Design Document
1204 - Team1
TicTacToe Game

**Team Responsibilities:**

Paige Mortensen

- main
- checkValidMove
- playGame
- updateScore

Edward Martines-Anaya

- getMenuChoice
- showBoard
- setResetBoard
- getMoveX / getMoveO

Karla Montano (? - functions will be distributed to other group members if needed)

- showScores
- getPlayers
- win
- tie

Thomas Murray (?)

- n/a

**Timeline:**

- Paige / Edward meet up @ 2 on friday for 2 hrs, may work on it more that day
- Karla completes showScores and getPalyers, Edward setResetBoard, Paige Main
- Meet up on Monday for ~1hr @6 to resolve any issues from the weekend
- Meet Tuesday 3:30 for at least an hour
- Finalize on Wed

**Functions:** **Note we may update parameters as we continue coding

main()
**Functionality:** contains a do while for menu choice. 2 players are initialized and scores are set to 0.
Logic → while game is not won, continue playing
→ if game is won or tied, increment the score of the winning player and update scoreboard .txt file

getMenuChoice()
**Input parameters:** none
**Returned Output:** int
**Functionality:** gets user option and returns an int to the function (to be eval in switch statement)

showScores()
**Input parameters:** file pointer
**Returned Output:** void
**Functionality:** this will play out option 2 of the menu and display the top 10 scores to the screen

playGame()
**Input parameters:** an array of chars, pointer to player 1, pointer to player 2
**Returned Output:** void
**Functionality:** this function is where all game play will happen → calling functions, not saving/handling data

updateScore()
**Input parameters:** file pointer
**Returned Output:** void
**Functionality:** this will update the scores.txt file → access an array of names and an array of scores, iterate through the score array, find where it should be placed, and place the score, and print the new top 10 scores to the file

getPlayers()
**Input parameters:** file pointer
**Returned Output:** void
**Functionality:** this will get the names of the players and display their names

showBoard()
**Input parameters:** array of characters
**Returned Output:** void
**Functionality:** this will display the most current board to the screen, we need to initialize the elements in the array for the board to "empty spaces"

getMoveX()
**Input parameters:** 2 int variables for setBoard
**Returned Output:** void
**Functionality:** prompts user and gets move location for player x

getMoveO()
**Input parameters:** 2 int variables for setBoard
**Returned Output:** void
**Functionality:** prompts user and gets move location for player O

win()
**Input parameters:** array of chars
**Returned Output:** _Bool
**Functionality:** this will check if the game is over by winning (comparing winning character moves)

tie()
**Input parameters:** array of chars
**Returned Output:** _Bool
**Functionality:** this will check if the game is over by tie (if array has elements in every location)

checkValidMove()
**Input parameters:** array of chars
**Returned Output:** _Bool
**Functionality:** this will check if the spot is already "full" or if the spot DNE on the board

setResetBoard()
**Input parameters:** array of chars
**Returned Output:** void
**Functionality:** this will initialize the board to be empty, and clear after a game has ended