ADVANCED PROGRAMMING ASSESSMENT 1

English-Draughts

OCTOBER 4, 2020

BOKYUNG LEE
2431088L

Contents

Aims & Rules

English-Draughts

This socket programming for English-Draughts game is for a two-player in 8 * 8 board. Game follows regular rules of English-Draughts except calculating time, recording and winning process therefore the match continued in turn and when one player wins game ends. In each turn of the game, opponent cannot make any actions unless will get warning message while actions are available by mouse click.

Entire game process screenshots are attached in appendix.

Normal rules for 2 players are as follows:

1. The piece can be moved to empty square that is diagonally adjacent to and below it.
2. The piece can be jumped and remove opponent’s piece when both pieces are at the diagonally adjacent location.
3. The piece which had reached to the opponent’s side may changed to the square piece that occupies backward function and called “King”.

Winning process as follows:

1. Player1 and Player2 may both make “King” pieces.
2. Player who moves pieces to result no remaining pieces of the opponent may win the game.

Code Running

The program requires access to the “/src” file.

1. Run the server (Server.java). Then the server connects via the port and host address.
2. Run the client (Client.java). Then player1 may ready. Session 1 may on the progress.
3. Run the client (Client.java) in another command-line page. Then player2 may ready and the game will be started. Session 2 may on the progress.

Known issues:

1. Server could not participate in game.
2. When game ends both players screen may close with alert message: “Game is over”
3. To restart the game, it is necessary to rerun “Client.java” twice for two players.
4. However server may not closed until the user manually closes.

Application protocol & Mechanisms

The server is listening on port 8765 of “127.0.0.1” host address. Both the server and client may initiate commands.

1. Client connection to server – Socket listener connects via host address and port number

When the connection to server fails due to the unknown host exception, error message dialog came out and system will exit the program automatically.
While the problem occurred in connection broadly, IOException may catches and prints out error message dialog then exit the program automatically.

2. Client name – client name should be inserted by player

A client name should not be null and the length should be longer than 0.

```
String name = (String) JOptionPane.showInputDialog(null, "Enter your name to
Connect", "Connect to Server", JOptionPane.OK_CANCEL_OPTION);
```

The client constructor may print message dialog and exit the client program.

```
JOptionPane.showMessageDialog(null, "Name should not be null", "Error",
        JOptionPane.ERROR_MESSAGE, null);
System.exit(0);
```

3. Run the session

When both clients plays the game, session runs to send data back and forth. If the exception occurred, “Connection is being closed” message will printed out and disconnect the player who is still online.

```
System.out.println("Connection is being closed");

if(player1.isOnline())
        player1.closeConnection();

if(player2.isOnline())
        player2.closeConnection();
```

4. SendData – send player’s game data

Player’s data may passed and return 1 if the result is successful and -1 if the result is failed.

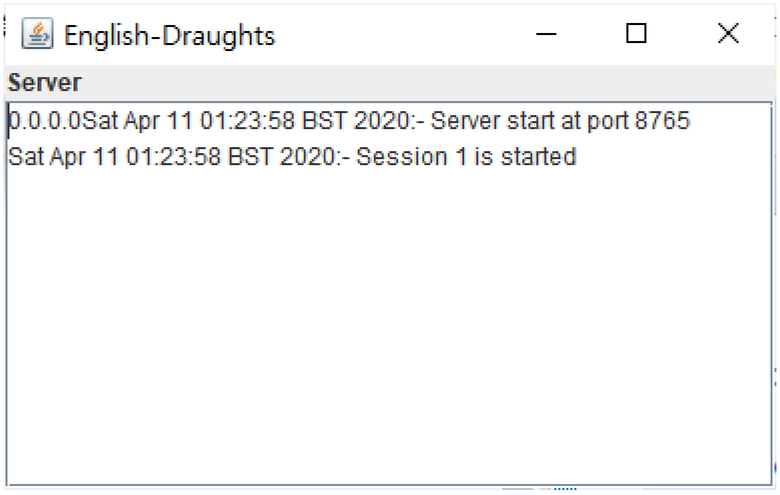
Additionally, when the result is failed, prints out “sending: player not found”

