Othello – Requirements Document

Taylor Spooner

Jordan White

Stephen Robinson

Chandler Underwood

10/3/2013

**Game Rules**

This program is a standard adaptation of the game “Othello” sometimes also called “Reversi”. The rules of the game are summarized below from an online article listed at the end of this document under “Sources”.

The Game

Othello is a two player board game.

The game is played on an 8x8 gridded board of one solid color. (Not checkered like a chess board) The game contains discs colored black on one side and white on the other. One player places discs white side up while the other places discs black side up. The object of the game is to have the most discs of your color on the board when the game ends.

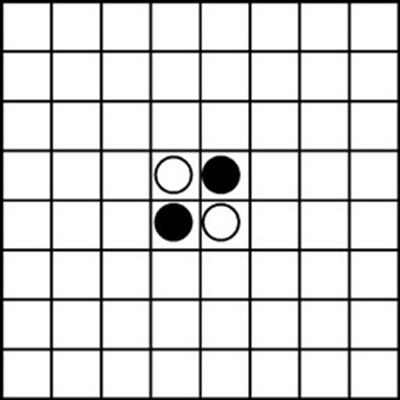
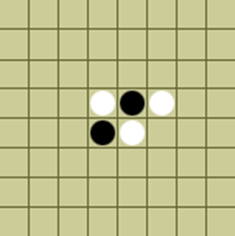
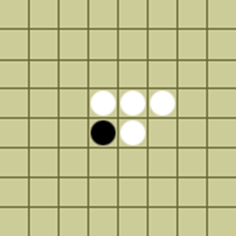
The game starts with 4 discs on the board in the arrangement shown below in Figure 1.

Figure 1: Starting configuration

Game Play

Players take alternating turns placing one disc per turn on the board. The player controlling black discs will always make first move. Players must place disc with their color facing up. A player must place their disc on an empty square that is next to a disc of the opposite color and so that a straight line. can be drawn starting at the new disc, going through one or more oppositely colored discs, and ending on a disc of that player’s color. For the rest of this document we will refer to a legal move as a “flank”.

After a move is made, each of the opposite colored discs that have been flanked will be flipped over to show the opposite color. This is how players will earn points throughout the game. An example of a legal move is shown in Figures 2-4 below.

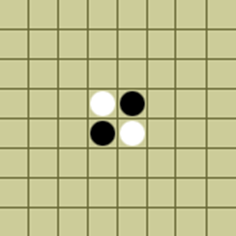


Figure 2: Player’s turn to place a white disc.

Figure 3: A white disc is placed, satisfying a legal move.

Figure 4: The black disc that got flanked is flipped over to white.

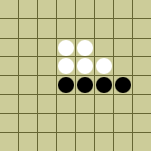
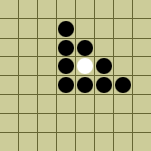
Flanks may be done horizontally, vertically, or diagonally. It is possible to flank in more than one direction, shown in Figure 5. The flipping of discs occurs normally, with all of the flanked discs changing colors.

Figure 5: A black disc is placed that flanks the white discs both vertically and diagonally.

If no legal moves are available to a player, the player must “pass” and give up their turn without placing any discs. If no legal moves are available to either player, the game is over. At this point the player with the most discs on the board is the winner.

Scoring

Scoring in Othello is very straightforward. As mentioned above, the player with the most number of discs on the board at the end of the game wins. Each disc of your color is worth one point. Each time you have discs get flanked, you lose points. Conversely, each time you flank opponent’s discs, you earn points.

**Use Cases (UML)**

See Figure 7 for use case diagram.

* Start Game: Choose disc color and types of players.
* Place Disc: Place one disc, your color up, on the board in a legal position. Flanked discs will automatically flip colors.
* Pass: Press the pass button and end your turn without making a move if no legal moves are available.
* Ask For Hint: Press the hint button if you are unsure of the legal moves available to you. Board will highlight any legal moves.
* Quit Game: Press the quit game to end game early, or to leave the game after it is over.

