Programming III – Class Diagrams
Josiah Railton – 2.21.23

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, Player player) : 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 for given player. +makeMove(int row, int column, Player player) 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 player1 +Player player2 +ReversiBoard board +Player turn

+startGame()
  Create the board and players and get the game ready
+changeTurn()
  If turn == player1 set turn = player2;
  If turn == player2 set turn = player1;
+nextTurn()
  changeTurn()
  If turn player doesn't have move
  changeTurn()
  if turn player doesn't have move
  endGame()
  GUI.clearPossibleMoves()
  GUI.displayTurnInfo(turn)
+resetGame()
  Clears the board and any game information
  Set turn player to player1
+endGame()
  GUI.displayWinner(Board.getWinner())
  resetGame();

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

Not sure if buttonClicked is really a function of the GUI class, or just some event function, but it is related.

GUI
+Game game

+buttonClicked(Player player)
  If valid click for given player:
  Changes the button clicked based on the player's color
  Game.nextTurn()
+displayWinner(Player winner)
  Display the given player as the winner
+displayTurnInfo(Player turn)
  If turn == player1
  display message for player 1's turn
  displayPossibleMoves(player1)
  If turn == player2
  display message for player 2's turn
  displayPossibleMoves(player2)
+displayPossibleMoves(Player player)
  For each row
  For each column
  If legalMove(row, column, player)
  Change button to display possible move indicator
+clearPossibleMoves()
  For each row
  For each column
  If cell is empty, then reset button icon to empty, and remove move indicator