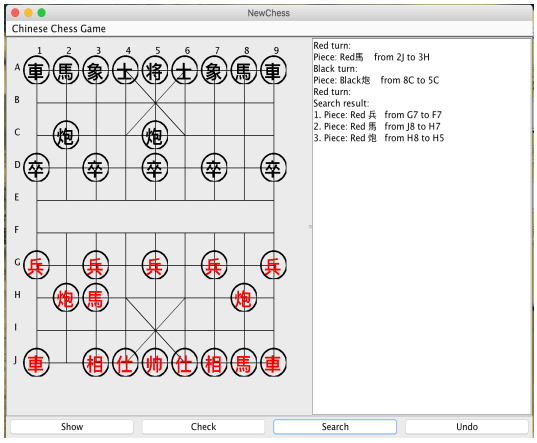


Figure 4.2.1.5

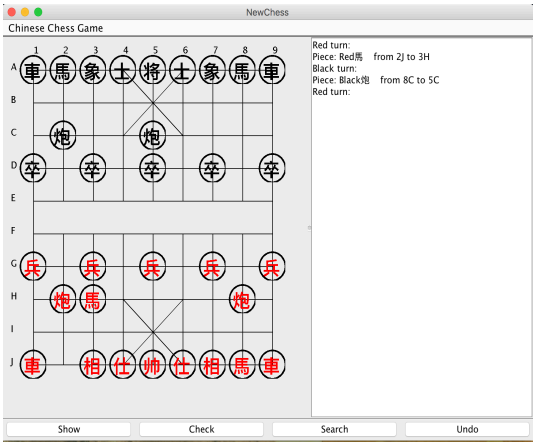


Figure 4.2.1.6

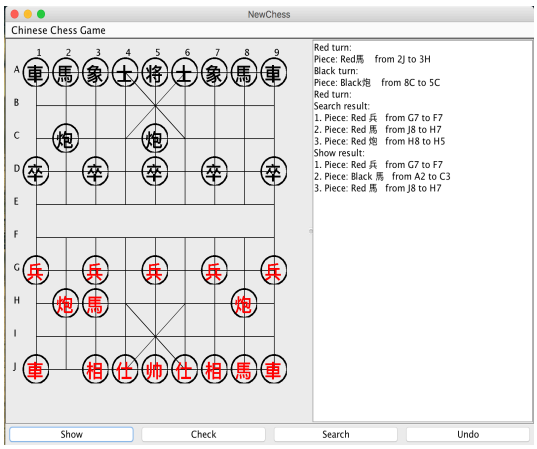


Figure 4.2.1.7

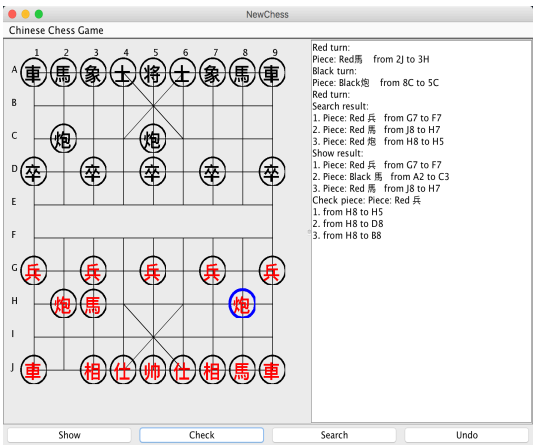


Figure 4.2.1.8

4.3. Xiangqi engine

4.3.1 The description of original engine

The quantity of engines written by Java is limited. The reason of choosing one is that this engine not only use bit board to increase the speed, but also use many methods to enhance the performance of searching. In addition, this engine has achieved almost all the basic functions, and will be briefly explained below.

4.3.1.1 Xiangqi board

This engine is storing the pieces in a bit board. David Eppstein (1997) point out that using this board can enhance the speed of evaluation and move generation.

4.3.1.2 Move generation

One advantage of bit board is only generating the partly movements (James Swafford, cited in xpbase, 2005). However, before every searching, this engine will generate all possible moves to check positions. This detail will be tried to modify in the next several weeks.

4.3.1.3 Rules

This engine has achieved all the rules of Xiangqi, including starting from own side, ending without own side, and special rules of different pieces types.

4.3.1.4 Evaluation

Evaluation is to assess the value of current board. Different types of pieces have different value, and the value will change in the different place. The larger the value is, the better the red side situation is, or vice versa. From large to small, the order of average weights on the different pieces is rook, cannon, knight, advisor, bishop, pawn, and king respectively.