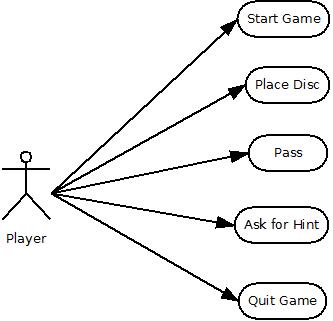


Figure 7: Use Cases

**GUI Windows**

There will be one main GUI screen during the course of the game. All actions will be performed through this GUI. Othello is a perfect information game, so all players will see the same GUI. The GUI window will look similar to the draft in Figure 8 shown below.

The following interactions can occur between the player and the GUI:

* A textbox in the top left of the screen will notify the user of whose turn it is. This textbox will update automatically after each turn is finished.
* There will be two textboxes on the right of the screen to notify the user of the current score. One textbox will be labeled “Black Score” and the other will be labeled “White Score”. These will also update automatically after each turn is completed.
* Pressing the “Hint” button on the left side of the screen will highlight any squares where a legal move can be made by the current player.
* Pressing the “Pass” button will end the current user’s turn. The pass button will only be enabled if there are no available legal moves.
* Pressing the “Quit” button on the bottom of the screen will prompt the user asking if they really want to the quit the game. If the user agrees, the game will automatically be over and exit out.
* A user places a disc onto the board, by touching the square where they want a disc to be placed. If it is a legal square, the disc will appear and any flanked discs will flip over. If the move was illegal, the screen will flash.

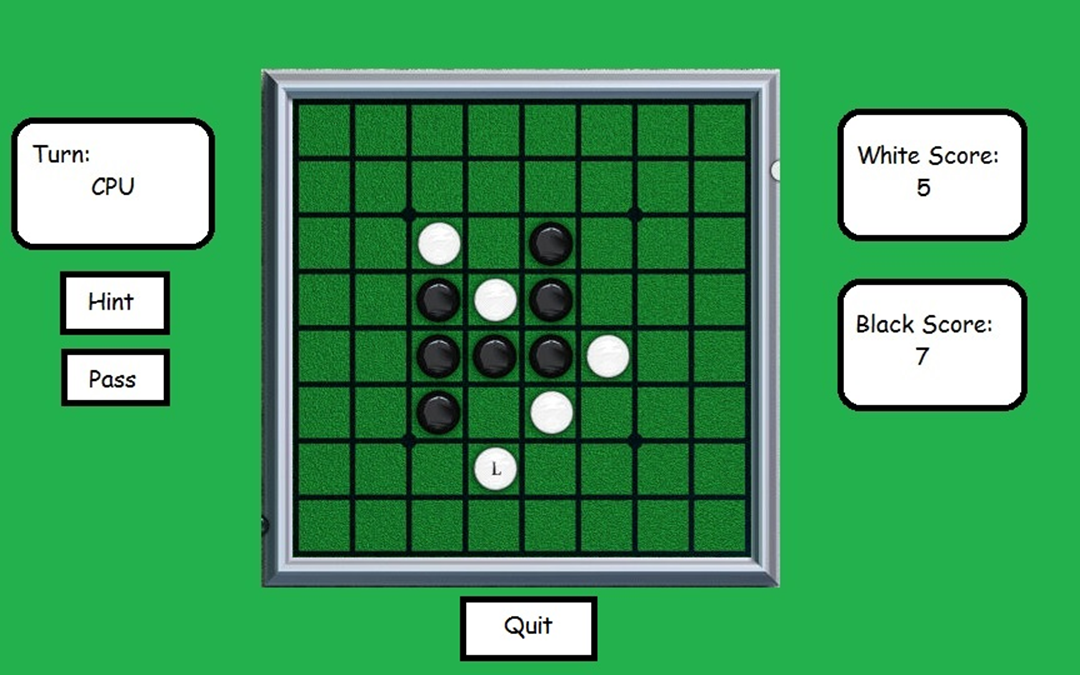


Figure 8: GUI

**Example Run-Through**

Below we will describe one possible play through of the game.

