PowerCraft

Yamini Duggal: yaminiduggal@gmail.com

Chiranjeevi Appalarouthu: appalarouthu1997@gmail.com

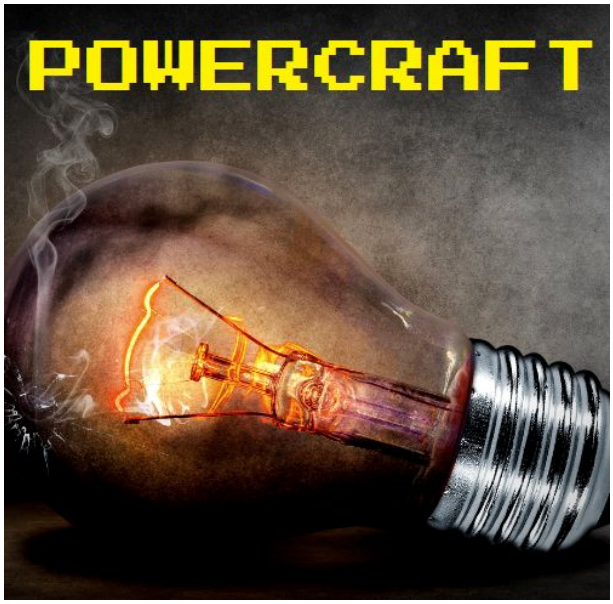


TABLE OF CONTENTS

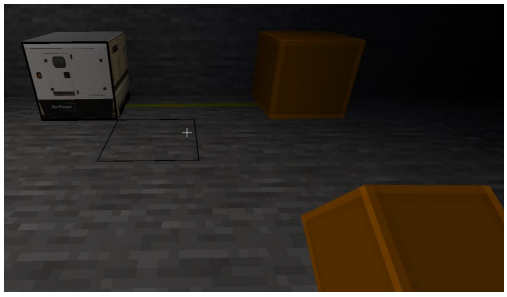
EXECUTIVE SUMMARY

PowerCraft is a tool that simulates accurate electrical power systems. It is used for instructional and research purposes to introduce the basics of electrical power components. PowerCraft is a way for students to experiment with electrical power systems without having to purchase actual devices. The PowerCraft mod currently includes the following components:

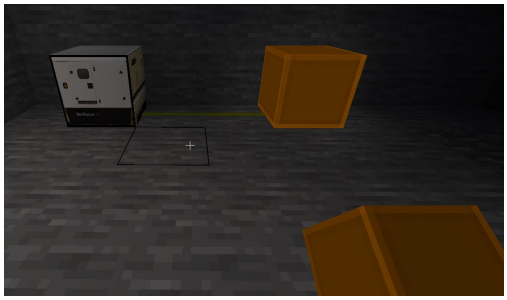
- Generator: power-producing device
- Cable: power transmitting device
- Light: power-consuming device

Summary of test results:

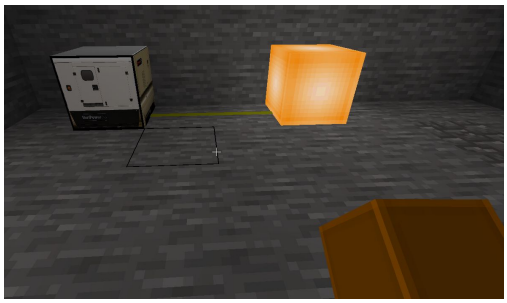
A generator block is connected to a cable block which is connected to a light block.



In this case, the voltage of the generator does not match and exceeds the voltage of the light therefore the light is not glowing.



In this case, the voltage of the light does not match and exceeds the voltage of the generator, therefore the light is on but dim.



In this case, the voltage of the generator matches the voltage of the light, therefore the light is glowing.

BACKGROUND

The sponsor for PowerCraft is Dr. Daniel Conte de Leon. The opportunity associated with this project is to enable the creation of realistic and accurate models of electrical power transmission, distribution, control, and usage scenarios for research and instruction.

Initially, PowerCraft was to be created in Minecraft using components from existing mods. Minecraft is a game that enables the creation of virtual worlds. Mods are used to modify the game and contribute to the virtual worlds. Many mods exist for energy and mechanics, and a couple to simulate electrical systems. However, none of those mods model electrical systems in a realistic manner to enable research and education in electrical power systems.

Ongoing research for this project has uncovered that there is a better option than Minecraft to develop PowerCraft. Currently, PowerCraft is a mod being developed in Minetest without the use of existing mods.

