Classes

Client – DONE

Methods

* Open socket
* Open input/output streams
* Set up board and then pass that to game class -- TO DO
  + Position and piece type
* Update board
* Ask for player2 input (moves)

Server – TO DO

Methods

* Open serversocket
* Open input/output streams
* Open socket
* Set up board and then pass that to game class
  + Position and piece type
* Moves/messages
  + Strings
    - 0-player to player
    - 1- game to game
      * 1- start x, start y, end x, end y
  + Waiting to send moves alternatively
  + Send messages when you get them

OFFLINE GAME

Methods

* Changed so it can handle
* To use
  + executeAction
* modify
  + run

ONLINEGAME – TO DO

Methods

* Join or host a game
  + If host: gui get port   
    if join: get domain name or port num

~~LocalPlayer~~

~~Methods~~

* ~~Ask player1 for input- getInput(); @return move~~
* ~~Display results to player1~~
* ~~Displayer board to player1~~
* ~~Update board~~

~~Online~~

* ~~Create gui?~~

~~Methods~~

* ~~Get player input: getInput()~~
* ~~Pass move to game class~~
* ~~Get result from game class~~
* ~~Update both player classes~~
* ~~Update the boards~~
* ~~Display the results to the locaplayer class and send to server to send to client class~~

While someone hasn’t won; have each player take turns making a move;

Function:

Client creates connection; server asks for input; player2 inputs move in client; server reads move and passes to game class; game class looks at board objects and if player2s piece is of higher rank (as determined by the piececomparator; passes both pieces into comparator class), player1 loses their piece; the board objects are both updated; the player classes are updated; the game class determines the result and passes it to the server and the localplayer classes; the server passes it to the client

Then the local player classes asks player1 for their move; this gets passed to the game class; game class looks at board objects and if player1s piece is of higher rank (as determined by the piececomparator; passes both pieces into comparator class), player2 loses their piece; the board objects are both updated; the player classes are updated; the game class determines the result and passes it to the server and the localplayer classes; the server passes it to the client

DRAWS—ALL MOVEABLE PIECES ARE GONE--??

USER MANUEL—TO DO