5. ReceiveData – receive player’s game data

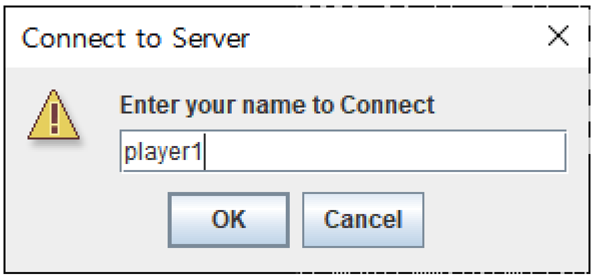
Receive the data and return while if the program catches IOException, prints out “No response” and return -1.

Appendix

Process of game.

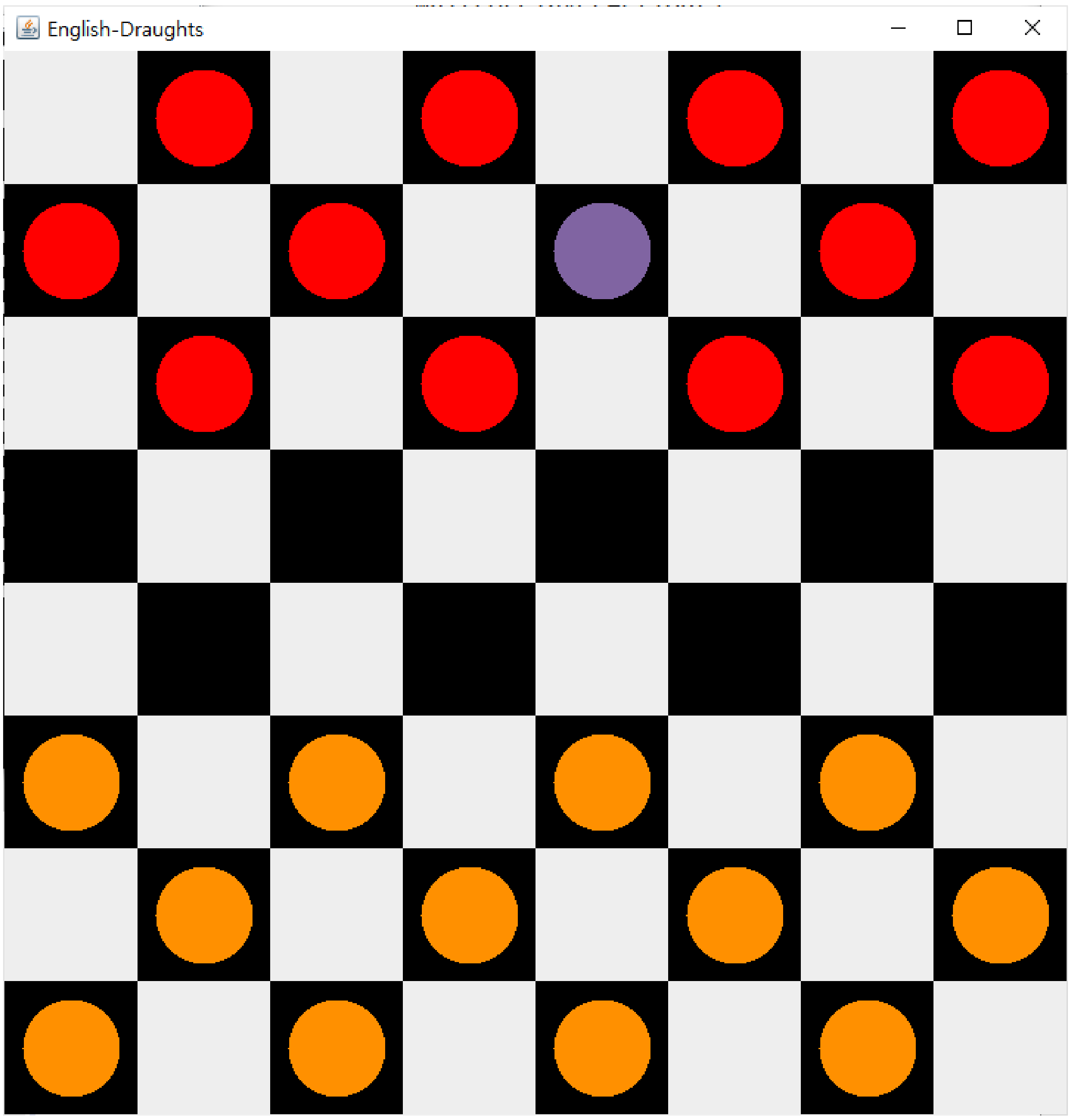


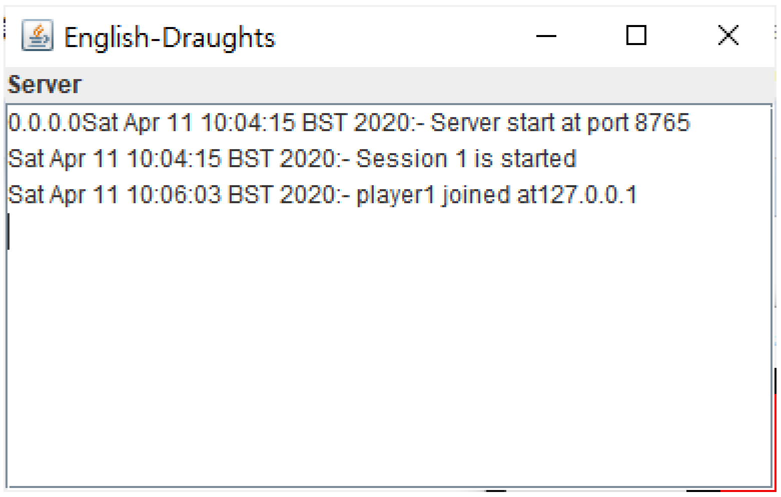
```
0.0.0.0Sat Apr 11 01:23:58 BST 2020:- Server start at port 8765
Sat Apr 11 01:23:58 BST 2020:- Session 1 is started
```



