

Tic-Tac-Toe

The goal of the assignment is to design and implement a Tic-Tac-Toe server and a Tic-Tac-Toe client that communicate with each other over a TCP/IP socket. In order to play the game, pairs of clients communicate with each other via the server.

The outline of the play is as follows. First the server starts and listens on some chosen port number. Then, each client starts and connects to the server at this port. The server assigns each client a player number: 0 or 1 and play begins.

Player 0 goes first and claims a board position by typing a number between 0 and 8 on the keyboard. This move is sent to the server, which updates the board by marking this position as being owned by player 0. It then sends the board position to both players. It is then the turn of Player 1 to play a round in the same way.

The players take turns until either:

1. one of the players has claimed three board positions in a row, horizontally or vertically, or diagonally, in which case this player wins, or
2. the board is filled in and neither player has won, in which case the game is a draw. The game ends when either one player wins or the game is drawn.

Note that a player may choose to play on a position that has already been claimed by itself or another player. In this case, the player's move is ignored.

See further instructions on the next page.

Further notes about the assignment:

1. You are encouraged to work in groups of two, three or four. If you cannot find partners, you may work alone. Please submit one copy of the assignment using subversion; if you are working with partners, all names should appear in the Python code (as comments).
2. Note that you will get a grade of zero if your program does not run.
3. You are to submit 3 files: `server.py`, `client.py` and documentation. Part of your documentation should be a succinct manual explaining how to run your program. The syntax of the commands should be as shown in the demo in class:

```
python server.py <port number>
python client.py <IP address> <port number>
```

4. The board configuration should be an array with positions numbered thus (we need some uniformity here to help us grade the assignments consistently):

0	1	2
3	4	5
6	7	8

5. When establishing the socket connection, the addresses and port numbers should *not* be hardcoded.
6. The assignment will be run for grading on Python 2.7.2.