

CS 6613 Project Document

Name: Yang Xu Poly ID: 0447755

1. Objective:

Realize Othello game function, realize all the functions mentioned in the project file, including GUI View and setting different difficulty skill levels.

2. Development Environment:

JavaSE-1.6 + Eclipse

3. Classes and Corresponding functions:

In the implementation I realize 6 classes, their introductions are like below:

(1) The package othelloGame.basicStructure realize the kernel function of the program, it has four classes like below:

A. Board.java

This class describes the condition of the board and its correspond operations:

```
public Board(){}  
// construct function  
public Board(Board otherBoard){}  
// construct from another Board class, deep replication  
private boolean DiscExist(int xAxis, int yAxis){}  
// whether this disc is out of range  
public boolean checkPutDisc(int xAxis, int yAxis, int color){}  
// whether the disc put here would result in the outflanking of the other color discs,  
check every possible direction in the function  
public boolean checkPutDiscForColor(int color){}  
//whether the user using this color disc has valid move  
public void PutDisc(int xAxis, int yAxis, int color){}  
//put disc in that position of the board  
private int getNumberFromColor(int color){}  
//get the number of disc from the color  
public List<DiscAction> actions(int color){}  
// all possible actions for the user using this color disc  
public static boolean terminalTest(Board board){}  
//whether the game ends  
public static int utility(Board board){}  
// the utility of the board  
public static int heuristicUtility(Board board){}  
// the heuristic utility of the board  
private static int heuristicUtilityForColor(Board board, int color){}  
//the heuristic utility for the user using this color disc
```

```

public int getColorFromPoint(int x, int y){}
// what is the color of this position

```

B. DiscAction.java

This class abstract the operation of the disc:

```

public DiscAction(int xAxis, int yAxis, int color){}
// construct function
public int getXAxis(){}
// get x-Axis of the action
public int getYAxis(){}
// get y-Axis of the action
public int getColor(){}
// get color of the action

```

C. AlphaBetaNode.java

This class describes the tree node in the alpha-beta algorithms and the corresponding operations

```

public AlphaBetaNode(Board board, int alpha, int beta, int depth){}
//construct function
public Board getBoard()
//get the corresponding board
public void setAlpha(int alpha)
//update alpha value
public void setBeta(int beta)
//update beta value
public void setChildrenNode(AlphaBetaNode childrenNode, DiscAction
childrenAction){}
//generate child node
public List<AlphaBetaNode> getChildrenBoards()
//get all the children nodes
public List<DiscAction> getChildrenActions()
// get the actions result in the children nodes
public int getAlpha()
//get alpha value of the node
public int getBeta()
//get beta value of the node
public int getDepth()
//get depth of the node

```

D. AlphaBetaTree.java

This class realizes alpha-beta algorithm and cut-off search algorithm

```

public AlphaBetaTree(Board board)
// construct function
public AlphaBetaTree(Board board, int level)

```

```

// construct function according to difficulty level
private void AlphaBetaSearch(Board board)
//realize function ALPHA-BETA-SEARCH(state) function in alpha-beta search
algorithms
private int maxValue(AlphaBetaNode node)
//realize function MAX-VALUE(state,  $\alpha$ ,  $\beta$ ) function in alpha-beta search algorithms
private int minValue(AlphaBetaNode node)
//realize function MIN-VALUE(state,  $\alpha$ ,  $\beta$ ) function in alpha-beta search algorithms
public int getMaximumDepth()
//get the maximum depth of this tree
public int getTimesPruningMax()
// get number of times pruning occurred within the MAX-VALUE function
public int getTimesPruningMin()
// get number of times pruning occurred within the MIN-VALUE function
public int getMaximumValue()
// get maximum node value
public int getNodesNumber()
//get total number of nodes generated
public void printAlphaBetaTreeInfo()
// print alpha-beta tree information
public DiscAction getNextAction()
// the available actions for the root node

```

- (2) The package othelloGame.commandLineMode realize the command line mode of the game, it has one class "CommandLineMode.java":

```

public static void main(String[] args)
// main function, realize the function of command line mode here
public static String getCommandSymbolFromColor(int color)
//get symbol from color, and use it to print the board
public static void commandPrintBoard(Board board)
//print the current board condition

```

- (3) The class "GUIMode.java" realize the GUI mode of the game:

```

public static void main(String[] args){}
// main function
private static String getPackagePath(){ }
// used to find resource file
public GUIMode(){ }
//construct function
public void paint(Graphics g){ }
// update the view
private void processMouseClicked(MouseEvent e){ }
// used to process the event of mouse click, first human move and computer moves by
using alpha-beta search or cut-off search

```

```
private void processClosing(WindowEvent e){}  
// process event of close window
```

4. High Level Description:

Please see the corresponding part in instruction