The Github: https://github.com/GlitchyDev/ITSE_GameProject
Some Website Info: http://iproject.com.ng/computer-science/project-topics.html?page=1

https://stackoverflow.com/questions/37648269/how-to-improve-performance-of-javafx-graphic-drawing
4.1 (Induction & Sets)

Minimum Requirements:
- Light Switch off
- Haunted Skull pathfinder player lose
- Chunk Ticking and Unloading
- UI cleanup
- Main Menu Sprites
- Login/PlayerSprite Customise]

SQL Stuff:
- Client (AKA game) doesn't actually touch the table
- Client Username-String

Things to do:
- 1: Add in Haunted_Skull (4 Card directions, 4 smoothing, chase after player, deal 5: damage on jump
- 2: Player Implementation
- 3: Saving System
- 4: Add a method for Chunks to purposefully unload after being out of range for x seconds or a massive distance of 1000 blocks (Instant teleport)
- 5: Chunk fix for entities trying to exit chunk during chunk tick (Add to new chunk, add to Remove Later" list for current ticking chunk
- 6: Networking Scaling
- 0: Finish Lighting: **Lighting Spread is done, custom weights are needed 10/4**
- ~~8: Post Rendering effects: Not feasible~~

Things done:
- Scale up Sprites automagically when importing, on choice by sprite: COMPLETED
- Add in EntityDistance: COMPLETED

Concept Idea
- And endless "Dungeon" is the world, a shambling wasteland of forgotten ruins, caves, and deadly secrets.
- "Miners", people who equip a similar Yellow Jackets and Goggles and go deep into the "Dungeon" to grab necessary supplies for their town, fighting off people twisted by the dungeon and other dark things.

- You play as [Pro] (Protagonist) a Miner sent out into the dark, can you survive the infinite dark?
    - You can use the “Alert” feature to notify nearby “Miners” that you need help in multiplayer, however you will also bring the twisted right to your door!
    - Inventory is based “Capacity Size”, you can carry as many items as you want, however each item has a “Space Size”, which fills up your bag slowly

Broad goals for completion:

- Create Framework for Saving and Loading*
    - Need to be-able to load either the entire world, or specified chunks
    - Needs to be run before Singleplayer, or on Server Creation
- Create framework for terrain generation* COMPLETED 9/15/2017
- Add more Entity and Block Functions
    - Health COMPLETED 9/13/2017
    - Damage COMPLETED 9/13/2017
    - Enter/Exit/Interact COMPLETED 9/15/2017
- Add some Networking Capabilities
    - This involved writing each Logic to check for what Network role the Program has at that moment
- Add some AI
    - Pathfinding AI COMPLETED 9/23/2017
- Controller Input Lag (for executing combos)

SQL Table

- Account Information
- Username [Key]
    - UUID (String)
    - Password (String)
    - Skin (String)

World:

- Attributes
    - Name of world
    - Players
        - Devotedworker: Last seen 1,1

- 1,1.txt
    - Stuct

****

Idea: Dungeon Crawler
Required Features:

- Generating Terrain: <Robert|Charlie>
- Code Base <Done|Robert>
- Multiplayer [Robert|Abdullah|Charlie] || Optional
    - Local Hosting (Multiple Controllers)
- Saving (SQL saving) [Abdullah|Charlie]
- Action Tree's (AI) <Robert|Tanner>
- Controller <Robert|Ben>
- Input <Ben|Robert> (Robert Sets up JInput, Ben creates Combo presses)
    - X Input to work
    - Accept Multiple Controllers

NonCoding

- Sprites: [Robert]
- Sounds: [Ben]

Game States:

- Overworld
- Main Menu
- Control Adjusting
- Multiplayer (Player Dedicated Server House)
- Pause (Singleplayer)
- Inventory

What needs to be implemented/Concerns

- Specific "Overall Network Role" which represents
    - Standalone (Single Player?)
    - Client (Single Window Connected to a Server)
    - ListenServer (Server with a Local Player)
    - DedicatedServer (Server with only Command Line)
        - Each (Player) Role comes with the Modifier "Local" for multiple Players
- Multiplayer Concerns
    - Server: TOTAL Authority (Minus Controls)
    - Client: Send inputs (If we do "Combos" wait until Combo is done and send combo attempt)
    - All actions in game need to be tied to a packet
        - Entity Move
        - Entity Animate
        - Entity Remove

- Saving Concerns
    - Saving “Dungeon Chunks” for both Client and Server, and knowing when to remove “Chunks” from active Memory. As it stands generated chunks will remain in memory infinitely

    - Keeping Saving Format “Nice and Easy” so with updates shit don’t break D_D
- For “Local Clients/Local Clients Servers
    - Generate a “Camera” center point between all players
        - Allow the Camera to keep as many players in view as it can
            - For 2 Players, always keep it focused on Player 1
            - For 2+ Players keep it focused on the closest players not out of range
    - Send this to Server or to any classes that depend on something being in view
- Limitations
    - Java sockets don’t connect through Kennesaw's Firewall, Local WIFI is the best one can do under these circumstances

Game Feature Ideas:

- As the Player moves, other creatures move
    - For multiplayer, all players would have to move before anything happened Which might break game flow, unless game flow is already slow
- Turn Based (Slow action, not ideal for Multiplayer unless its built to be slow)
- “Every X Second” action (Like Necrodancer but wouldn’t be very original)
- **Realtime! [Ideal for Multiplayer] (With Cooldowns for movement and attack)* What we did**

System Syntex Setup

- JInput (Polling System, no event based management, sadly its the only one that really works, and it doesn’t rescan for controllers very well, if at all)
    - We will
        - Detect Controllers on startup (For Main Menu Navigation)
            - Have a “Rescan for Controllers” button on Local Multiplayer, and for it to display currently connected controllers
- For LocalClients
    - For “Networked” (Non LocalStandalone) there needs to be a UUID assigned to each player, and each “Socket” can have multiple UUID’s tied to the receiving end, and the UUID of each player need to be sent along with the Packet. For Local Multiplayer a UUID is tied to each Controller Object
- For “Dedicated Server” vs “Standalone/Client/ListenServer”, a Java Run Args can determine which is which or possibly a option on the main Menu