Player’s turn: place disc:

The textbox on the left side of the screen will tell the user it is their turn. The user will examine the board and decide where to place their disc. Once they have decided on a legal square to place their disc, the user will tap the game board at that location.

If the user cannot find a square to place a disc at, they may press the hint button to reveal all legal moves. It is possible that no legal moves are available. In this case, the user would press the “pass” button and their turn would automatically end.

In this case, we will assume the player was able to make a legal move, and tapped the game board at a valid location.

A disc of this player’s color will appear at the tapped location. All opposite colored discs involved in this flank will flip color. The GUI will update the scores on the right side of the screen, and then will update the textbox to tell the user it is no longer their turn.

Opponent’s turn: place disc:

The user will now wait for their opponent to make a move. The only action the user will be able to perform at this point is to quit the game.

Once the opponent has made a move, the GUI will update all information (scores, turn, and discs on the board).

The opponent’s turn is now over, and the player may make a move. These two steps will repeat until the game is over.

Game Over:

Once the game has come to a state where neither player is able to make a legal move, the game is over. At this point the GUI will alert the user that the game is over and tell the user whether they are the winner or the loser.

No more actions are available to the user at this time besides quitting the game.

Quitting the Game:

At any point during the game the user can press the “Quit” button at the bottom of the screen. When this button is pressed, the user will be asked if they are sure they wish to quit the game.

If the user confirms, the game will automatically end and the GUI will shut down.

**Unusual Conditions**

The game will be designed in order to stop players from making any illegal moves. If a player attempts to place a disc while it is not their turn, the screen will flash, and their touch will be discarded. If during their turn a player attempts to place a disc in a non-legal square, the screen will flash and their touch will be discarded. If a player has legal moves available to them, the pass button will be disabled, and touching the button will have no results.

**AI Players**

There will be three levels of AI, ranging from completely random to difficult. The user will be able to select between these using the drop down lists on the game configuration screen. The AIs will function by examining the tree of possible moves and eliminating branches which are improbable to occur or not in the AI’s favor. The AI will take into account advantages such as placing a piece on the edge or corner, and will choose branches which hinder the player making such moves. The hardest AI will look down three moves ahead in the game tree and try to choose the best branch each time. The medium AI will look down one move in the game tree and make a move that will yield it the most points. The easy AI will make a random legal move, regardless of how many points that move will earn it.

The AI players will be able to render the board onscreen if a game is AI versus another AI. Also there will be a delay of one second between moves so that when an AI plays against a person, it will appear to be pondering the players move. Also when an AI plays another AI the delay will be slightly longer so that the observing player can follow the game more easily.

**Project Schedule**

We split the task of creating the game Othello into four subtasks: AI, Game State, GUI, and Player Interface. Stephen Robinson will be in charge of creating the AI, Jordan White will be handling the Game State, Taylor Spooner, with help from Chandler Underwood, will be creating the GUI, and Chandler Underwood, with help from Taylor Spooner, will deal with Player Interface.

A detailed schedule with estimates for hours is outlined below. The time estimates represent hours worked from each teammate.

* 10/06 – 10/12: Complete and turn in requirements document. Begin design specification by completing a UML class diagram that shows relationships between all classes, and the names of any operations for each class. (6 hours)
* 10/13 – 10/19: Fall Break
* 10/20 – 10/26: Complete design specification draft by adding all instance variables to class diagram, adding descriptions for each variable, and descriptions for all methods. We will also create UML sequence diagrams to detail major operations of the program. (9 hours)
* 10/27 – 11/02: Design review in lab October 29th. We will use feedback from the design review to finalize our design specification. Then we will begin implementation of the game. This will mean completing the game state class, the GUI xml layout, and an early version of the human player that cannot make any moves but will be able to start the game and look at the GUI. (12 hours)
* 11/03 – 11/09: Continue implementation. This phase will include completing the local game class, completing the game action classes, and completing a ‘dumb’ AI player. (12 hours)
* 11/10 – 11/16: Complete ‘smart’ AI player. Test game and fix any bugs we are able to find before “Test-to-Succeed” day in class. (12 hours)
* 11/17 – 11/23: “Test-to-Succeed” day in class November 19th. Find and implement solutions to any bugs our fellow classmates found in our program. (6 hours)
* 11/24 – 11/30: Thanksgiving Break
* 12/01 – 12/07: “Test-to-Fail” day in class December 3rd. Fix any last bugs found in the program. Completed project to be turned in Friday, December 6th. (6 hours)

