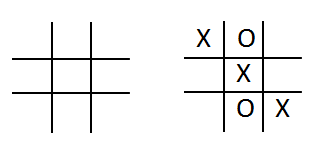
**C/C++ TIC-TAC-TOE LAB REPORT**

**1) Enter your name, student ID, platform (Mac or PC) and date**

Name and Student ID: Samuel Indurkar, 0888068

Class: CIS054 C/C++ Programming

Platform (Mac or PC): gcc and Eclipse on MAC  
Date: 7/7/2017

**DESCRIPTION**Create a program using a two dimension array and functions that lets two players play the tic-tac-toe game. The program is to read a number from 1 to 9 from the keyboard alternating between the X and the O. Ask which square the X player wants, and then ask which square the O player wants. Continue until the game is won by either player, or all the squares have been filled. (Called a Cat's Game or a Tie Game)

A function must be used to display the tic-tac-toe game board, and a function must be used that determines if the game has been won. You can create the program on your own, or start with the sample code provided.

**EXTRA CREDIT** (3 points)Modify the **DisplayTicTacToe()** function using the | and – characters to draw lines for the tic-tac-toe display.

**LAB REPORT  
2) Determine the Inputs, Processing and Outputs before creating the program**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSING** | **OUTPUTS** |
| read a number from 1 to 9 alternating between the X and the O | Verify the number entered is valid.  Verify the location for number entered is actually empty.  Check if we have three-in-a-line winner.  if winner found, announce the winner else  if no more space, announce tie | Announce winner, either X or O; or if no more space, then announce tie. |

**DISCUSSION**

**3) Complete the DISCUSSION section. It does not need to be long, but it needs to be complete.**3a) What did you do to develop the program? ("Followed the Directions" is not a complete description)

Read a number from 1 to 9 from the keyboard alternating between the X and the O; Use rand() to randomly choose either X or O.

Verify the number entered is valid.

Verify the location for number entered is actually empty.

Check if we have three-in-a-line winner.

if winner found then announce the winner else

if no more space, announce tie

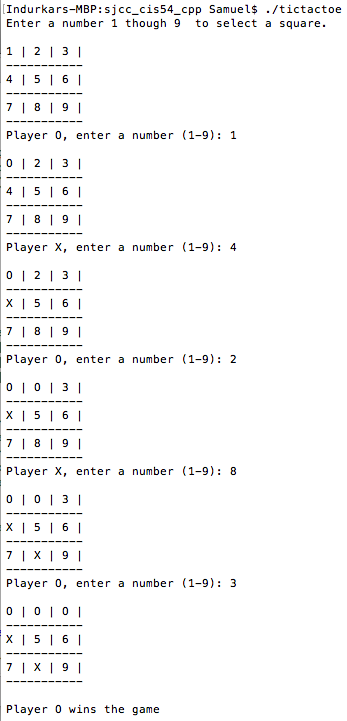
3b) What problems did you have and how did you overcome the problems?

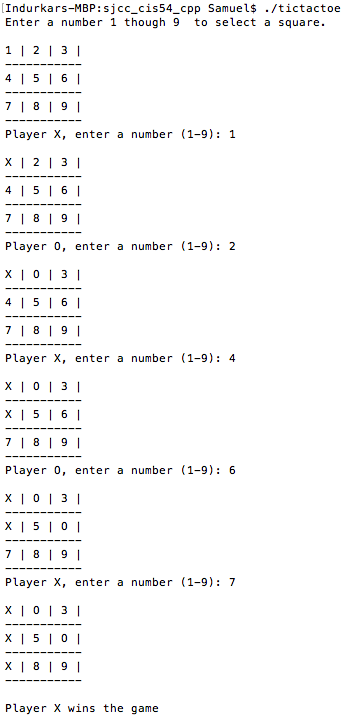
No problem

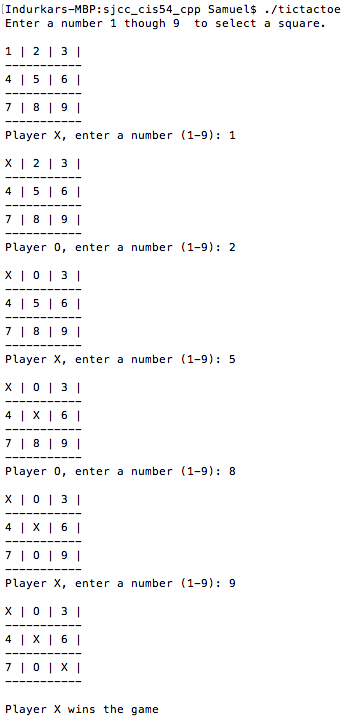
**PROGRAM OUTPUT**

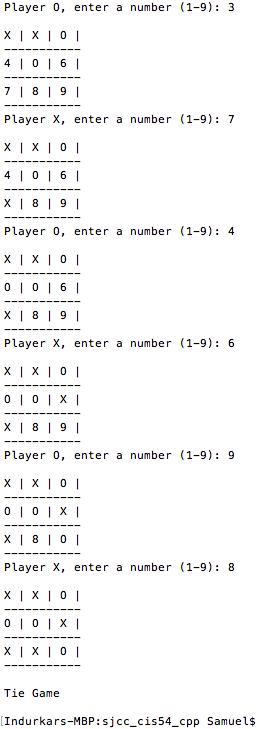
**4) Show screen shots for at least FOUR sample games: 1) a win going across, 2) a win going down, 3) a win going at a diagonal, 4) a tie game in which there was no winner.**

