Project 1:

Tic Tac Toe

Assembly Programming

48982

Prepared for: Dr. Mark E. Lehr

Prepared by: David Haro, Student

Objective

The objective is to create a functional game/program with a minimum of 100 lines of code while
using the C++ language and the basic methods that have been covered in the class up to loops
and with the addition of functions.
Goals
The goal is to put together a simple game of Tic Tac Toe which can be played along with another
human player or in a single player mode with the computer. Ideally the best opponent is one that
provides a challenge, however, programming all the possible strategies that can be used to
guarantee either a win or a tie in Tic Tac Toe can be rather long and cumbersome.
Therefore the computer is set up as a novice, and can be beaten easily.

Project Size:

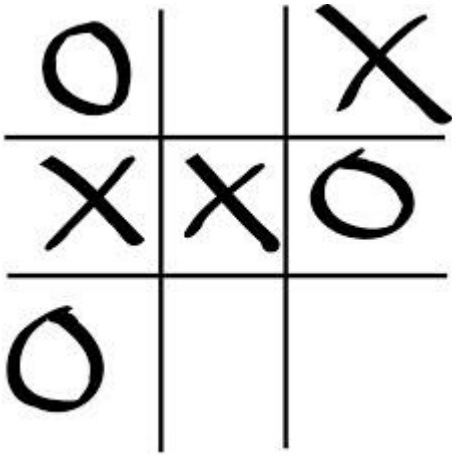
- Roughly 210 lines of code with about 16 lines dedicated to empty space with comments.
- Number of Variables: 17 Variables
- Number of Methods Used: 7

The rules of the game are simple. It is played with 2 players, each player select a symbol, either "X" or "O". Each symbol is used to fill a space in a 3 by 3 graph. Each turn a player will select from 1 space and the first player to get 3 in a row, whether vertical, horizontal or diagonal, will be the winner, else the game is deemed a draw.

Message from the Programmer

This program, while simple, took a lot of researching,
troubleshooting and debugging to get it where it is now.
The program implements concepts taught in C++ in Assembly. I hope you have
as much fun playing with this game as it took me to put it
together. (Hint: It was pretty fun and rewarding.)

David Haro



Pseudocode

Execute
Display game mode selectors
if single player is selected
    player= (player%2)?1:2; to generate player turn.
    player 1 inputs choice
        if choice is available square is taken by player
display board
checkWin
    if game is in progress, continue to next turn
    if game is a draw display draw
    if game is a win, display winner and board.
player++
else player repeats choice input.
player--
computer generates choice
if choice is available, square is taken by computer
display board
checkWin
if game is in progress, continue to next turn
if game is a draw, display draw
if game is a win, display winner
player++
else computer generates a new number for input
    player--
exit

Files

project1.s

Main/Source File 61 Lines of Code

This file is the main file which displays the introduction message and prompts for input from the user to determine if they would like to play the game or exit. This leaves room for expansion for the game in case the game grows to allow two users to play on one computer.

divmod_ML.s

Source File 67 Lines of Code

This file is borrowed from the professor's division modulus functions. The neat presentation and organization of this file allows for plug and play capability. The modulus method is used for generating random numbers for AI to input to the grid.

game_run.s

Source File 601 Lines of Code

Due to the length of this file, you can tell that much of the programming is inside this file. This file contains 34 different variables to store strings, scan, and print formatters for the table to play, the default data in the table, the input data from the user and the computer.

Methods/Functions/Labels

print_table

This method calls printf 6 times in order to display the play grid for Tic Tac Toe

start_play

This method calls the player variable and uses modulus to determine which player is currently playing. This allows to branch to appropriate labels during turns. This is always stored back in the player variable.

player_one/player_two

These methods read in input from user OR by random number generator and compares the input to determine which square has been chosen.

square_p1/square_p2

These methods are the same for each player. The method reads in data and compares the square with the squares next to them to determine if they are all equal. In the end if it has determined that each square shares the same player mark, it will check the player mark to determine the winner.

player_chkwin

This method reads in the squares to check for a winning combination. This is called at the end of each player turn. The method completes with a branch to the print table along with a display of which player has won (if there exists a winner).

External Libraries

Time

C time library used for seeding random number generator

srand/rand

C srand/rand library used for generating random numbers by use of the computer clock.

scanf

C built in function for reading in data.