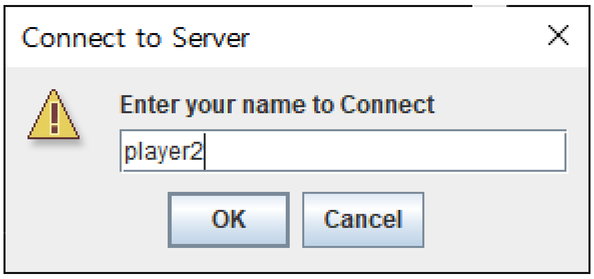
Enter your name to Connect

player1





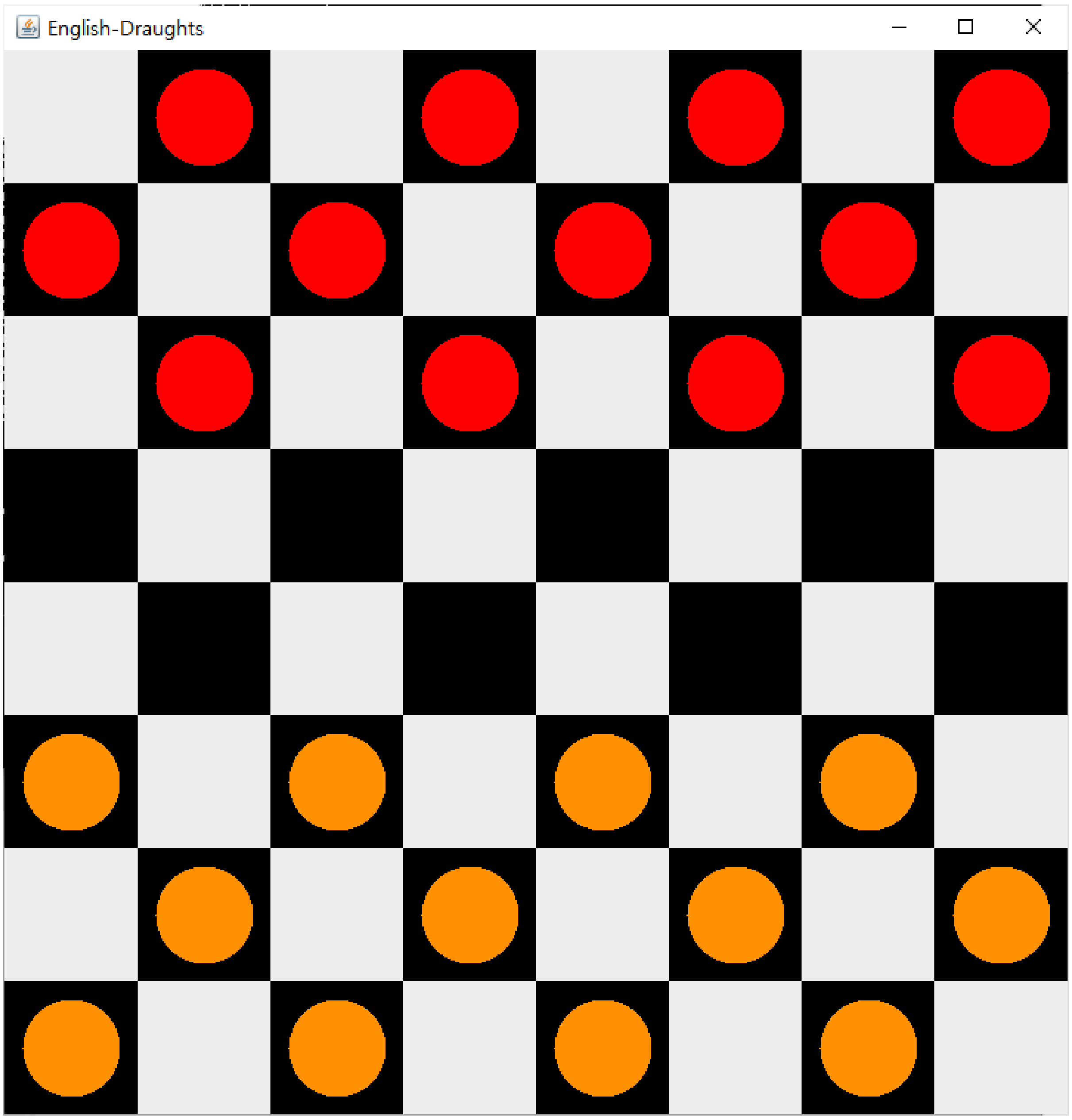
```
0.0.0.0Sat Apr 11 10:04:15 BST 2020:- Server start at port 8765
Sat Apr 11 10:04:15 BST 2020:- Session 1 is started
Sat Apr 11 10:06:03 BST 2020:- player1 joined at127.0.0.1
```

