**Ultimate Tic Tac Toe Plan**

**Key Variables:**

* String player1Name; (the name of player 1)
* String player2Name;(the name of player 2)
* Int player1Moves;(keeps track of the number of moves player 1 makes)
* Int player2Moves;(keeps track of the number of moves player 2 makes)
* Int array miniBoard0[] = new int (); (stores the information for the top left mini boards)
* Int array miniBoard1[] = new int ();(stores the information for the top middle mini boards)
* Int array miniBoard2[] = new int ();(stores the information for the top right mini boards)
* Int array miniBoard3[] = new int ();(stores the information for the middle left mini boards)
* Int array miniBoard4[] = new int ();(stores the information for the middle middle mini boards)
* Int array miniBoard5[] = new int ();(stores the information for the middle right mini boards)
* Int array miniBoard6[] = new int ();(stores the information for the bottom left mini boards)
* Int array miniBoard7[] = new int ();(stores the information for the bottom middle mini boards)
* Int array miniBoard8[] = new int ();(stores the information for the bottom right mini boards)
* Int array ultimateBoard[] = new int ();(stores the information for the top, middle, and bottom mini boards)
  + For the arrays X will be player 1, O will be player 2 and 0 will be an empty box

**Methods:**

* Main
* Advance Turn
* Check for winner
* Highlight box
* Menu Screen
* Instructions
* Leaderboard
* Game play screen
* Mouse Clicked

**Other Ideas:**

One idea that we plan to implement into the project would be a text file that can store the high scores of previous users and write it to the main to output and update as more games are played.

**Screen Layouts:**

****

****

****

****

****

****

**Pseudocode:**

/\*\*Ultimate Tic Tac Toe

\* James and Jordan

\* December 12, 2016

\* PLAN ONLY

\*/

public class UltimateTTTPlan {

//Declare Key variables and GUI components

//\*\*METHOD\*\* void highlightBox (parameter: index of button selected in mini board from previous turn

// parameter: ultimateBoard array)

{

//DESCRIPTION:

//Determines based on the users last move,

//Which mini board is to be played on by current player

//And set that panel's background green

//Method will be called after every turn to highlight the appropriate miniBoard for the next turn

//Set background of panel\*Y\* to the default background colour

//If user's last click was miniBoard\*Y\* [x] and ultimateBoard [x] > 0 (that miniBoard has been won)

//Set background of all panels to green

//Else

//Set background of panelx to green when user's last click was miniBoard\*Y\* [x]

// x represents the index of the array for the mini board and the panel number of the ultimate board

// \*Y\* represents the mini board that was last played on

//For example: if player 2's selection was miniBoard4 [5], highlight panel5 green

}//End of highlightBox

//\*\*METHOD\*\* int advanceTurn (parameter: the player that took the previous turn

{

//DESCRIPTION:

//Advances the turn to determine which player goes next

//Method will be called after every turn to determine which player is going next

//If previous turn was 1

//Set current turn to 2

//Else

//Set current turn to 1

//return: the current turn number

}//End of advanceTurn

//\*\*METHOD\*\* boolean checkWinner (parameter: array which corresponds to the board being checked. ex

// miniBoard2, or ultimateBoard)

{

//DESCRIPTION:

//Checks to see if a board of 3x3 tic tac toe has been won by a player

//Method is able to check mini boards and ultimate board (the ultimateBoard will still be referred to a

// miniBoard locally in this method)

//\*Y\* represents which miniBoard is being checked, determined by the parameter

//Method will be called after every turn to see if the previous turn resulted in a miniBoard or the ultimate

// board Being won

//Check all possibilities of a winning board

//If miniBoard\*Y\* [0] == miniBoard\*Y\* [1] and miniBoard\*Y\* [0] == miniBoard\*Y\* [2]

//return true

//Else if miniBoard\*Y\* [0] == miniBoard\*Y\* [3] and miniBoard\*Y\* [0] == miniBoard\*Y\* [6]

//return true

//Else if miniBoard\*Y\* [0] == miniBoard\*Y\* [4] and miniBoard\*Y\* [0] == miniBoard\*Y\* [8]

//return true

//Else if miniBoard\*Y\* [3] == miniBoard\*Y\* [4] and miniBoard\*Y\* [3] == miniBoard\*Y\* [5]

//reutn true

//Else if miniBoard\*Y\* [6] == miniBoard\*Y\* [7] and miniBoard\*Y\* [6] == miniBoard\*Y\* [8]

//return true

//Else if miniBoard\*Y\* [1] == miniBoard\*Y\* [4] and miniBoard\*Y\* [1] == miniBoard\*Y\* [7]

//return true

//Else if miniBoard\*Y\* [2] == miniBoard\*Y\* [5] and miniBoard\*Y\* [2] == miniBoard\*Y\* [8]