4.3.1.5 Searching

The searching method is the merit of this engine. Most Java engines are only used one or two methods to perform the searching part, while this engine use more than three methods to enhance the performance of creating one best move. Firstly, a hash table is used to record the searched move and to reduce the repeat working. Secondly, as other Xiangqi engine, Alpha-Beta pruning algorithm is the main method to search the best move. Thirdly, this engine has an option to choose whether using Adaptive Null-Move Pruning, invited by Ernst Heinz (1999, cited in xqbase, 2005), to cutting the movement pruning when after this move the evaluation value will decrease. In addition, this engine also uses several methods to improve the performance of Alpha-Beta pruning algorithm, such as Principal Variation Search and Iterative Deepening.

4.3.1.6 Opening book and endgame databases

This engine uses an opening book in Forsyth-Edwards Notation to help increase the performance in the beginning while does not have an end game database. The end game database will be tried to add one month later.

4.3.2 The modifications on Advanced Xiangqi

To achieve the Advanced Xiangqi, two functions are needed: giving different tips on current board and checking a special move, while these two functions can be grouped together. Checking a special move can be seen as updating the board with this move, searching and then undoing the updated move. Therefore, we made two modifications made on the previous engine. Firstly, because the Acerge's engine can only provide one best move, the focus on modifying engine is to find a way to display the different moves. The current method to solve this problem is changing the depth of searching function, and separating the situations based on whether captures a piece. Then the output should only show the different hints to users. However, in the middle game, the engine still can only give one or two possible moves. The next step of modifying engine is trying to find the reason of this problem and figure it out. Possible methods to solve this problem are adding a breadth first searching method rather than just using depth first method, and adding a new evaluation function rather than just using one evaluated functions. Secondly, to display the move path, the searching function should provide a list of move nodes rather than a single move node. At the beginning, to adding this function, the first attempt is using search method again after one search. The extremely slow speed is the reason of abandoned this method. Currently, the newest attempt is adding a list in the searching class to record the searching path, and return it to display.

Chapter 5 Project management

5.1 Programming language

The first key decision of the project is that the programing language of the software development is Java language. The reason for making this decision is that Java language is one of the most compatible programming languages that means that Java language could run in most of the developing environment including the android application development. In addition, Java is also the programming language that all the group members most familiar with since there were a lot of practices relevant with coding design, testing and debugging in the core module in qualified year.

5.2 Development process

The group decided to develop the android advanced Xiangqi application in the five stages. The first stage is research for the chess engine and user interface development knowledge. The second stage is requirement specification and agreement on it. The third stage is developing a simple Xiangqi text-based game. The forth stage is developing desktop Xiangqi game which meets the requirements. The fifth stage is developing the android application based on the desktop game in the forth stage.

Firstly, it is extremely vital to found out the resource needed in the project including the knowledge and skills for developing the software. In the case of this project, it is necessary to find a suitable Xiangqi chess engine written in Java language and learn how to use it properly. Besides, the user interface design knowledge is a new field for us to study and apply it into the

project. Secondly, the requirement specification is the core component in software engineering and could be considered as the guidance for the following software development. Thirdly, after the group reached an agreement on the requirement specification, we decided to develop a text-based Xiangqi game in order to enhance the basic cognition of the Xiangqi game. Finally and most importantly, since the group members lack of experience and knowledge to develop an android application, there are many programming skills and unknown risks that need to be studied deeply. Therefore, we decided to develop a desktop version of the game that is easy for testing and debugging, and then we could transfer the desktop version to the android version based on the finished desktop program. Because it is more convenient to use the Advanced Xiangqi game on a mobile device, so we planned to study and achieve more in the project. Therefore, the last stage of the project is to develop an android version of the Advanced Xiangqi game.

Chapter 6 Future work and risks

Though we choose Java + SDK as tools in this project due to java this familiar language, there is still difficult to transform from desktop version to android version. There are several reasons listed:

6.1 Language different

Reason: Desktop only uses java language but it needs add lots of packages and files to be an android app.

Example: "hello, android" in two different situations (figure 6.1.1 and figure 6.1.2)

