

CS250 Assignment Two

Tic-Tac-Toe

Date Assigned: Monday, February 19, 2007

Date Due: Friday, March 2, 2007

Points: 35

For this assignment you are to use the principles of object oriented programming to program the game of Tic-Tac-Toe. You should set up the program so that a human can play against another human. You do not need to program any computer gameplay.

You are to create two classes for this program, and design a driver (main) that will use the classes to allow two humans to play against each other. Your program should perform any necessary error checking, and all the input and output should be handled from the driver. Your classes are not to interact with the user directly. In other words, your classes should not contain any cin or cout.

Since this is the first object-oriented program that you will write, I am going to give you the specifics about the classes you need.

GameBoard class

This class represents the current state of the gameboard. You can represent the board however you like, but the values of the board items should be integers (0 for empty, 1 for player one, 2 for player two). Your class can contain any other private functions that you need, but you must provide the following public functions:

<i>Function</i>	<i>Description</i>
GameBoard()	Constructor. Initializes the board to an empty game.
void printBoard()	Displays the game board onto the screen. You should use X and O for the output, not 1 and 2.
int getVal(int r, int c)	Returns the value of the board at a given row and column. The row and column values range from 1 - 3, and if either the row or column is invalid, the function should return -1.
bool setVal(int r, int c, int v)	Sets the value of the board at a given row and column to the given value. If the row or column is invalid the function should return false. It should return true otherwise.

Player class

The player class is used to store player information. For each player you will need to store their name and their id (1 for player one, 2 for player two). Again, the player class can contain any private functions but it must provide the following public functions:

<i>Function</i>	<i>Description</i>
Player(char * n, int id)	Constructor. Initializes the value of the name and the id.
void displayName()	Displays the name of the player onto the screen.
void makeMove(GameBoard &b)	This function accepts a GameBoard object and uses it to select the player's next move.
bool hasWon(GameBoard b)	Returns true if the player has won the game, false otherwise.
bool hasLost(GameBoard b)	Returns true if the player has lost the game, false otherwise.

Driver (Main program)

The main program should create objects from the above classes to simulate game play. You should ask for the name of two human player, then display an empty board. You are then to alternate gameplay displaying the state of the game board after each move, asking the human players for their move whenever it's their turn. You should indicate a tie or win at the end of the game.

Other details

- You should create 5 files for this assignment. A header and source file for the GameBoard class, a header and source file for the Player class, and a source file for the main.
- You must follow the coding standards.
- The class functions should be exactly as I describe them above. You must use the same names, return values, and argument types. This will allow me to test your classes against a driver that I will create.

How to submit

- Submit a hard copy of the code by 9am on the day the assignment is due. Your code must be stapled together in the following order: main.cpp, gameboard.h, gameboard.cpp, player.h, player.cpp
- Name your project "02PUNETTicTacToe", substituting PUNET with your PU Net ID, and place the complete project folder in the CS250 drop folder on Turing by 9am on the day the assignment is due.

Goals

1. Implement a program using OOP techniques.
2. Code and test your program one method at a time .
3. Learn to organize your program using .h & .cpp files for code reusability.

4. Use the new C++ coding standards.
5. Do basic error checking.