

README- README file for Networks Project |
| Tufts University, COMP 112 [Fall 2015] |
| created by: Sean McKeever, Mathurshan Vimalasvaran |
| last modified: 7 December 2015 |

I	CONTENTS	I
I	I. Overview	I
I	II. Functionality	I
I	III. Limitations	I
I	IIII. How To Run	I

I. OVERVIEW

We developed a peer to peer poker game. Users will be able to connect to a server

to join a peer hosted poker table of up to 5 players. Many tables with different hosts can exist at one time.

II. FUNCTIONALITY

-POKER!

-Server recognizes new users and keeps track of old users' info

-Server is able to find open tables for players

-Server allows users to "cash out" and return their chips for dollars (dollars aren't tracked)

-Server allows users to "buy chips" and use dollars to get more chips (dollars aren't tracked)

-Server uses "update" protocol to keep track of tables

-Main peer updates server each hand on the number of players in the table and the number of chips each player has.

-If main peer disconnects, each peer updates server of disconnection and server removes

table from its database. This is in case more than one peer disconnects, the server is

going to get the alert from someone.

-If peer disconnects, main peer treats it as a "fold" move for the round, updates the

server, and then removes the player from its own data.

-If the table is not full, and every member of a table disconnects at the same time

then the server
will learn about this when a new user attempts to connect to this table. If a new
user cannot
connect to a table host, the server will remove that table from its database and
send the
user to a new table.

III. LIMITATIONS

-If the entire game goes down (all people in the table disconnect at once), then
the
server has no way of knowing the game state at the moment of failure, since
there
is no one left to update the server.
-In order to focus on the networking side of things, very complex edge cases of
poker
were disregarded, including the case of an all in split pot.
-The server does not handle bad requests or a broken pipe during a receive call
-There are a few buggy output statements that will print twice in a row at times
-As more of a design flaw than a limitation, the Player class could have been split
into
smaller subclasses and been implemented with inheritance.

IIII. HOW TO RUN

Refer to test_server.py, test_player1.py, and test_player2.py