Assignment 2 part 1: Othello Networking UI

Value: 6% of your overall grade.

Due date: November 22nd, 2020 (Sunday, midnight)

Purpose:

If microwave ovens, door locks and security cameras can be networked these days, why not our Othello client? We need to get some Internet functionality in here!

But seriously:

We're going to start adding some network functionality to our game. When we're done, you'll be able to connect to a server and communicate with other Othello players. It will, in essence, be a very simplistic Discord client and server.

Minimum Requirements

It would be ideal if you had the full working game at this point, but as a basic minimum for this assignment, you will need a UI that contains:

- The pink "output area" as a text area, and
- A menu system of some sort.

For Assignment 2 part 2, you will also need:

- A text field, and
- A working "Submit" button.

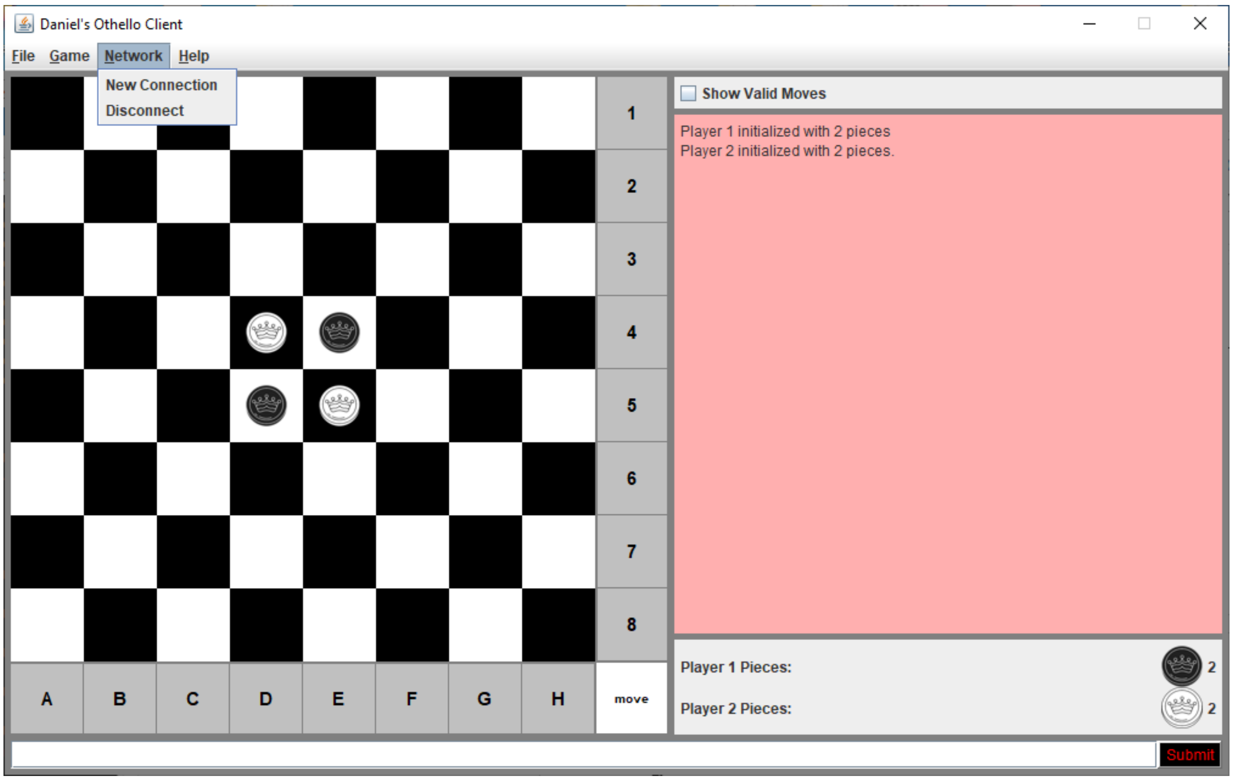
Note that if you have a working assignment, disregard the above, you're in a good position.

