Tic Tac Toe Program

Brian Ramirez

CSC 17C

Professor Lehr

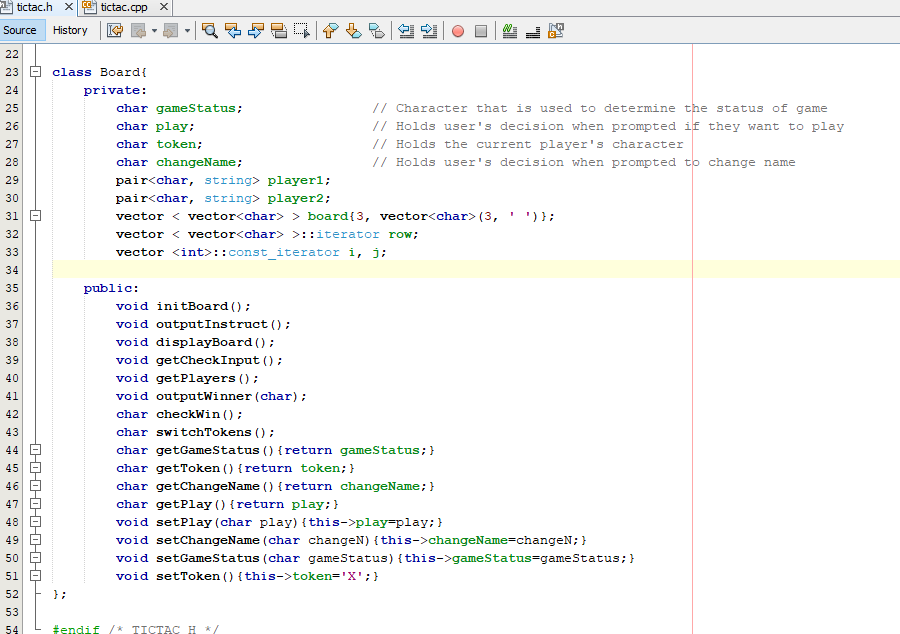
Guthub Link:

Description:

This project is a Tic Tac Toe program, it is a board game in which players take turns to place a character, usually an ‘X’ or an ‘O’, on a 3x3 grid. The program begins by displaying a set of instructions so that players are aware about the rules of the game and how the program will run. Below the instructions, the players are prompted to choose whether they want to start a game. To start a game you either enter ‘y’ or ‘Y’ and the game will begin, however you can also enter ‘n’ or ‘N’ and the program exits. Once the game starts the players are prompted to enter a name. Then the first player will go first and will be ‘X’ by default. The way the player enters their moves is by entering the row number (between one and three), pressing ENTER, then entering the column number (also between one and three). Players keep taking turns until one player gets three characters in a row. Then my member function checkWin() will look whether there is three characters in a row and determine the winner, or if there is no winner then it declares the game as a tie. Finally the players have the option to play again if they do decide to play again then they get prompted with the choice to change their name. If they choose not to start a new game, the program exits.

Folder/File Layout:

My program is split into a header file and two cpp files. The header, tictac.h, contains the Board class for the game object. The tictac.cpp file contains all the method function declarations for the functions inside tictac.h.



Objects:

My program has one object, the Board object. This object includes a six variables, mainly to hold player/game information. For example, board is a 2d vector that is used to hold the information for the 3x3 grid/matrix, player1 is a pair that holds the player’s token and the player’s name, and play holds the user’s decision when prompted if they want to play. There are 16 functions in this object that utilize the private variables. For instance, initBoard() initializes the vector to fill all the spaces on the board with blanks, displayBoard() displays the game board, and switchTokens() switches the token from the current player to the opposite token (to get ready for the next turn).

|  |
| --- |
| Board |
| * gameStatus: char * play: char * token: char * changeName: char * player1: pair<char, string> * player2: pair<char, string> |
| * initBoard() * outputInstruct() * displayBoard() * getCheckInput() * getPlayers() * outputWinner(char) * checkWin() * switchTokens() * getGameStatus() * getToken() * getChangeName() * getPlay() * setPlay(char ) * setChangeName(char ) * setGameStatus(char ) * setToken() |