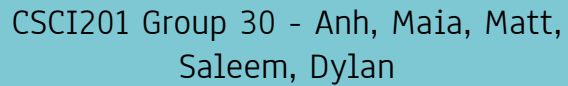





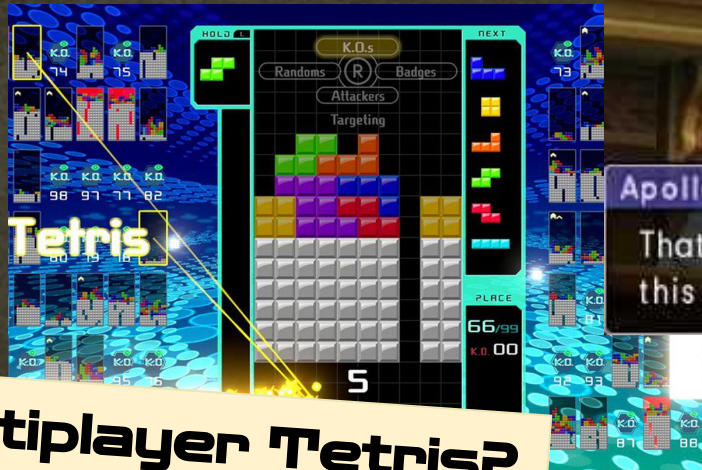
Connect Four



CSCI201 Group 30 - Anh, Maia, Matt,
Saleem, Dylan



Original Ideas

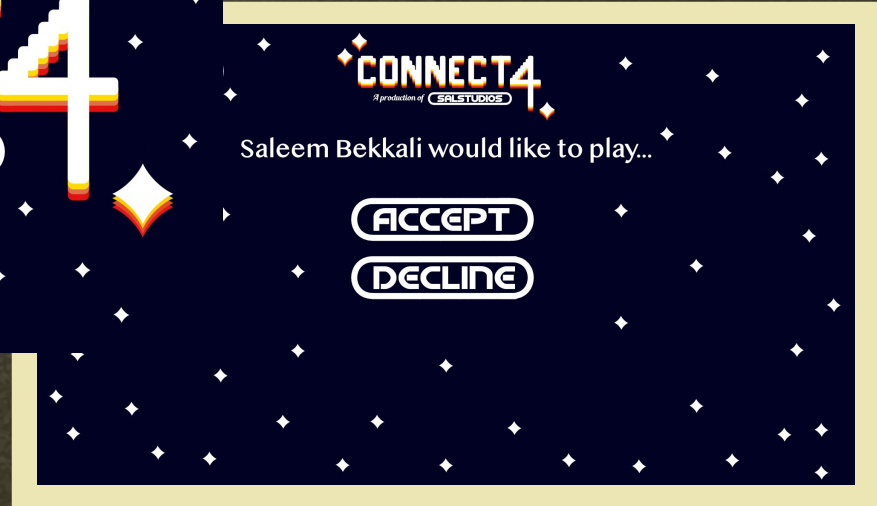


Multiplayer Tetris?



Visual Novel?

Project Design Process - Brainstorming



Software & Tools Used

- Eclipse, Java
- JDBC
- JavaAWT
- JavaFX (attempted)
- Java Message Digest & byte -> hexadecimal conversion (hashing)
- SQL, MySQL
- EGit



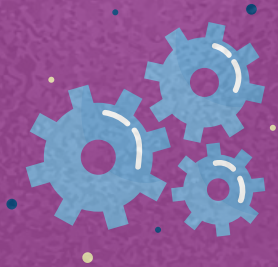
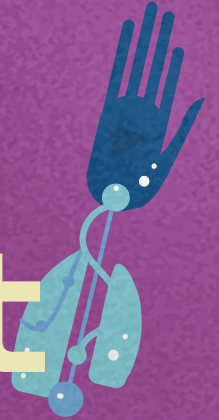
Topics Used

- Game Design
- UI Design
- Database Design
- Passcode Hashing
- Version control
- Multithreading
- Networking

Data Structures

- CopyOnWriteArrayList
- ConcurrentHashMap
- 2D CopyOnWriteArrayList
- Key Set from ConcurrentHashMap (Concurrent Set)
- SQL Database