Operation:

You will first need to add a new menu, called Network, which will appear between "Game" and "Help" as the third menu.

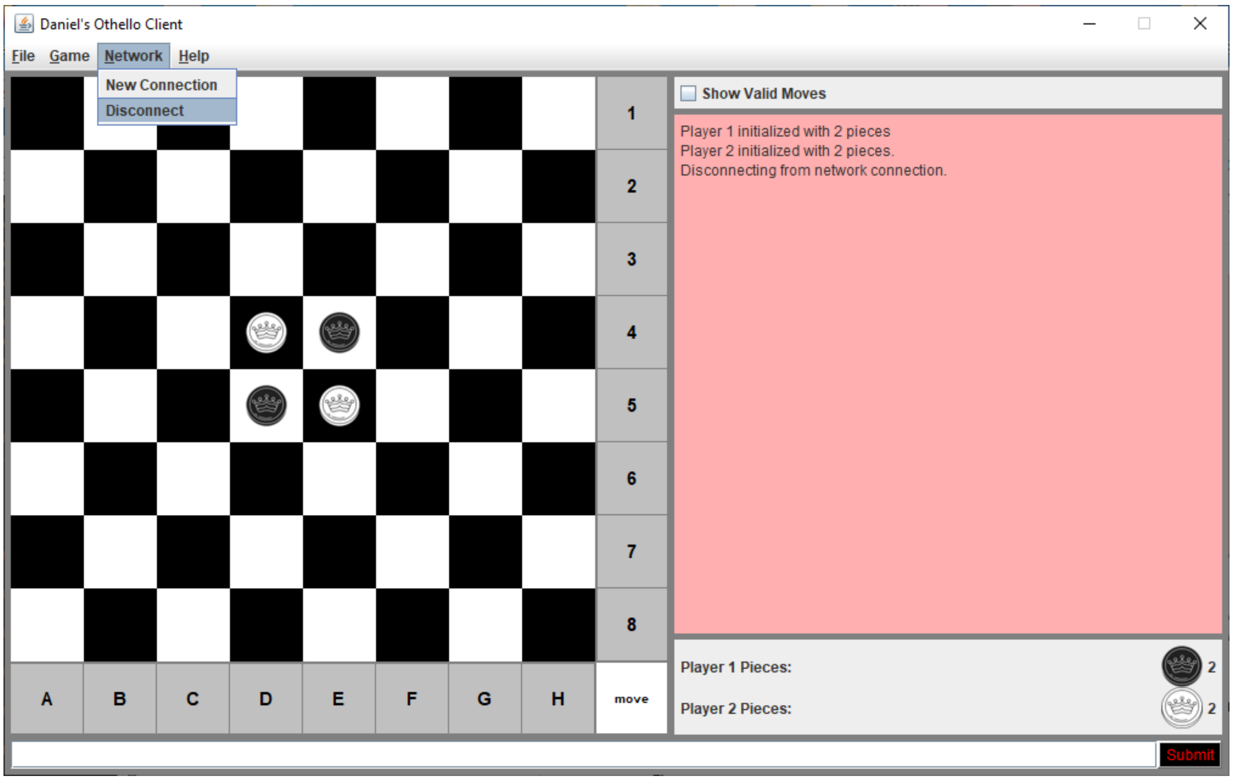
It will have two menu items":