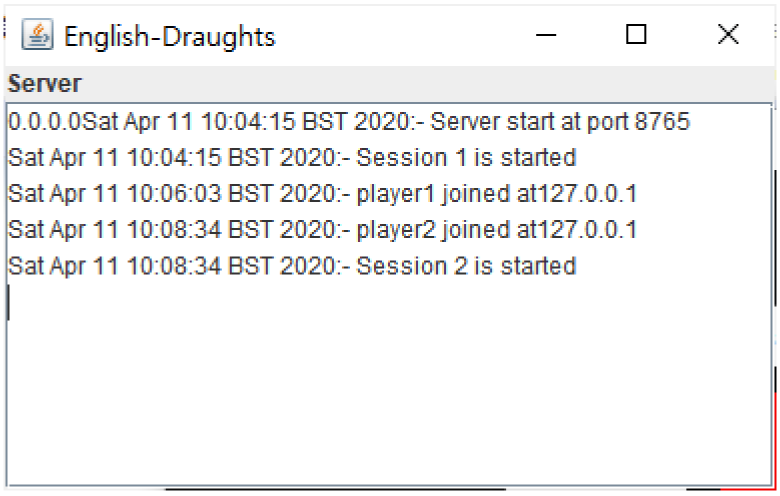


Connect to Server

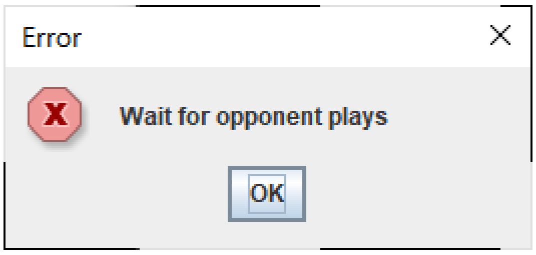
Enter your name to Connect

player2

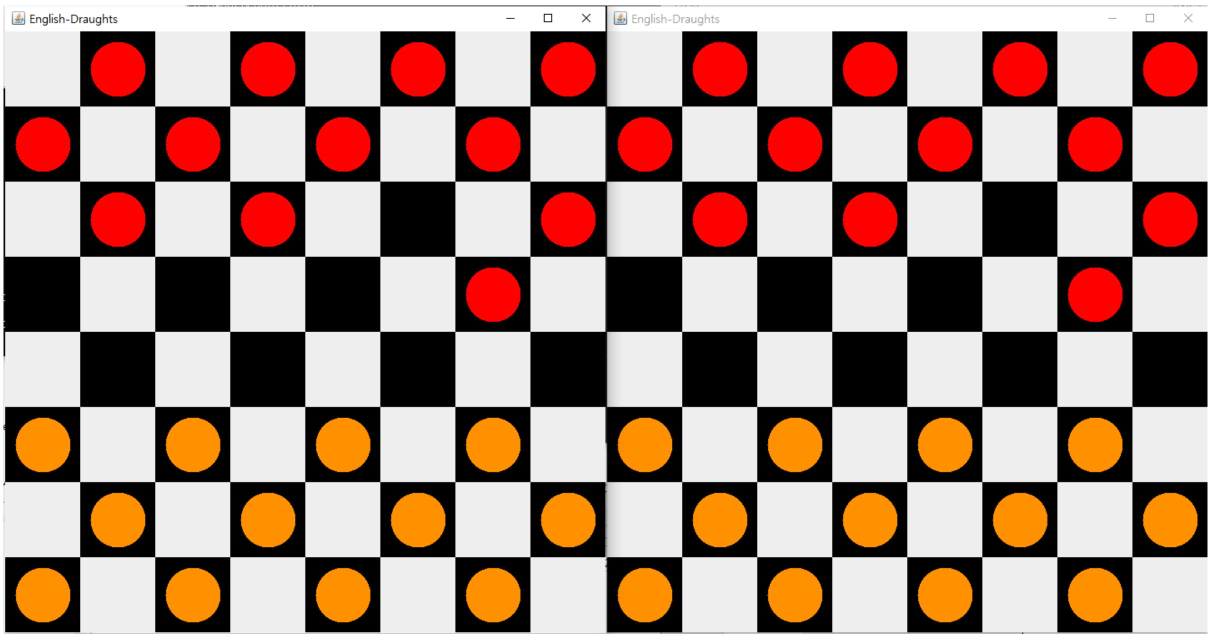


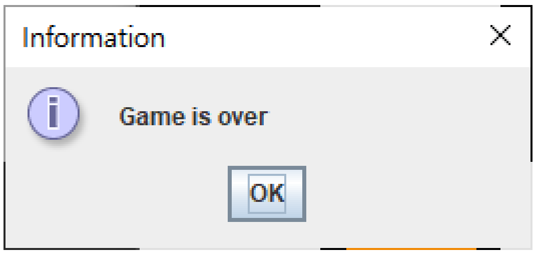


```
0.0.0.0Sat Apr 11 10:04:15 BST 2020:- Server start at port 8765
Sat Apr 11 10:04:15 BST 2020:- Session 1 is started
Sat Apr 11 10:06:03 BST 2020:- player1 joined at127.0.0.1
Sat Apr 11 10:08:34 BST 2020:- player2 joined at127.0.0.1
Sat Apr 11 10:08:34 BST 2020:- Session 2 is started
```