Final Project Design

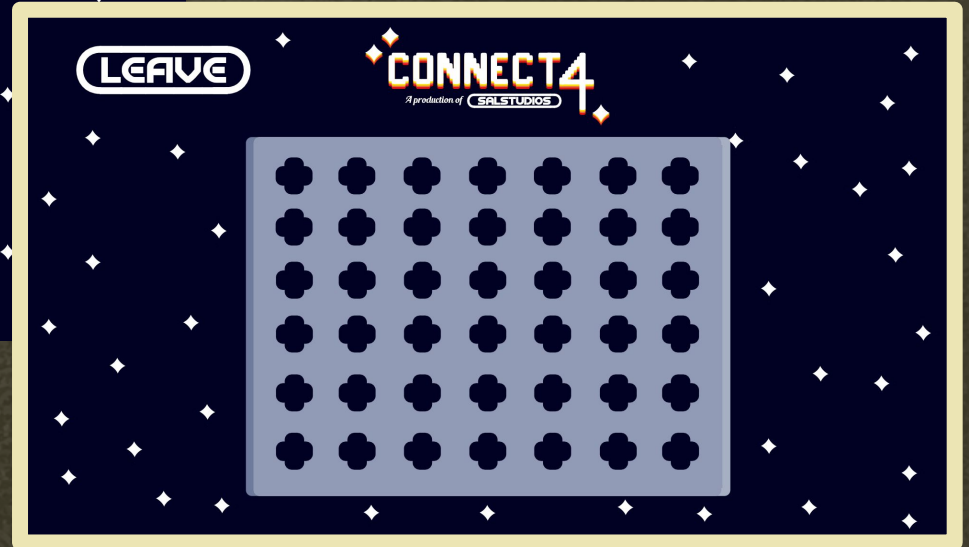
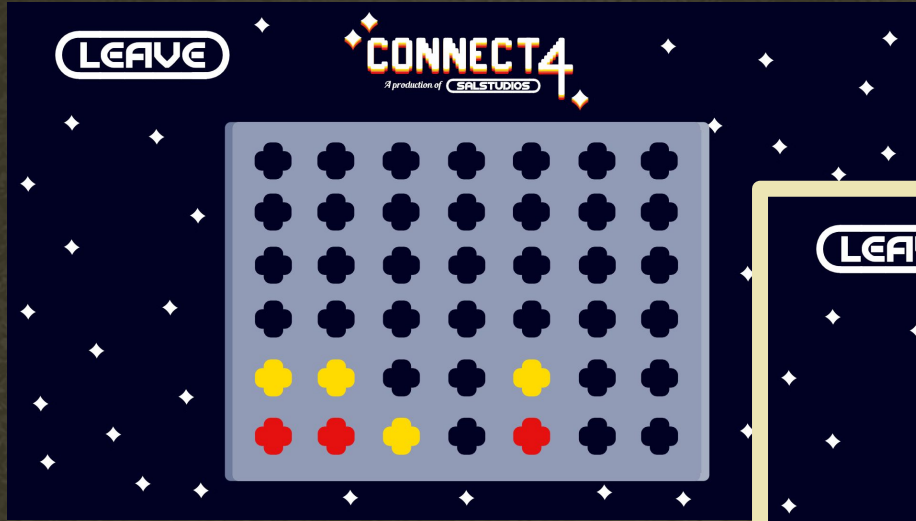




Classes

- GameBoard
- Player
- Servermain
- Clientmain
- FrontEnd

Game Board and Logic



Multithreaded Player Class

```
private synchronized boolean invite(String user) throws IOException {
    if (inGame) {
        return false;
    }
    else if (inQueue) {
        inQueue = false;
    }
    write("invite");
    write(user);
    String response = Servermain.readInput(input);
    if (response.equals("accept")) {
        inviteFlag = true;
        return true;
    }
    else if (response.equals("quit")) {
        shutdown();
        quitFlag = true;
        return false;
    }
    return false;
}
```

Provides functionality for
actually playing the game and
processes invites between
registered players

```
private boolean takeTurn() throws IOException {
    String col;
    Integer valid;
    do {
        write("move");
        col = Servermain.readInput(input);

        if (col.equals("forfeit")) {
            opponent.relay("forfeit");
            return false;
        }
        else if (col.equals("quit")) {
            opponent.relay("forfeit");
            quitFlag = true;
            return false;
        }
        valid = insert(Integer.parseInt(col));
    } while (valid == 0);
    board.print();
    write("success");
    opponent.relay(col);
    return true;
}
```


Networking: ServerMain

```
/* [COMPLETE] Create GUEST player object & add to database & player list.  
 * Write "success" to client, return true.  
 * If DB error, writer "error" to client, return false.  
 */  
public static boolean createGuest(BufferedReader br, PrintWriter pr, Socket s) {  
  
    try {  
  
        Random rng = new Random(0);  
        String user = "Guest#" + Integer.toString(rng.nextInt(Integer.MAX_VALUE)%10000);  
  
        while(users.containsKey(user)) { //if we already have this username, re-randomize  
            user = "Guest#" + Integer.toString(rng.nextInt(Integer.MAX_VALUE)%10000);  
        }  
  
        st = conn.prepareStatement("INSERT INTO c4players VALUES(0,?,null,0)");  
        st.setString(1, user);  
        int success = st.executeUpdate();  
    }  
}
```

Creates & stores all player threads
Queries & Updates SQL Database

Networking: ClientMain

Pulls data from Player class
and allows player to
connect to the server and
play

```
public void contactPlayer() throws IOException {  
    System.out.println("Please choose what you would like to do today:\n"  
        + "1) \"play\" to play a random opponent or check for invites\n"  
        + "2) \"find\" to search for an opponent to play\n");  
    String input = scanner.nextLine().toLowerCase().trim();  
    isQuit(input);  
    socketOutput.println(input);  
    String response;  
    try {  
        switch(input) {  
            case "play":  
                response = readInput(socketInput);  
                switch (response) {  
                    case "timeout":  
                        System.out.println("Timeout error. Please try again.");  
                        contactPlayer();  
                        break;  
                    case "match":  
                        opponent = readInput(socketInput);  
                        System.out.println("Match found with opponent: " + opponent);  
                        break;  
                }  
            }  
        }  
    }
```




JavaAWT

The struggle, and the pivot to JavaFX



What Worked and What Didn't?



Thank
you!