//return true

//Else if miniBoard\*Y\* [2] == miniBoard\*Y\* [4] and miniBoard\*Y\* [2] == miniBoard\*Y\* [6]

//return true

//Else

//return false

}//End of checkWinner

//Menu Screen (Creates the menu screen that allows the user to select what they would like to do)

//This will created on one large panel that will have it's visibility set to true when user needs to see it

{

//Set JPanels to the JFrame of the menu screen

//Create GUI components like buttons for the user to press to decide where they will place their X or O

//Also a button for the user to use for exiting

//Constructor to set up GUI

//Create a window that has a title called "Ultimate Tic Tac Toe"

//Set the size of the window for the gameplay screen

//Create the layouts for each panel of the JFrame

//Set a layout for each panel of the JFrame

//Create action listeners for each button to see what the user presses and act on it

//Add components to the panel like the buttons

//Add panels to the frame

//Call an image to set as the background

//Output the image

//Catch any exceptions

//Set the frame to visible

//If the user selects the play game button

//Send them to the play game screen

//If the user selects the instruction button

//Send them to the instruction screen

//If the user selects the leaderboard button

//Send them to the leaderboard screen

}//End Menu Screen

//Gameplay Screen (The screen where the user will play the game. Outputs the board the number of moves

//the X and O for the moves users have made)

{

//Set JPanels to the JFrame of the gameplay screen

//Create GUI components like buttons for the user to press to decide where they will place their X or O

//Also a button for the user to use for exiting and a text field for name input

//Constructor to set up GUI

//Create a window that has a title called "Ultimate Tic Tac Toe"

//Set the size of the window for the gameplay screen

//Create the layouts for each panel of the JFrame

//Set a layout for each panel of the JFrame

//Create action listeners for each button to see what the user presses and act on it

//Add components to the panel like the buttons

//Add panels to the frame

//Call an image of the board to output

//Output the image

//Catch any exceptions

//Set the frame to visible

//Output a screen to get the players to input their names

//If the user selects start game then begin the game

//Start game loop

//Call advanceTurn to determine who's turn it is

//Alert users who's turn it is

//Highlight the miniBoard or miniBoards that can be played on for this turn

//Do this by calling highlightBox

//Get user's button click using action listeners

//Only the highlighted miniBoard's buttons will be enabled

//As well, only empty buttons in the miniBoard will be enabled (spaces that are not X or O)

//Set the button clicked to X or O depending on the current turn

//Check if the miniBoard has been won by calling checkWinner

//If the user has won miniBoard

//Highlight the miniBoard either red for player 1 or blue for player 2

//Check if the user has also won the ultimateBoard by calling checkWinner

//If the user has won the ultimateBoard (the game is over)

//Highlight the three squares that they won with their colour

//Output a game over message for the user

//Send the total move that each player made to a text file that can be read from the leaderboard

//Display main menu

//If the ultimate game has not been won

//Output the players name and their number of moves

//Highlight the name of the user who is taking their turn

//If the user selects exit

//Output a game over message

//Return them to the menu screen

//End game loop

//Exit when the game is over of the user selects the exit button

}//end Gameplay Screen

//Instruction screen(If the user selects the instruction screen button then send them to the instruction

//screen where they can read the rules

{

//Set JPanels to the JFrame of the instruction screen

//Create GUI components like buttons for exit

//Constructor to set up GUI

//Create a window that has a title called "Ultimate Tic Tac Toe"

//Set the size of the window for the gameplay screen

//Create the layouts for each panel of the JFrame

//Set a layout for each panel of the JFrame

//Create action listeners for each button to see what the user presses and act on it

//Add components to the panel like the buttons

//Add panels to the frame

//Call an image to set as the background

//Output the image

//Catch any exceptions

//Set the frame to visible

//Output the instructions for the user to read in the main panel in the center

//If the user selects exit

//Return to the menu screen

}//end instruction screen

//Leaderboard screen (If the user selects the leaderboard button then send them to the leaderboard screen

//and read from a text file to output the highest score for each user

{

//Set JPanels to the JFrame of the instruction screen

//Create GUI components like buttons for exit

//Constructor to set up GUI

//Create a window that has a title called "Ultimate Tic Tac Toe"

//Set the size of the window for the gameplay screen

//Create the layouts for each panel of the JFrame

//Set a layout for each panel of the JFrame

//Create action listeners for each button to see what the user presses and act on it

//Add components to the panel like the buttons

//Add panels to the frame

//Call an image to set as the background

//Output the image

//Catch any exceptions

//Set the frame to visible

//Read from the text file and output the high scores from previous games

//Close text file

}//end leaderboard screen

//Main

{

//Create a JFrame

//Close the frame if the user press the x button on the window

}//End main

}//End of class