- New Connection, and
- Disconnect



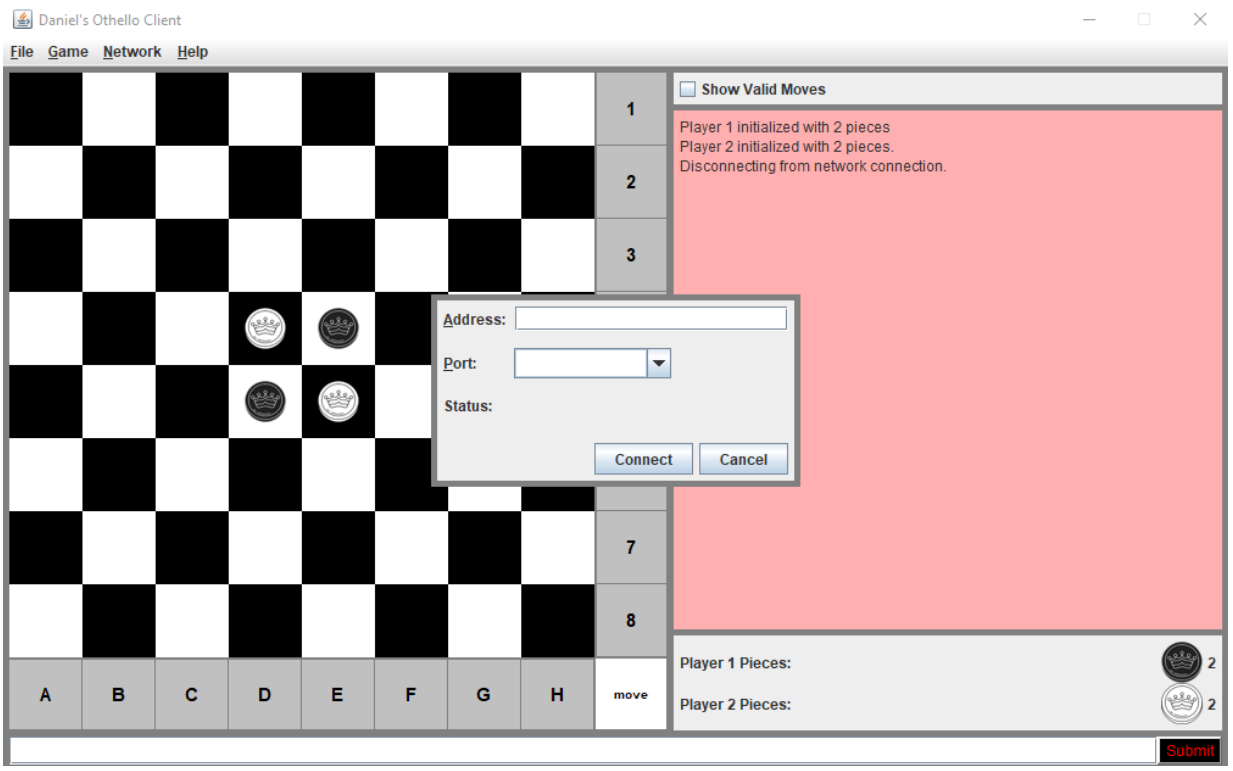
```
Player 1 initialized with 2 pieces
Player 2 initialized with 2 pieces.
```

When “Disconnect” is selected, all it should do is add a bit of text to the output zone: “Disconnecting from network connection.”



```
Player 1 initialized with 2 pieces
Player 2 initialized with 2 pieces.
Disconnecting from network connection.
```

Selecting “New Connection” will make a modal dialog appear:



