Design Document
Erin Keith
TicTacToe Game

Data

Board – 2D array of chars (3 x 3)
Player 1 Name – string
Player 2 Name – string
Player 1 Score – int
Player 2 Score – int
Saved Player Names – array of strings
Saved Scores – array of ints

Game Play

User is given the option to play or view scores. If the user chooses to play, a match (series of games) begins. Each player is prompted for their name, then they take their turns placing their piece. A game is over when one player wins (three in a row) or it's a tie (the board is full). They can continue to play; when they're done their score (how many games they won) is saved to a file. If they choose to view scores, that file is loaded and all names and scores are displayed to the screen.

Functions

main()
Functionality: Create arrays for board, player names, saved names, saved scores. Create variables for user inputs and player scores. Display menu until user chooses to quit. If the user chooses to play the game, begin a match. Clear the board between every game. Keep track of each game's winner and increase that player's score until they choose not to continue playing. Save both scores. If the user chooses to see the scores, display them.

getMenuChoice()
Input Parameters: none
Returned Output: integer user choice
Functionality: Display application menu choices. Return user choice.

beginMatch()
Input Parameters: char array player 1 name, char array player 2 name
Returned Output: none
Functionality: Prompt player 1 for name and save. Prompt player 2 for name and save. Display player's pieces.

clearBoard()

Input Parameters: integer dimension, 2D char board array
Returned Output: none
Functionality: Set all elements of the board to a SPACE.

displayBoard()
Input Parameters: integer dimension, 2D char board array
Returned Output: none
Functionality: Display each piece on the board (blank, X, or O), and the corresponding row and column numbers.

getMove()
Input Parameters: integer dimension, 2D char board array, char piece, integer* row, integer* column
Returned Output: none
Functionality: Continue to prompt the user for a move until they have entered a valid move. Update the board with the player's piece.

validMove()
Input Parameters: integer dimension, 2D char board array, integer row, integer column
Returned Output: bool
Functionality: Verify that the spot on the board is open, and that the row and column values are within bounds of the board.

playGame()
Input Parameters: integer dimension, 2D char board array
Returned Output: char winning player piece
Functionality: Set the first player's piece. Prompt the correct player for their turn. Display the board. Get the user move. Switch player piece. Continue until game is won or a tie. After the game is over, display the correct message. Return the appropriate piece (either the winning player or a null character for a tie).

boardFull()
Input Parameters: integer dimension, 2D char board array
Returned Output: bool
Functionality: Check all elements of the board for a SPACE.

gameWon()
Input Parameters: integer dimension, 2D char board array, move row, move column
Returned Output: bool
Functionality: Start at the latest move on the board (move row and move column). Start counting in each direction. If there are as many or more same

pieces in a row as the dimension, the game is one. Check both diagonals, horizontal, and vertical.

readScores()
Input Parameters: integer scores array, 2D name strings array
Returned Output: integer number of scores
Functionality: Open scores file. Store names and scores into respective arrays. Close file.

writeScores()
Input Parameters: integer number of scores, integer scores array, 2D name strings array
Returned Output: none
Functionality: Open scores file. Write names and scores from respective arrays to file. Close file.

saveScoreScores()
Input Parameters: integer new score, char name array
Returned Output: none
Functionality: Read scores. For each of the scores, if new score is greater, swap scores and names in the arrays. Otherwise if it is the end of the scores or the max, save the score and name in the arrays. If the current number of scores is less than the max, increase the number of scores. Write scores to the file.

stringCopy()
Input Parameters: char source array, char destination array
Returned Output: none
Functionality: Copy each character from the source string to the destination string. Make sure there is a null character at the end of the destination string.