Noah Jaffe
CMSC 481
11/05/2018

TCP Socket Programming - The tic tac toe protocol:

Tic-Tac-Toe Protocol Explanation:

SENDING FROM SERVER TO CLIENT:

pack message response length '!I'	`packed_msg_len = struct.pack('!I', <unsigned int>)`
SEND PACKED: MESSAGE RESPONSE LENGTH	`connection.sendall(packed_msg_len)`
SEND: MESSAGE	`connection.sendall(<string>.encode())`
pack expecting response val '!I'	`packed_response_val = struct.pack('!I', <unsigned int>)`
SEND PACKED: EXPECTING RESPONSE VAL	`connection.sendall(packed_response_val)`

RECEIVING FROM SERVER TO CLIENT:

RECV PACKED: MESSAGE RESPONSE LENGTH	`packed_msg_len = connection.receive(4)`
unpack '!I' and .recv that many bytes	`msg_len = struct.unpack('!I', packed_msg_len)`
RECV: MESSAGE	`message = connection.receive(msg_len).decode()`
RECV PACKED: EXPECTING RESPONSE VAL	`packed_response_val = connection.receive(4)`
unpack '!I'	`response_val = struct.unpack('!I', packed_response_val)`

SENDING FROM CLIENT TO SERVER:

pack single digit val '!I'	`packed_int_val = struct.pack('!I', <unsigned int>)`
SEND PACKED: SINGLE DIGIT VAL	`connection.sendall(packed_int_val)`

RECEIVING FROM CLIENT TO SERVER:

RECV PACKED: SINGLE DIGIT VAL	`packed_int_val = connection.recv(4)`
unpack '!I'	`int_val = struct.unpack('!I', packed_int_val)`

CONSTANTS AND GLOBALS:

```
TTT_SERVER_PORT = 13037
TTT_PRTCL_TERMINATE = 0
TTT_PRTCL_EXPECTING_NO_RESPONSE = 1
TTT_PRTCL_EXPECTING_INT_RESPONSE = 2
TTT_PRTCL_EXPECTING_FIRST_ARGS_RESPONSE = 3
TTT_PRTCL_PACKED_UNSIGNED_INT_SIZE = 4 #4 is the size of a packed '!I'
value
TTT_PRTCL_REQUEST_FIRST_ARGS =
```

```
Please send an unsigned int representing if the client wishes to make
the first move.
     0 -- sever should go first
     1 -- client should go first
```

```
TTT_PRTCL_GOT_FIRST_ARGS_ERR =
```

```
Failed to receive proper game initiation arguments. Terminating
connection.
Next time Please send an unsigned int representing if the client
wishes to make the first move.
     0 -- sever should go first
     1 -- client should go first
```

```
TTT_PRTCL_INSTRUCTIONS =
```

```
Welcome to Tic Tac Toe!
Enter [0-8] for the position of your move, or 9 to quit:
0|1|2
-----
3|4|5
-----
6|7|8
```

```
TTT_PRTCL_INVALID_CLIENT_INPUT =
```

```
Invalid input, try again.
```

```
TTT_PRTCL_REQUEST_CLIENT_TURN =
```

```
 | |            0|1|2
-----           -----
 | |            3|4|5
-----           -----
 | |            6|7|8
Enter [0-8] for the position of your move, or 9 to quit:
```

```
TTT_PRTCL_CLIENT_ERR =
```

```
Sorry, that was invalid input. Please try again.
```

TO CREATE YOUR OWN CLIENT:

I have provided a ttt.py that defines the four functions described above and has all the constants defined inside.
To use it you just need to import it.

example usage:

```
import ttt
...
ttt.SEND_FROM_CLIENT_TO_SERVER(my_socket_connection, my_data)
```