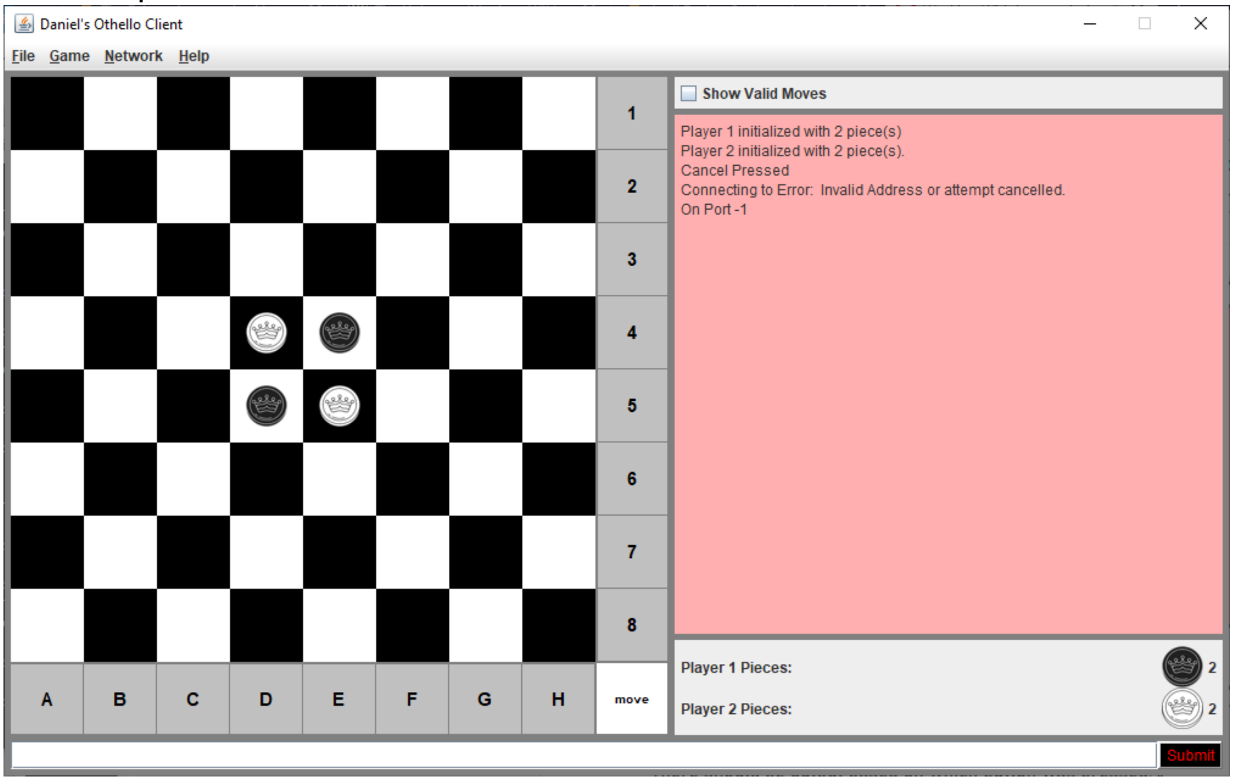
```
Player 1 initialized with 2 pieces
Player 2 initialized with 2 pieces.
Disconnecting from network connection.
```

Player 1 Pieces: 2

Player 2 Pieces: 2

There should be output based on which button was pressed.

Cancel is pressed:

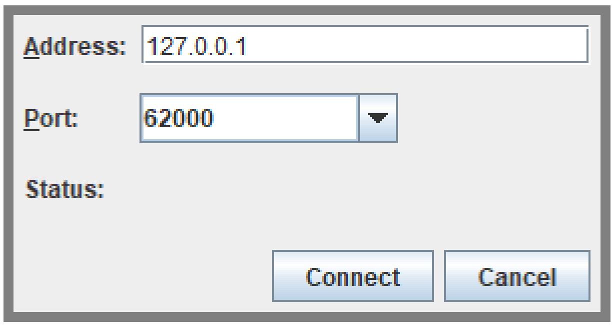


```
Player 1 initialized with 2 piece(s)
Player 2 initialized with 2 piece(s).
Cancel Pressed
Connecting to Error:  Invalid Address or attempt cancelled.
On Port -1
```