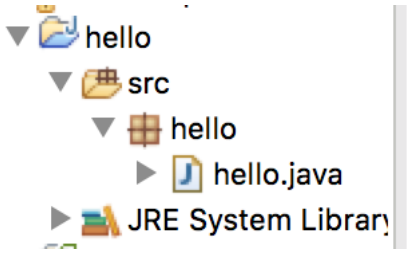


Figure 6.1.1

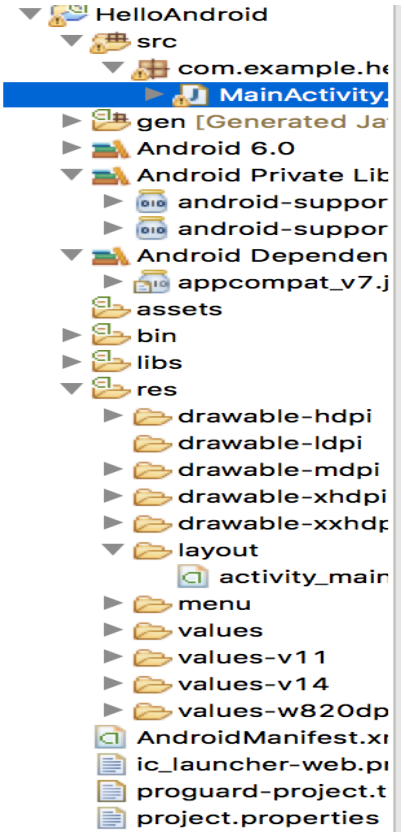


Figure 6.1.2

6.2 Different performance

The performance of desktop edition and android edition has numerous differences.

For example: User clicks mouse on the desktop and then gets the action. But in the phone, it would be replaced by touching. As experienced coder said, there are some obvious problems on exchanging method between "mouseClick" and "touch".

In the "mouseClick", generally it divides into three actions. Mousedown, click and mouseup are to perform functional steps. But for the "touch", this only has touchstart and touchend. According to two methods test (Ye, 2013), the speed of mouseClick is much slower than touch. Even a second delay on the actions and this project has lots of calculation on engine, which already influences user' experience, would decrease enthusiasm to some degree.

6.3 Different Content

Due to the space and functional difference, the content has to be changed. Example: In the desktop version, we could list steps and functional results on the history list. On the phone, the screen does not have so much big space to show this and we would change it as real movements.

Chapter 7 Time plan

Up to now, our group has finished most work as this plan. In the first month, research is the mainly work. Then we embarked on desktop version. During the whole semester, our group had regular formal meetings and weekly informal meetings. We appreciate deeply that the supervisor invests time and effort to our work. To explain detailed, a table could be better to show the circumstance.

Time	Task	Status
2015.9.18	Timeline first made	Finished
2016.9.25	After first meeting with supervisor, timeline should be changed with a disscuss	Finished
2015.9.28	Check the timeline	Finished
2015.10.5	Check the needed equipments	Finished
2015.10.9	Start work as planning -- research	Finished
2015.10.30	Start Desktop version	Finished
2015.11.30	Finish UI and engine	Finished
2015.12.1	Start Interim report	Finished
2015.12.7	Finish Interim report draft	Finished
2015.12.10	Dicuss report with Paul	Finished
2015.12.13	Improve interim report	
2015.12.14	Finalization	
2015.12.18	Hand report in	
2016.1.20	Start android part (learn android language by ourselves)	
2016.2.27	Finish UI part	
2016.3.3	Finish movement	
2016.3.10	Finish engine	
2016.3.17	Finish android part	
2016.3.28	Start testing	

2016.4.20	Start pre and finish GRP final work	
2016.4.29	Final GRP work and report DDL	
2016.5.1	Final presentation practice	
2016.5.4	Open day and presentation	

References (Partly finished)

Acerge (2007) Java象棋，AI引擎 (The Download address of Java Xiangqi, AI engine). Available at: http://download.csdn.net/detail/acerge/253434

Bodlaender, H. and Duniho, F. (2014), "Xiangqi: Chinese Chess" available at: http://www.chessvariants.com/xiangqi.html (accessed December 12, 2015)

David Eppstein (1997) ICS 180A, Spring 1997: Strategy and board game programming. Lecture notes for April 8, 1997. Board representations.Available at:
http://www.ics.uci.edu/~eppstein/180a/970408.html

Heinz EA (1999) Adaptive Null-Move Pruning, ICCA J. 22 (3):123-132.

Research group of Xiangqi White Paper (2014) 《象棋白皮书》关于象棋现状的调查报 1.4.2 象棋软件 (Xiangqi White Paper: the report of Xiangqi status. Chapter 1.4.2 Xiangqi Software). Available at:
http://www.gdchess.com/bbs/UpLoadFile/2014-7/2014073001315249746.pdf

Technical Committee of Computer Games, Chinese Association for Artificial Intelligence (2015) 2015全国计算机博弈大赛冠亚季军获奖明细与统计 (The statistics of computer AI competition in 2015). Available at:
http://www.caaigames.net/info/news150928.asp

Xqbase Web (2005) 引言——论中国象棋信息化的发展 (Introduction: the informatization development of Chinese Chess). Available at: http://www.xqbase.com/protocol/cchess_intro.htm

Xqbase Web (2005) 空着向前裁剪 (Null-Move Forward Pruning). Available at: http://www.xqbase.com/computer/advanced_nullmove.htm

Ye, X (2013), available at: http://www.cnblogs.com/yexiaochai/p/3377900.html (accessed December 12, 2015)