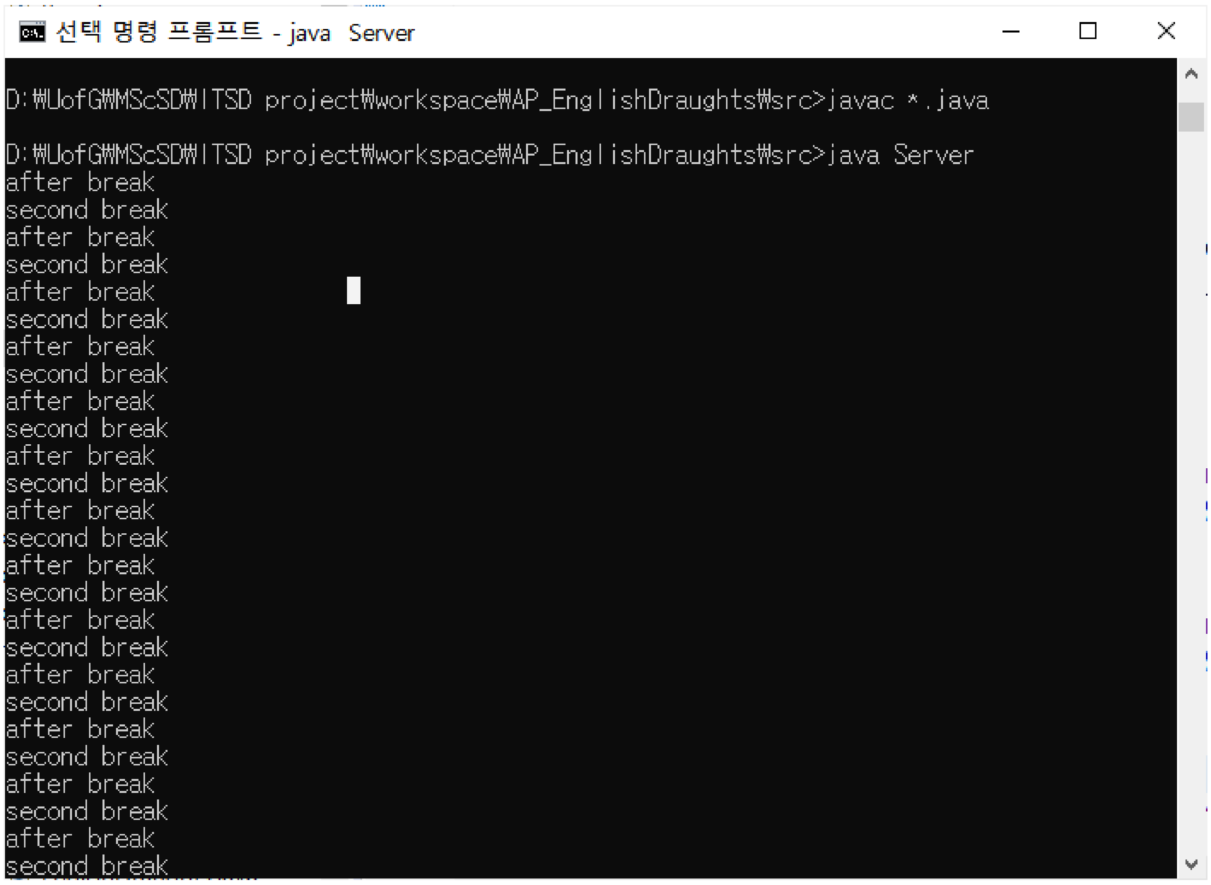


Wait for opponent plays



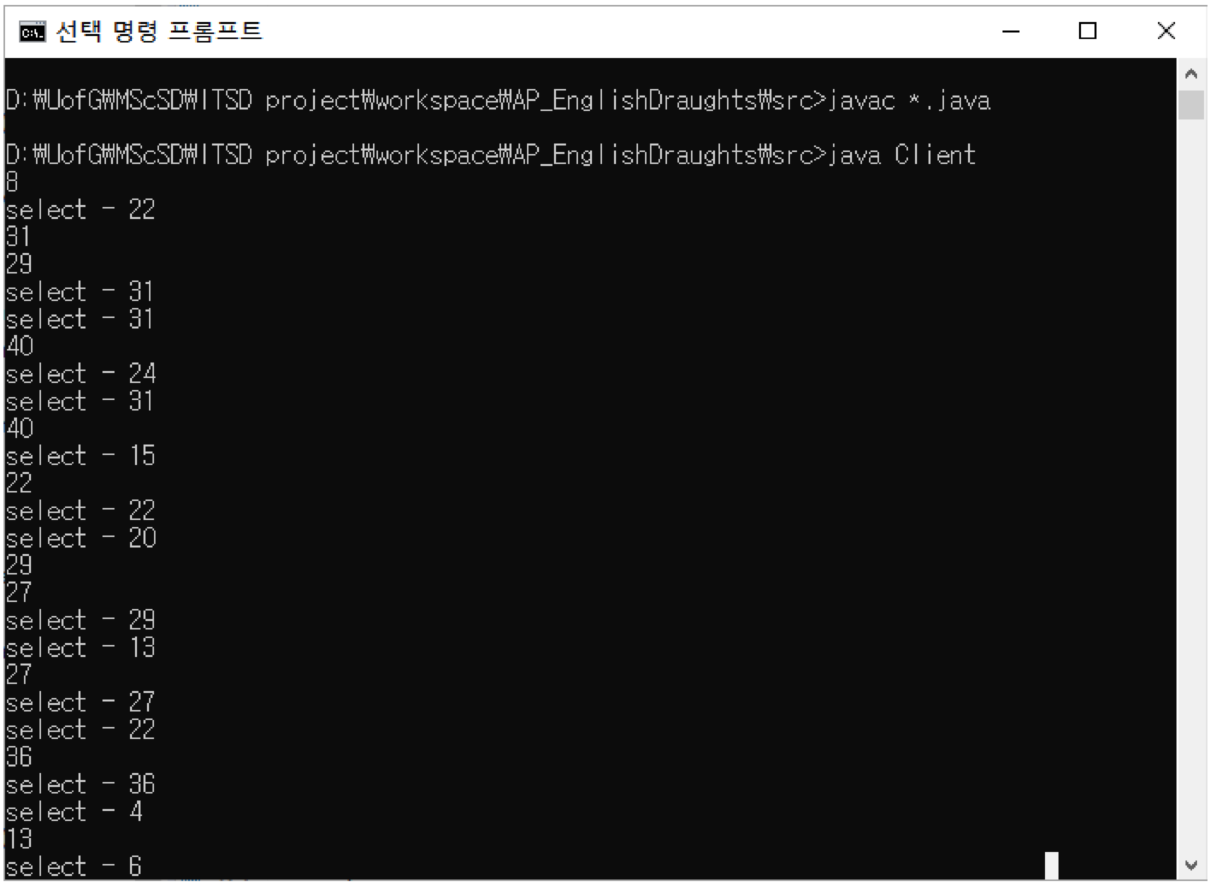


Game is over



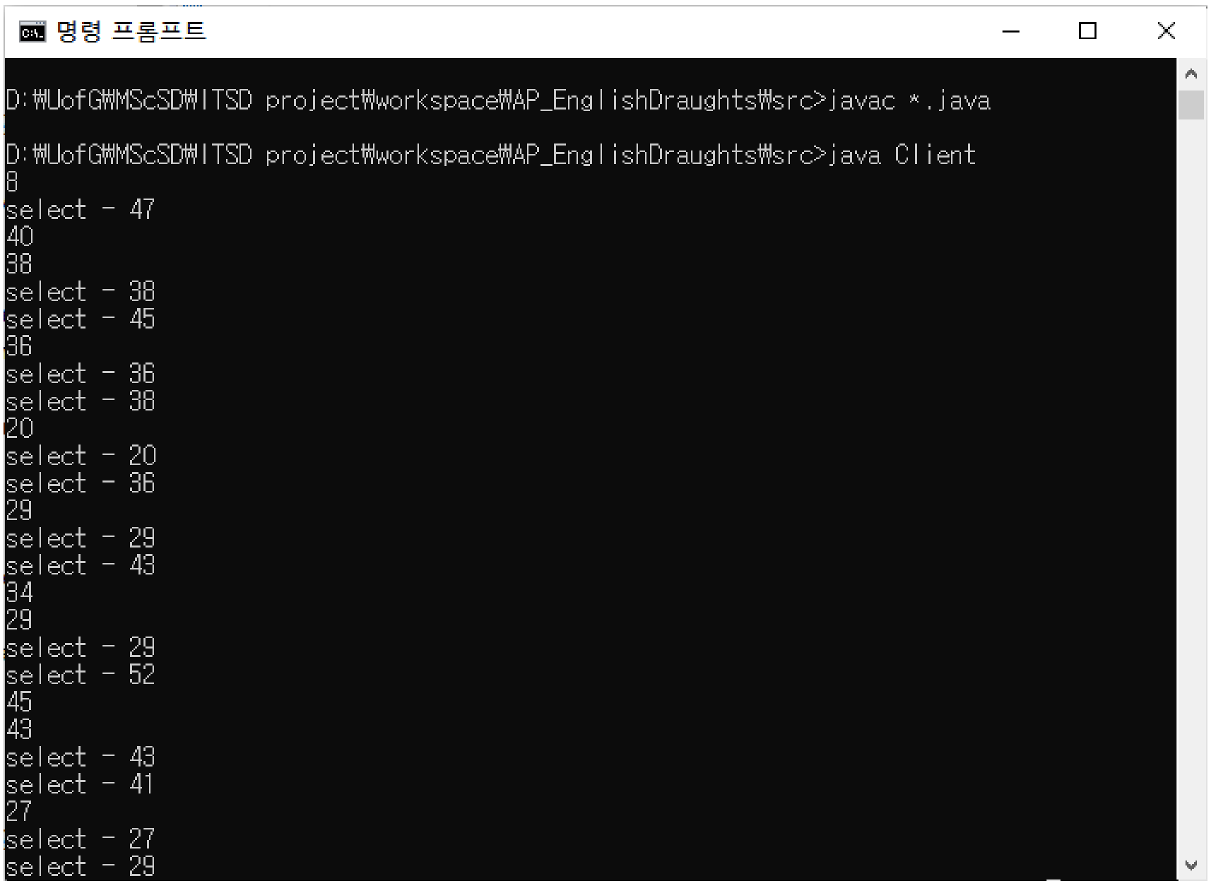
```
D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>javac *.java

D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>java Server
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
after break
second break
```



```
D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>javac *.java

D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>java Client
8
select - 22
31
29
select - 31
select - 31
40
select - 24
select - 31
40
select - 15
22
select - 22
select - 20
29
27
select - 29
select - 13
27
select - 27
select - 22
36
select - 36
select - 4
13
select - 6
```



```
D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>javac *.java

D:₩UofG₩MScSD₩ITSD project₩workspace₩AP_EnglishDraughts₩src>java Client
8
select - 47
40
38
select - 38
select - 45
36
select - 36
select - 38
20
select - 20
select - 36
29
select - 29
select - 43
34
29
select - 29
select - 52
45
43
select - 43
select - 41
27
select - 27
select - 29
```