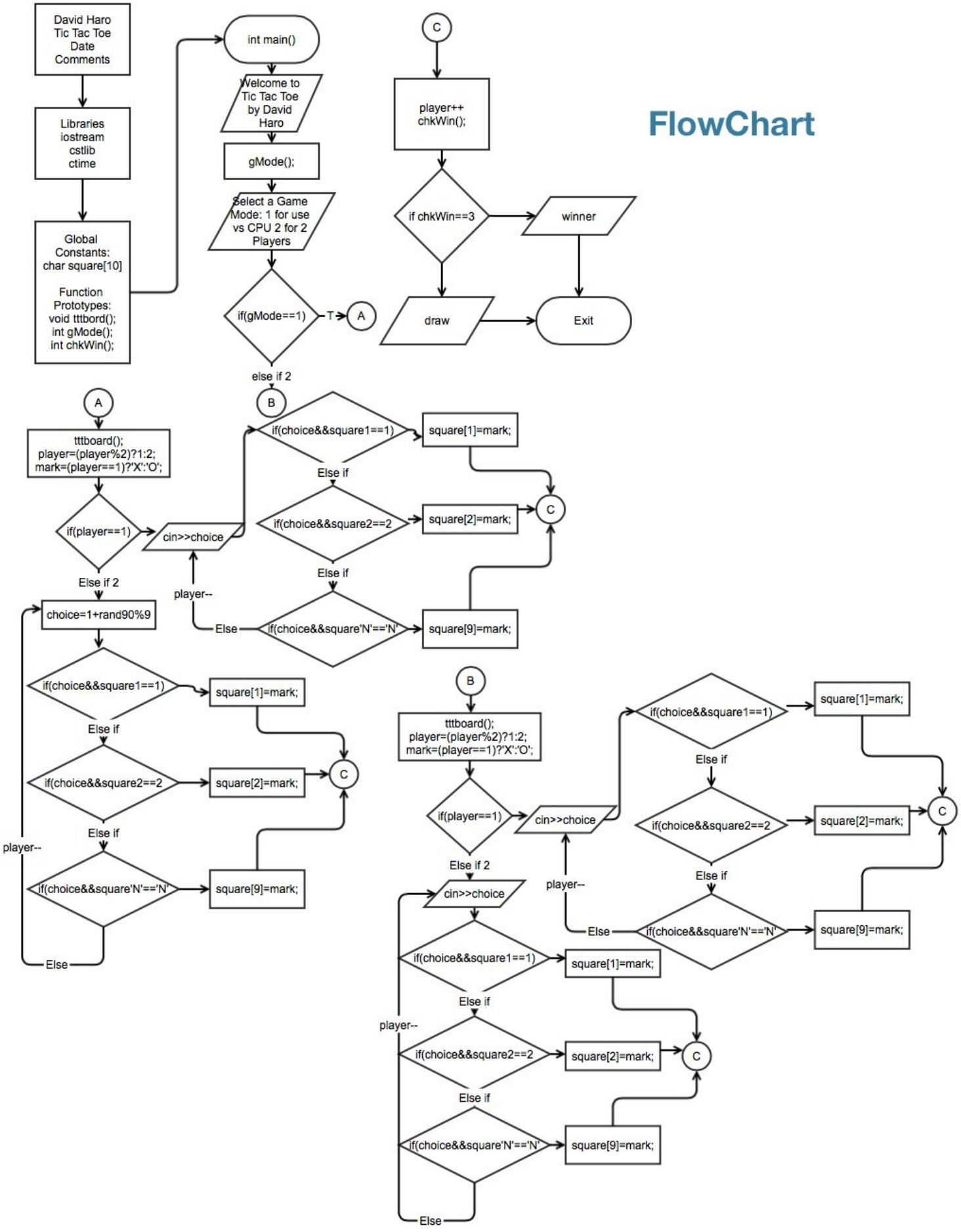
printf

C built in function for displaying formatted text.

Strcmp

Taken from string library. Used to compare two strings. It returns a true or false to register 0. Very useful when comparing strings.

FlowChart


David Haro
Tic Tac Toe
Date
Comments
Libraries
iostream
cstlib
ctime
Global
Constants:
char square[10]
Function
Prototypes:
void tttbord();
int gMode();
int chkWin();
int main()
Welcome to
Tic Tac Toe
by David
Haro
gMode();
Select a Game
Mode: 1 for use
vs CPU 2 for 2
Players
if(gMode==1)
T
A
else if 2
B
C
player++
chkWin();
if chkWin==3
winner
draw
Exit
A
tttboard();
player=(player%2)?1:2;
mark=(player==1)?'X':'O';
if(player==1)
cin>>choice
Else if 2
choice=1+rand90%9
if(choice&&square1==1)
square[1]=mark;
Else if
if(choice&&square2==2
square[2]=mark;
Else if
if(choice&&square'N'=='N'
square[9]=mark;
Else
player--
B
tttboard();
player=(player%2)?1:2;
mark=(player==1)?'X':'O';
if(player==1)
cin>>choice
Else if 2
cin>>choice
if(choice&&square1==1)
square[1]=mark;
Else if
if(choice&&square2==2
square[2]=mark;
Else if
if(choice&&square'N'=='N'
square[9]=mark;
Else
player--