Refer to previous lab assignments for instructions on how to capture a screen or portions of a screen for either the PC or a Mac



****

****

****

**PROGRAM LISTING**

**6) Copy and paste the code that YOU typed to make the program work. Your program should include a comment block at the top that shows the name of the program, date, version and your name.**

/\*

\* tictactoe.cpp

\*

\* Created on: Jul 7, 2017

\* Author: Samuel

\*/

// Version: 1.0

#include <iostream> // needed for cin and cout

#include <cstdlib> // needed by srand() and srand()

#include <ctime> // needed for the time() function

using namespace std;

// function prototypes

void DisplayTicTacToe(char board[3][3]);

char CheckForWinningGame(char board[3][3]);

int main(int argc, char\* argv[])

{

char board[3][3] = { // index for positions in the array

{'1', '2', '3'}, // [0][0] [0][1] [0][2]

{'4', '5', '6'}, // [1][0] [1][1] [1][2]

{'7', '8', '9'} // [2][0] [2][1] [2][2]

};

int squareCounter = 0; // count how many squares are used

char gameWinner = '-'; // winner of the game ('X' or 'O')

char inputCharacter; // input from the user (should be '1' to '9')

int inputInteger; // input converted to integer 0 to 8 (start from 0)

char player; // current player, set to either an 'X' or an 'O'

// select the starting player

srand( (unsigned int)time(NULL)); // seed the random number generator

if ( rand() % 2 )

player = 'X'; // odd player is x

else

player = 'O'; //even player is o

// starting instructions

cout << "Enter a number 1 though 9 to select a square." << endl;

// Ask for a 1-9 until all squares are filled or the game has been won

while (squareCounter < 9 && gameWinner != 'X' && gameWinner != 'O')

{

DisplayTicTacToe(board);

cout << "Player " << player << ", enter a number (1-9): ";

cin >> inputCharacter;

// Arrays start counting from index 0

// convert inputCharacter from ASCII ('1' to '9') to integer (0 to 8)

// to select the row and column index for the 3x3 array

inputInteger = inputCharacter - '1'; // now the inputs are 0,1,2,3,4,5,6,7,8

int row = inputInteger / 3; // row will be 0 (for 0,1,2), 1 (for 3,4,5), 2 for (6,7,8)

int col = inputInteger % 3; // col will be 0 (for 0,3,6), 1 (for 1,4,7), 2 for (2,5,8)

if (inputInteger < 0 || inputInteger > 8) // see if the input is below 1 or greater than 0

cout << "Illegal value, try again" << endl;

else if (board[row][col]=='X' || board[row][col]=='O')

cout << "This space has already been used, try again" << endl;

else

{

board[row][col] = player; // put in an 'X' or an 'O'

squareCounter++; // count the number of squares completed

// select the next player

if (player=='X')

player = 'O'; // switch from 'X' to 'O' for next move

else

player = 'X'; // switch from 'O' to 'X' for next move

}

gameWinner = CheckForWinningGame(board); // returns 'X' or 'O' if the game is won

}

if (squareCounter == 9)

{

DisplayTicTacToe(board);

cout << endl << "Tie Game" << endl << endl;

}

else if (gameWinner =='X' || gameWinner == 'O')

{

DisplayTicTacToe(board);

cout << endl << "Player " << gameWinner << " wins the game" << endl << endl;

}

return 0;

}

// ------ Display the Tic Tac Toe board

void DisplayTicTacToe(char board[3][3])

{

cout << endl;

for (int row=0; row<3; row++) // display three rows

{

for (int col=0; col<3; col++) // display three columns each row

cout << board[row][col] << " | ";

cout << endl << "-----------" << endl; // end of row

}

// cout << "--------------------------------" << endl << endl;

}

// ------ Check all eight possible combinations for a win.

// returns the player ('X' or 'O') if the game has been won

// returns '-' if there is currently no winner

char CheckForWinningGame(char board[3][3])

{

char player;

// check going across the rows to see if all squares are the same

int row;

for (int i=0; i<3; i++)

{

row = i;

player = board[row][0];

if (board[row][1] == player && board[row][2] == player) return player;

}

// check going down the columns to see if all squares are the same

int column;

for (int i=0; i<3; i++)

{

column = i;

player = board[0][column];

if (board[1][column] == player && board[2][column] == player) return player;

}

// check going across the left diagonal \ to see if all squares are the same

player = board[0][0];

if (board[1][1] == player && board[2][2] == player) return player;

// check going across the right diagonal / to see if all squares are the same

player = board[0][2];

if (board[1][1] == player && board[2][0] == player) return player;

return '-'; // if there is no winner

}