**Concerns**

When we first made a draft of this requirements document, we gave a presentation to other classmates and two professors. They gave us feedback on our requirements, voicing any concerns they had with our proposed ideas. We will address these concerns below.

The first concern raised with our proposal was about the AI. We were asked “How many moves ahead will the AI look? If they look too far, it will take up time and make the program slow.” We have since decided on a set number of moves, 3 or less, for the AI to look ahead. This should keep the program running smoothly.

The second concern was if the AI had two moves of equal value, how would it pick which move to make? We decided that if the AI was faced with two moves that would be virtually equivalent, it would recognize that the two moves were equal and it would simply pick the latest move it evaluated.

Another question raised was “If the best move in the long run of the game is to pass, are you allowed to pass?” The answer to this is no. You may only pass if there are no available moves.

The final concern with our proposal was that you only have to click once on the board, and a move will be made. This could be an issue for a player with large or clumsy fingers who may accidentally place a disc on the wrong square. After talking about this issue we have decided that we don’t want to have multiple presses just to place one disc, as this would make the game more tedious and repetitive to play. To combat any issues of this sort, we have decided that we will make the game board fill up as much of the GUI as possible so that each square will be a good size and accidental presses shouldn’t occur.

Othello – User Manual

(draft)

Taylor Spooner

Jordan White

Stephen Robinson

Chandler Underwood

Starting the Game:

Run the program. The first screen allows user to specify certain game settings. These settings include running the game locally, running the game over network, users name, and type of AI.

If user chooses LOCAL GAME tab from the top of GUI:

* The user will be able to choose both types of players that will play in the game.
  + Options for type of player include:
    - Human Player: runs a graphical game on local machine.
    - Easy Computer Player: an AI player that makes random moves.
    - Difficult Computer Player: AI player that will make moves to earn the most amount of points possible.
    - Network Player: player that will be playing from a different machine
  + Two local human players is not an option.
* There is a text field to give any local players a name
* A button on the bottom of the screen allows you to save current settings as default.
* At the top right of the screen an IP code will be listed if the tablet is connected to the internet. Give this code to any remote players.
* Press the start button game on bottom right-hand corner to start the game once all settings are configured.

If user chooses REMOTE GAME tab from the top of GUI:

* You will be able to select from Human, Easy Computer, and Difficult Computer player types
* You will need to enter the IP code of the host tablet.
* After the host tablet has started the game, press the start game button to connect.

Playing the game:

You should now see the GUI with the board in a starting configuration. On the left side of the screen, a text box will tell you whose turn it is. If it is your turn, you can start the game by making a legal move.

Making a move:

* To make a valid move, tap the game board on a square that does not currently have a disc on it, and will flank the opponent’s discs.
* If you tap a square that is not a valid move, the screen will flash, warning you of this invalid attempt.
* If you are unsure of where to make a valid move, press the hint button on the left side of the GUI to see valid squares.
* If no valid move is available, you will be warned of this, and you must press the pass button on the left side of the screen.

After a move is made:

* The board will update itself, adding the disc you just placed, and flipping the color of any opponent’s discs you may have captured.
* The right side of the GUI will update the score of both players’ scores.
* Your opponent will make a move, all of your buttons will be disabled until their move is over.

Ending the game:

* You can end the game at any time by pressing the quit button found below the game board.
  + The other player will be notified that the game has ended.
* Once there are no more legal moves to be made by either player, both players will be notified that the game is over.
  + The winner will be given a congratulatory message, and the loser will be informed of their loss.