**Project Proposal: Othello**

This project will be a board game called Othello. The purpose of this project is an enjoyable game to play. Othello is a game played on an 8x8 grid, similar to checkers and chess. The object of the game is to get the most pieces on the board your color by the time there are no spaces left on the board. Each player switches off placing his color piece on the board in a location such that there are opposite color pieces encapsulated by two pieces of his color. These pieces are then flipped over to the color of the current player. The game ends when either the entire board is filled with pieces or when a player no longer has any pieces of his color on the board, thus he is unable to encapsulate any of the opponents pieces. A player’s turn is skipped if he has pieces of his color left on the board yet he is unable to play anywhere where it would allow him to obtain pieces of the opposite color.

This will include constant use of Sets due to the fact that each time a player makes a move, multiple sets of opposite color pieces will be made in case those pieces should be changed to the other color. Up to eight stacks could be made on each move in the game. This will challenge me to conserve memory in the computer in order to reduce lag as much as possible.

Classes (UML):

|  |
| --- |
| Piece |
| -Color color  -int row  -int col |
| +Piece(Color c, int row, int col)  +changeColor() : void  +getColor() : Color  +getRow() : int  +getCol() : int |

|  |
| --- |
| Space |
| -Piece p |
| +Space()  +addPiece(Piece p) : void  +getPiece() : Piece  +isFilled() : boolean |

|  |
| --- |
| Player |
| -boolean turn  -int spaces  -Color color |
| +Player(Color c, boolean t)  +getTurn() : boolean  +getColor() : Color  +getSpaces() : int  +setSpaces(int s) : void  +changeTurn() : void |

|  |
| --- |
| Game |
| ? |
| ? |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |