Client.Client Class Reference

Public Member Functions

def **__init__** (self)
setup the client More...

def **createGame** (self, username, password)
inform the server we wish to create a game More...

def **test** (self)
test function More...

def **poll** (self)
poll function More...

def **join** (self, address, username, password)
ask to join a specific game with username and password More...

def **listGames** (self)
Accessor: returns a list of games client has heard about. More...

def **addToQueue** (self)
Adds data to the queue. More...

Static Public Attributes

int	**BROADCAST_PORT** = 44470
int	**TRANSMIT_PORT** = 44469
string	**BROADCAST_ADDRESS** = "255.255.255.255"
string	**BOARD_FILE** = "text/DublinBoard.txt"

Private Member Functions

def **_transmit** (self, transmit_port)
to be run in a thread and handle outgoing messages to specific server More...

def **_newGame** (self, data)
handles a GAME message from the server by adding it to the list of open games self the object pointer data the data TODO: wtf is data More...

def **_gameStart** (self, data)
handles a START message from server by creating the board and passing it to GUI self the object pointer data the data TODO: wtf is data More...

def **_newTurn** (self, data)
handles a TURN message from the server by telling GUI who's turn has begun More...

def **_hasQuit** (self, data)
Calls the board.RemovePlayer() method if a player quits the game. More...

def **_bought** (self, data)
update the owner of some space in board to be player with given id More...

def **_paid** (self, data)

update players balances as they change, 1 or more depending on transaction More...

def **_jailed** (self, data)
some player got sent to jail so change their jail status More...

def **_sentchat** (self, data)
send a message from the server to the textbox display More...

def **_drewCard** (self, data)
you drew a card so tell the GUI about it and check if you're on bail More...

Private Attributes

_connection_queue

_socket

_transmitter

_open_games

_board

_local_player

Constructor & Destructor Documentation

◆ __init__()

def Client.Client.__init__ (self)

setup the client

Member Function Documentation

◆ _bought()

def Client.Client._bought (self,
data
)

private

update the owner of some space in board to be player with given id

Parameters

self the object pointer

dict data the data stream from client & server

◆ _drewCard()

```
def Client.Client._drewCard ( self,
                              data
                            )
```

you drew a card so tell the GUI about it and check if you're on bail

Parameters

- **self** the object pointer
- **dict** data the data stream from client & server

◆ _gameStart()

```
def Client.Client._gameStart ( self,
                               data
                             )
```

private

handles a START message from server by creating the board and passing it to GUI self the object pointer data the data TODO: wtf is data

◆ _hasQuit()

```
def Client.Client._hasQuit ( self,
                             data
                           )
```

private

Calls the board.RemovePlayer() method if a player quits the game.

Parameters

- **self** the object pointer
- **dict** data the data stream from client & server

◆ _jailed()

def Client.Client._jailed (self,
data
)
private

some player got sent to jail so change their jail status

Parameters

- **self** the object pointer
- **dict** data the data stream from client & server

◆ _newGame()

def Client.Client._newGame (self,
data
)
private

handles a GAME message from the server by adding it to the list of open games self the object pointer data the data TODO: wtf is data

◆ _newTurn()

def Client.Client._newTurn (self,
data
)
private

handles a TURN message from the server by telling GUI who's turn has begun

Parameters

- **self** the object pointer
- **data** the data TODO: wtf is data

◆ _paid()

```
def Client.Client._paid ( self,
                          data
                        )
```

private

update players balances as they change, 1 or more depending on transaction

Parameters

- **self** the object pointer
- **dict** data the data stream from client & server

◆ _sentchat()

```
def Client.Client._sentchat ( self,
                              data
                            )
```

private

send a message from the server to the textbox display

Parameters

- **self** the object pointer
- **dict** data the data stream from client & server

◆ _transmit()

```
def Client.Client._transmit ( self,
                              transmit_port
                            )
```

private

to be run in a thread and handle outgoing messages to specific server

◆ addToQueue()

```
def Client.Client.addToQueue ( self )
```

Adds data to the queue.

Parameters

- **self** the object pointer

◆ createGame()

```
def Client.Client.createGame ( self,
                               username,
                               password
                             )
```

inform the server we wish to create a game

◆ join()

```
def Client.Client.join ( self,
                         address,
                         username,
                         password
                       )
```

ask to join a specific game with username and password

Parameters

self	the object pointer
int	address the address (INT ARE YOU SURE?)
String	username ...y'know, the user's name
String	password ...uhh...

◆ listGames()

```
def Client.Client.listGames ( self )
```

Accessor: returns a list of games client has heard about.

Parameters

self	the object pointer

◆ poll()

```
def Client.Client.poll ( self )
```

poll function

Parameters

self	the object pointer

◆ test()

def Client.Client.test (self)

test function

Parameters

self the object pointer

Member Data Documentation

◆ _board

Client.Client._board private

◆ _connection_queue

Client.Client._connection_queue private

◆ _local_player

Client.Client._local_player private

◆ _open_games

Client.Client._open_games private

◆ _socket

Client.Client._socket private

◆ _transmitter

Client.Client._transmitter private

◆ BOARD_FILE

string Client.Client.BOARD_FILE = "text/DublinBoard.txt" static

◆ BROADCAST_ADDRESS

string Client.Client.BROADCAST_ADDRESS = "255.255.255.255"

◆ BROADCAST_PORT

int Client.Client.BROADCAST_PORT = 44470 static

◆ TRANSMIT_PORT

int Client.Client.TRANSMIT_PORT = 44469 static

The documentation for this class was generated from the following file:

- D:/Documents/CS3306/monopoly/Monopoly-master/Monopoly-master/**Client.py**

Generated by doxygen 1.8.13