**Programming III – Class Diagrams**

**Josiah Railton – 2.13.23**

|  |
| --- |
| **ReversiBoard** |
| -Player player 1  -Player player 2  -ReversiPiece[][] pieces |
| +ReversiBoard()  Set up players and board  +ReversiBoard(int size)  Set up players and board with given size  -blankBoard()  Create a board, an array of pieces that are blank  -addStart()  Adds the four starting pieces in the middle of the board  +cellIsEmpty(int row, int column) : Boolean  If row and column is blank return true; else return false  +onTheBoard(int row, int column) : Boolean  If row and column valid for the board return true; else return false  -getDirectionVariables(int direction) : int[]  Return an array of two integers based on a given direction  -checkDirectionFlip(int row, int column) : Boolean  Get cell changes for a given direction  Loop through that direction to see if piece is the opposite and on the board  If the next piece is the same type and on the board then return true  +legalMove(int row, int column, ReversiPiece piece) : Boolean  Return false if not on the board  Return false if cell isn’t empty  Loop through each direction to see if there is a legal move.  +makeMove(int row, int column, ReversiPiece piece)  If legal move:  Check each direction and flip any pieces that need to be flipped  +hasMove(ReversiPiece piece) : Boolean  Check each possible location on the board to see if a player has a legal move  +isDone() : Boolean  If hasMove(player 1), and hasMove(player 2) are true return false; else return true  +getWinner() : int  Loops through each cell, and adds to count black and white pieces.  Returns the integer type of the piece |

|  |
| --- |
| **ReversiPiece** |
| +WHITE  +BLACK |
| +ReversiPiece(int type)  Super(type)  +flipType()  Type = getOpposite()  +setType(int type)  If type is between MIN and MAX:  Error of invalid piece  Type = type  +getOpposite() : int  If type is WHITE, return BLACK  If type is BLACK, return WHITE  Return BLANK |

I feel like Game and Player will need a lot of modifications, but this is what I have so far.

Some of the board methods I feel would fit better in the game class, but maybe not.

|  |
| --- |
| **Game** |
| +Player player 1  +Player player 2  +ReversiBoard board |
| +startGame()  Create the board and players and get the game ready  +nextTurn()  Completes a turn and gets ready for the next turn |

|  |
| --- |
| **Player** |
| +String name  +Int currentColor |
| +setName(string Name)  Set name to Name  +getName()  Return name  +setColor(int COLORCONSTANT)  Set currentColor to COLORCONSTANT  +getColor()  Return color |