# Card Nim

This contains a server using basic sockets and a series of simple clients to play card nim

## Running the server

To run the server on localhost, from the top level, use the following command:

php server.php [port] [number of stones] [number of cards]

If you include a -o on the end, you can connect a websocket observer first as well.

Open observer/observer.html after starting the server to watch the two other clients play.

Note: html files in this repo are configured to use port 4000 by default, this can be configured

## Connecting via Telnet

To connect to the server to play via terminal run the following:

telnet localhost [port number]

An additional message is required to identify the client, the conents of this message are only important for websockets.

You can send any short string you want so long as it doesn't contain #Sec-WebSocket-Key:

Once two players are connected, you can request the current state with getstate and make moves with sendmove [number].

## Playing via Python, Java, and C++

To use these clients, after the server is up and running, run them with an optional command line argument for port number

4000 is the default port number for these clients

## Connecting via a browser

In the clients/html folder, there is an html file for playing the game visually.

To connect to the server, open the file in any modern browser which supports websockets, and follow onscreen prompts.

Note: html files in this repo are configured to use port 4000 by default, this can be configured

## Other Languages / Custom Bots / Game Protocol

We outline the basic communication steps here for those who want to use other languages, or write their own bots from scratch.

1. If the -o flag is active, connect to an observer via a websocket handshake
2. Wait for the first player to connect and identify with a dummy or websocket message
3. Wait for the second player to connect and identify with a dummy or websocket message
4. Send all participants their player num (1 or 2), number of stones, number of cards
5. Wait for messages from the current player (everything except getstate and sendmove [move] are ignored)
6. When a successful move is made, if the game isn't over, change players
7. When the game ends, stop listening and send out a final message with 0 or -1 depending on current player's turn

## Card Nim Rules

"You and your opponent are presented with some number of stones s.

The winner removes the last stone(s).

The first player chooses a card and removes exactly that number of stones.

The card then disappears from the first player's hand.

Similarly for the second player."

## Original Authors

Zachary DeStefano

Graham Todd

## Contributors

For questions about the code send an email to Jason Zhang (jzz2003@nyu.edu) or Raymond Lu (yl12043@nyu.edu).