
CMSC 201 Spring 2016

Project 1 – Tic-Tac-Toe

Assignment: Project 1 – Tic-Tac-Toe

Due Date: Monday, April 18th, 2016 by 8:59:59 PM

Value: 80 points

Project 1 is the first assignment where we won't be telling you exactly what to do! You will get the chance to make your own decisions about how you want your program to handle things, what its functions should be called, and how you want to go about designing it.

Project 1 will also be ***substantially longer*** than any of the single homework assignments you've completed so far, so make sure to take the time to plan ahead, and don't do any "cowboy" coding!

Remember to enable Python 3 before you run your programs:

```
scl enable python33 bash
```

Instructions

For this assignment, you'll need to follow the class coding standards, a set of rules designed to make your code clear and readable. The class coding standards are on Blackboard under "Course Documents" in a file titled "CMSC 201 - Python Coding Standards."

You should be commenting your code, and using constants in your code (not magic numbers or strings). You should also have a function header comment for every function that is not `main()`!

Re-read the coding standards!

You will **lose major points** if you do not following the 201 coding standards.

A very important piece of following the coding standards is writing a complete **file header comment block**. Make sure that your file has a comment block at the top (see the coding standards document for an example).

NOTE: Your filename for this project must be `proj1.py`

NOTE: You must use `main()` in your file.

Details

For this project, you are going to be building the classic game tic-tac-toe in Python. In this game, one player is X and the other player is O. Players take turns placing their X or O. If a player gets three of their marks on the board in a row, column or one of the two diagonals, they win. When the board fills up with neither player winning, the game ends in a draw.

Some additional requirements:

1. The game will be played by two human players – you do NOT need to worry about computer AI for this assignment.
2. The person who will start will be assigned randomly. Either player 1 or player 2 may start.
3. The symbol assigned to each of the players will also be random. Either player 1 or player 2 may be “O”.
4. You must show a representation of the board between each player’s turn.
5. During a player’s turn, you can enter the number of the square that you would like to choose, -1 to save the game, or -2 to load a game. All other choices will be rejected.
6. The file that is saved should save the board, whose turn it is, and the symbol of the currently player. Use a static file named tic.txt for the save file.
7. The file that is loaded should bring in the board, whose turn it is, and the symbol of the current player. Use a static file named tic.txt for the load file.
8. After the winner is declared, the user should be prompted to play again.

Random Numbers

In order to choose the starting player and symbol, you are required to randomize who goes first and which symbol they will start with. In Python this is easy to do if we `import` the random library. The code below shows exactly how to do this:

The `random.randint(X,Y)` function returns a random number between X and Y (inclusive).

```
import random

# to get a random player between 1 and 2
player = random.randint(1,2)

# to get a random symbol between 1 and 2
symbol = random.randint(1,2)

# you will need to convert 1 and 2 to "X" and "O"
```

Input Validation

For this project, we will require that you validate input from the user. You can assume that the user will enter the right type of input, but not that they will enter a correct value. In other words, a user will always give an integer when you expect one, but it may be a negative or otherwise invalid value.

You will need to validate the following things:

- On a player's turn, they can enter the numbers 1-9 (the positions on the board) or a -1 to save to the file or a -2 to load the file. If anything else is entered, the user should be prompted again.
- After a player has won, the user should be prompted if they would like to play again, the system should allow any version of Y/n/YES/no (including all combinations of uppercase and lowercase)

We have provided additional sample output below showing how this should work in your program.

Sample Output – General Gameplay

Here is some sample output of a game being played, with the user input in blue. The sample output continues on to the following page.

```

bash-4.1$ python proj1.py
Welcome to Tic-Tac-Toe
This is for two players
Player 2 will go first and will play with the O
1|2|3
-----
4|5|6
-----
7|8|9

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 5
1|2|3
-----
4|O|6
-----
7|8|9

Player 1 what is your choice?
(1-9) or -1 to save or -2 to load: 1
X|2|3
-----
4|O|6
-----
7|8|9

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 3
X|2|O
-----
4|O|6
-----
7|8|9

```

```
Player 1 what is your choice?  
(1-9) or -1 to save or -2 to load: 4  
X|2|O  
-----  
X|O|6  
-----  
7|8|9  
  
Player 2 what is your choice?  
(1-9) or -1 to save or -2 to load: 7  
Player 2 wins!  
Play again?: no  
Thank you for playing
```

Sample Output – Input Validation

Here is some sample output that shows input validation, with the user input in blue. The sample output continues on to the following page.

```

bash-4.1$ python proj1.py
Welcome to Tic-Tac-Toe
This is for two players
Player 2 will go first and will play with the X
1|2|3
-----
4|5|6
-----
7|8|9

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: -10
Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 14
Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: -2
Game Loaded
0|2|X
-----
4|0|6
-----
7|8|9

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: -1
File Saved
Play again?: yes
Player 1 will go first and will play with the X
1|2|3
-----
4|5|6
-----
7|8|9

```

```

Player 1 what is your choice?
(1-9) or -1 to save or -2 to load: -2
Game Loaded
O|2|X
-----
4|O|6
-----
7|8|9

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 1
That space is already taken.
Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 9

O|2|X
-----
4|O|6
-----
7|8|X

Player 1 what is your choice?
(1-9) or -1 to save or -2 to load: 7
O|2|X
-----
4|O|6
-----
O|8|X

Player 2 what is your choice?
(1-9) or -1 to save or -2 to load: 6
Player 2 wins!
Play again?: no
Thank you for playing

```

Submitting

Once your Project 1 is complete, it is time to turn it in with the `submit` command.

Don't forget to complete the header block comment for your file! Make sure that you updated the header block's file name and description.

You must be logged into your GL account, and you must be in the same directory as the Project 1 file. To double check this, you can type `ls`.

```
linux1[3]% ls
proj1.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `PROJ1`. Type in (all on one line) `submit cs201 PROJ1 proj1.py` and press enter.

```
linux1[4]% submit cs201 PROJ1 proj1.py
Submitting proj1.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can **double-check that your file was submitted** by using the `submitls` command. Type in `submitls cs201 PROJ1` and hit enter.

And you're done!