

Protocol for Lab 7 - CSE 5462 - Spring 2021
Version 5

All commands need to be sent with 5 bytes.

New Game Command

Protocol Version	Sequence Number	Command	undefined	undefined
------------------	-----------------	---------	-----------	-----------

EX: 0x 05 00 00 FF FF

05 represents the Protocol Version

00 represents sequence number 0, new game command should always be 0

00 represents New Game Command

The FF and FF can be anything as those columns are undefined and we don't care

Move Command

Protocol Version	Sequence Number	Command	Position	Game Number
------------------	-----------------	---------	----------	-------------

EX: 0x 05 01 01 31 00

05 represents the Protocol Version

01 represents sequence number 1

01 represents Move Command

31 Represents '1' in ASCII and will correspond to the top left position on the board

00 represents the game number. In this case the game number is 0

Game Over Command

Protocol Version	Sequence Number	Command	undefined	Game Number
------------------	-----------------	---------	-----------	-------------

EX: 0x 05 07 02 FF 00

05 represents the Protocol Version

07 represents sequence number 7

02 represents Game Over Command

FF is undefined, we don't care

00 represents the game number. In this case the game number is 0

Tic tac Toe grid layout

1	2	3
4	5	6
7	8	9

Player 1 is the "server"

- The server / player 1 goes first
 - Responds to client new game requests with the first move.

Player 2 is the "client"

Other Specifications:

- If server has a game running and gets another new game request, it handles it!
- Timeout Time: left to implementer
- On any errors (that aren't otherwise specified), close the socket
- Using Stream Sockets
- Server can handle multiple games....server MUST BE single threaded.
- Server will ignore new game requests if already full
- Server will ignore moves if the game doesn't exist or is over
- Clients are allowed to exit if they receive bad data
- Use select() for handling multiple sockets on server
- If the player loses they should send a Game Over Command to acknowledge they lost
- If player sends the last move they should wait until they receive the acknowledgement of game over
- Sequence number starts at 0 with the new game command gets incremented by both sides for each move and game over command
 - 0 for new game, 1 for server to client, 2 for client to server
- Timeout is left to implementer (you don't have to do it)

SOME SCENARIOS

- If the client asks server for a game, but the server can't run any more games...
 - Server just doesn't reply
- If the client sends a move to the server for a game that no longer exists...
 - Server just doesn't reply
- If the client gets data that it was not expecting ...
 - left to implementer