Player 1 Pieces: 2

Player 2 Pieces: 2

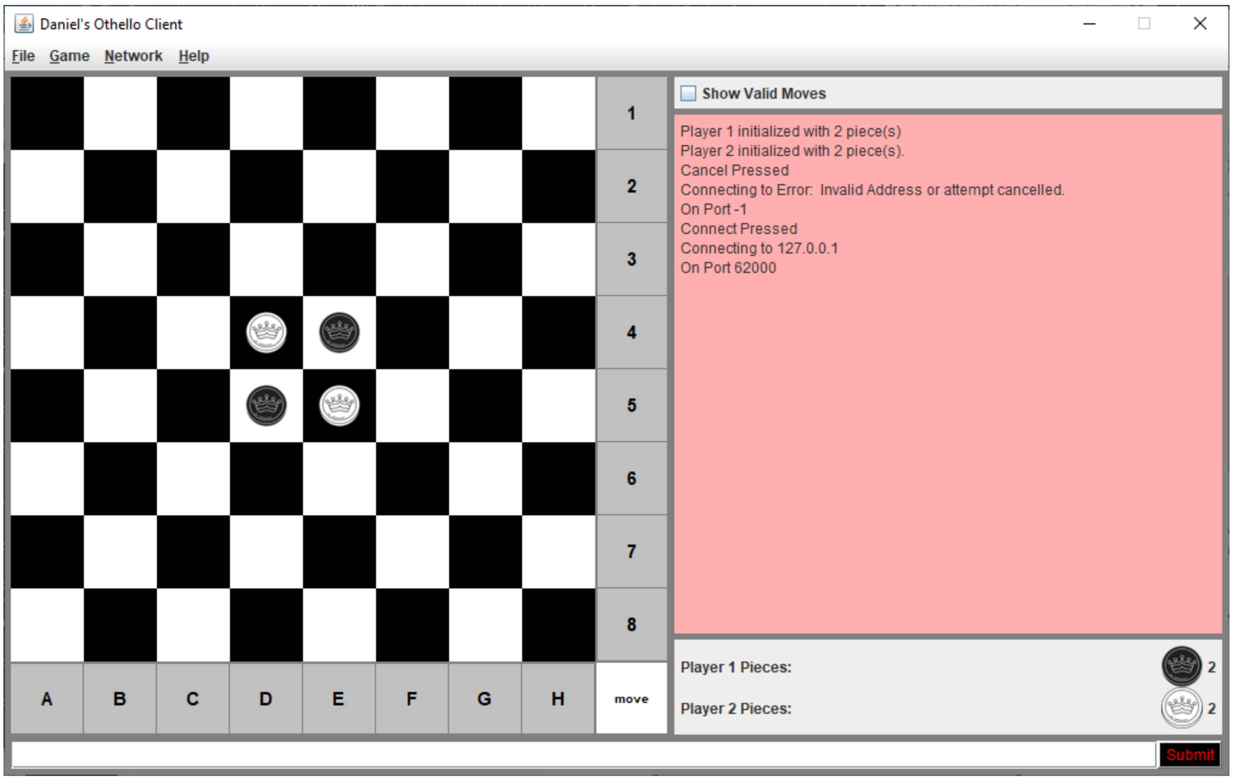
If I enter information (an example is provided below) and I press connect,


Address: 127.0.0.1

Port: 62000

Status:

I should get



```
Player 1 initialized with 2 piece(s)
Player 2 initialized with 2 piece(s).
Cancel Pressed
Connecting to Error:  Invalid Address or attempt cancelled.
On Port -1
Connect Pressed
Connecting to 127.0.0.1
On Port 62000
```

Player 1 Pieces: 2

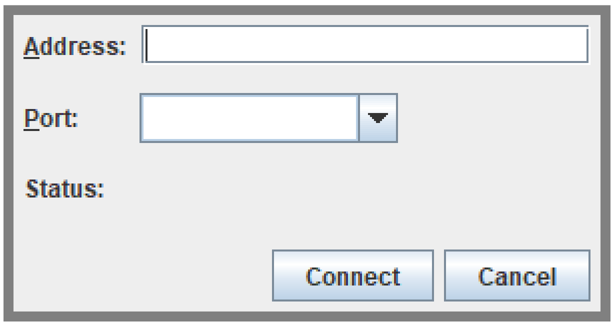
Player 2 Pieces: 2

In short, the Controller should read the values from the Dialog, and report those to the View for display.

