

Coursework Report

HONG HONG

40333300@napier.ac.uk

Edinburgh Napier University - Algorithms and Data Structure (SET09117)

Abstract

Checkers game is a group of strategy board games for two players. It involves diagonal moves pieces and mandatory captures by jumping over opponent pieces. This checkers game used Java language to build and decorate. It is on a 8*8 board and the pieces are on the starting position, and other functions necessary were also set.

Keywords – Checkers, game, Java, NetBeans, AI

1 Introduction

This checkers game is built by Java language. It has the features and functions necessary. It represents the 8*8 game board, players and positions of pieces. It can record play history, and each step that was played can be record and replayed. This game has three different styles and background music, players can choose according to their preferences. These features will be introduced below in details.

2 Panel

The panel of the game is a 8*8 board with pieces on the starting position. This window was name "Checkers" and a navigation bar was created. The navigation bar has two buttons, "levels" and "Options". For the "levels" button, when it is clicked, it will show three selection, players can select easy, medium or hard level depends on their preferences. For the "Options" button, it has five functions. It can record the steps that were played and redo them. The players can click "help" button when they do not know where to place next piece, it will show points on possible positions. It also has a "reset" button which can help players to reset their board to starting position. There were three different styles and background music have been set, players can select anyone they like and turn on/off the background music.

2.1 Backward and Forward

To implement the backward and forward functions, the game should be able to record play history.

```
1 public class Board implements Serializable{
2     private CheckerPosition[] board = new CheckerPosition[32];
3     private MoveList history = new MoveList(); // History of moves.
4     protected Board next = null;
5
6     public Board copy() {
```



Figure 1: **Meun** - Some Screenshots of sourcecode of menu

```
8 Board newBoard = new Board();
9 Coordinate temp = null;
10 newBoard.setHistory(history.copy());
11 for (int i = 1; i < 33; i++) {
12     temp = new Coordinate(i);
13     if (getChecker(temp) != null)
14         newBoard.setChecker(getChecker(temp).copy(), temp);
15 }
16 return newBoard;
```

After implement the memory function, some codes were added to let the current position of piece -1 to implement backward function and then +1 to implement forward function.



Figure 2: **Backward and Forward** - Some Screenshots of these two functions

2.2 Reset and help

To start game from initial state, a menu item-"resetBoard" was added. The initialize() method has been used to initialize the data of board and clear() method has been used to clear history data. The help function used getNext() method to get the next node object, and thus show possible next steps when the players need.

```

JMenuItem help = new JMenuItem("help", null);
help.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("Do help ");

        Board tmpBoard = null;
        Move move = tmpBoard.getHistory().first().getNext();

        pan.bestmovesfromhelp.add(move.getChecker().getPosition().get());
        pan.bestmovesfromhelp.add(move.getDestination().get());
        pan.repaint();
    }
});
options.add(backward);
options.add(forward);
options.add(help);
options.add(resetBoard);
options.add(styleAndMusic);

JMenuItem resetBoard = new JMenuItem("reset board", null);
resetBoard.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("Do resetBoard ");
        pan.pawns.clear();
        pan.board0.initialize();
        boardHistory.clear();
        currentPositionInBoardHistory = 0;
        isBack = false;
        pan.repaint();
    }
});

```

Figure 3: **Reset and help** - Some Screenshots of these two functions

3 Conclusion

To sum up, the game has some advantages. It has complete panel settings. It has background music and different styles of board. And it has implemented some import functions. It can record the play history, and each game that is played can be recorded and replayed. And at the same time, the replaced step can also be undid. In addition, the players can select easy, medium or hard level according to their ability. But the game also has some disadvantages-it is a stand-alone game. That means players can only play this game with computer, it can not be played between two players. To let this game become better, this question should be solved, and the game will be able to be play between two people, a person and computer, or between two computers.