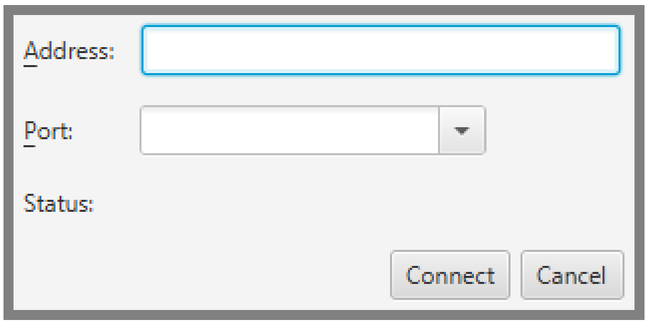
Your chief goal will be to implement the UI in this dialog. It must only use *border layouts*, *flow layouts* and *grid layouts*. You do not need to use all of those, but you may only use those.

This code will exist in a class called OthelloNetworkModalViewController. A stub has been provided for you with much of the implementation, but you will need to finish it. When you are done, the modal itself should look like this:

Swing:



JavaFX:



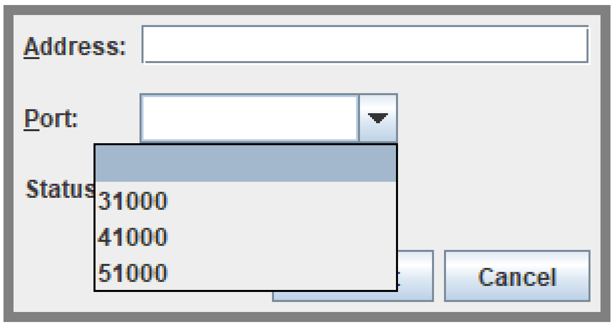
Appearance and Components

I am looking for a five-pixel border around the entire UI and five pixel horizontal internal spacing.

Address, Port and Status are labels. Address and Port contain mnemonics (A and P respectively), which are *labels for* (that's a very strong hint) the textfield and combo box they're beside.

So if I press alt-P, my cursor should jump to the combo box. If I press alt-A, it should jump back to the text field *beside* the word "address".

The combo box must start empty, and contain three port numbers: 31000, 41000 and 51000. It must be editable; the user should be able to type in any value.



Finally, the text field must be *20 columns wide*.

There are further comments inside the stub. Take the time to read through them, as you may be required to implement a small thing or two inside.

Extra Implementation Details for Swing

Instantiate your OthelloNetworkModalViewController early in your code as a field. It takes one parameter, which should be "this":

```
OthelloNetworkModalViewController networkDialog = new
OthelloNetworkModalViewController(this);
```

In your Controller, in the portion that deals with your Network menu items, you will need this code:

```
Point thisLocation = getLocation();
Dimension parentSize = getSize();
Dimension dialogSize = networkDialog.getSize();

int offsetX = (parentSize.width-dialogSize.width)/2+thisLocation.x;
int offsetY = (parentSize.height-dialogSize.height)/2+thisLocation.y;

networkDialog.setLocation(offsetX,offsetY);
networkDialog.setVisible(true);
```

This will make your modal appear when you select "New Connection", and center it on your Othello client.

You will then want to read the various getter methods off the network dialog to update your View accordingly.

Extra Implementation Details for JavaFX

Firstly, you won't need to pass any parameters. Your instantiation should look like this:

```
OthelloNetworkModalViewController networkDialog = new
OthelloNetworkModalViewController();
```

Secondly, to launch it, your Controller only needs to contain this line:

```
networkDialog.showAndWait();
```

That's it! At this point you would then read the network dialog's get methods to determine how to update your View, exactly the same as those using Swing.

Good luck on your assignment, and remember that "*If you want a good client, you will need a good server.*" --Anonymous Network Programmer.