Stakeholders	Benefits
Sponsor → Daniel Conte de Leon	Mentoring experience
Team → Chiranjeevi, Yamini	Project management and technical application experience
Customer → Professors, Students	Free experimentation

PROBLEM DEFINITION

There is no existing free mod that accurately simulates electrical power systems. PowerCraft was initiated to solve this problem.

A high-level overview of the project deliverables:

- Create a realistic power-producing device
    - Generator
- Create a realistic power transmitting device
    - Cable
- Create a realistic power-consuming device
    - Light
- Create GUIs for electrical values (such as voltage) for the generator and light
    - Stored to compare voltages of the different devices
    - Used to calculate current based on user input

PROJECT PLAN

The team roles and responsibilities include:

- Documenting meeting minutes
- Documenting any discoveries
- Meeting with the sponsor and team weekly
    - Allowing continuous feedback resulting in better product development
- Researching and exploring
    - Understanding the logic behind decisions
- Programming
    - Understanding in-built Minetest components
    - Learning Lua
- Presenting the product

Project Tasks:

Number	Task	Progress
1	Determining team communication	Complete
2	Determining meeting schedule and understanding project description	Complete
3	Obtaining Minecraft license + installation	Complete
4	Exploring Minecraft	Complete
5	Exploring Minecraft mods	Complete
6	Brainstorming implementation	Complete
7	Electrical system protocol research	Complete
8	Finalizing implementation strategy (basics)	Complete
9	Mod code exploration	Complete
10	Running Doxygen	Complete
11	Generating Documentation	Complete
12	Generating Class Diagram	Complete
13	Exploring OpenComputers and ElectricalAge mod	Complete
14	Exploring Minetest	Complete

15	Creating a mod in Minetest	Complete
16	Creating a block in Minetest	Complete
17	Deleting a block in Minetest	Complete
18	Creating a generator	Complete
19	Creating a consumer (light)	Complete
20	Creating a cable wire	Complete
21	Creating a GUI for the generator	Complete
22	Creating a GUI for the light	Complete

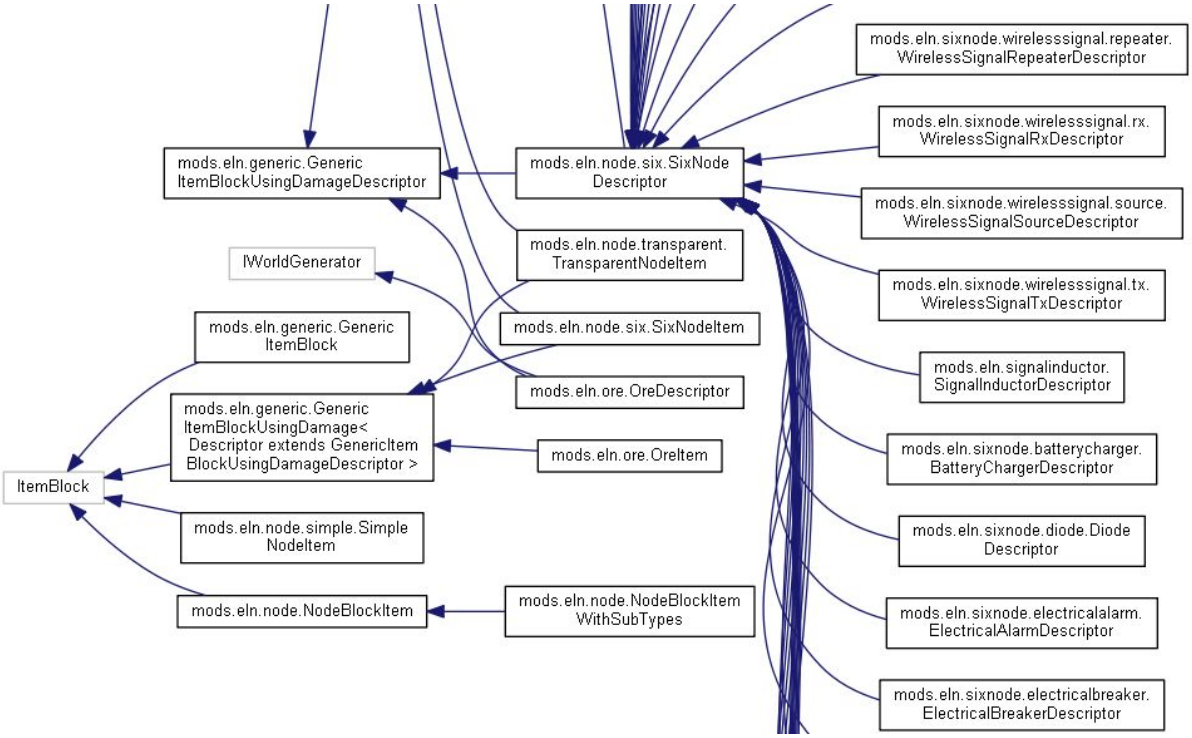
CONCEPTS

Considerations

Minecraft:

1. Download Minecraft
2. Download electrical age and open computers mod
3. Experiment with the different components of those mods
4. Look at the code for the different classes for those mods
5. Understand the functionalities (compared to real-world power systems)
6. Modify the mod (too complex to do this step)

This is a snippet of the structure for a class, ItemBlock, in the ElectricalAge mod. There were many such complex classes with undocumented code. Understanding how all the different classes and components are linked to one another would have taken too much time.



Terasology:

1. Download terasology
2. Research about existing electrical mods
3. Create a mod

Not many resources are available online for development in Terasology.

Selection

Minetest:

1. Download minetest
2. Research about existing electrical mods
3. Download mesecons (includes electrical components such as power plant, light, cable (not realistic))
4. Understand realistic electrical power systems
5. Model realistic components by creating a class diagram
6. Create a mod
7. Develop a Lua file for each new component (generator, cable, light)
8. Test the components and GUIs

Reasons for selecting Minetest:

- Hard to understand and reverse engineer the Minecraft mod code
- Difficult to integrate existing mod code with new code
- Starting from scratch will be more clear and straightforward

SYSTEM ARCHITECTURE

There is a generator class that contains a current variable which is calculated in the backend and a genVoltage variable which is based on the user input for the voltage of that generator and stored in a hash table with the key as the position (assigned using the X-Y-Z coordinates) of the respective generator block and the value being the inputted voltage.

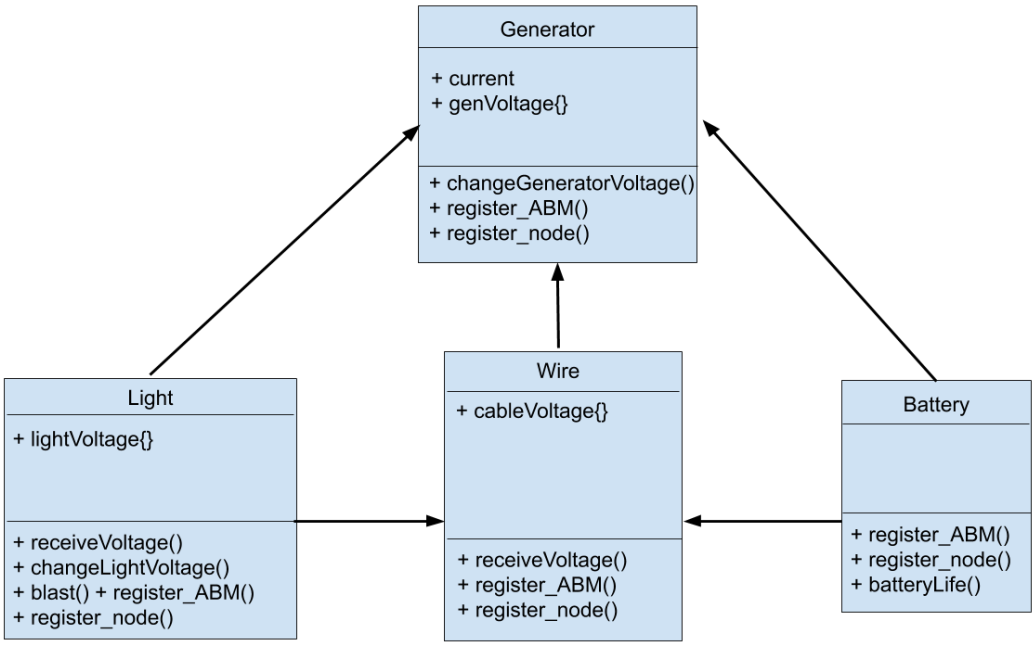
There is a wire class that inherits values such as current and voltage from the generator class in order to transmit it to the light.

There is a light class that inherits values such as current and voltage from the generator class to check whether or not the voltage of the light matches the voltage of the generator. It contains a lightVoltage variable which is based on the user input for the voltage of that light and stored in a hash table with the key as the position (assigned using the X-Y-Z coordinates) of the respective light block and the value being the inputted voltage.

The battery class will be implemented in the future. It inherits the voltage and current from the generator and wire class and keeps track of the remaining current.

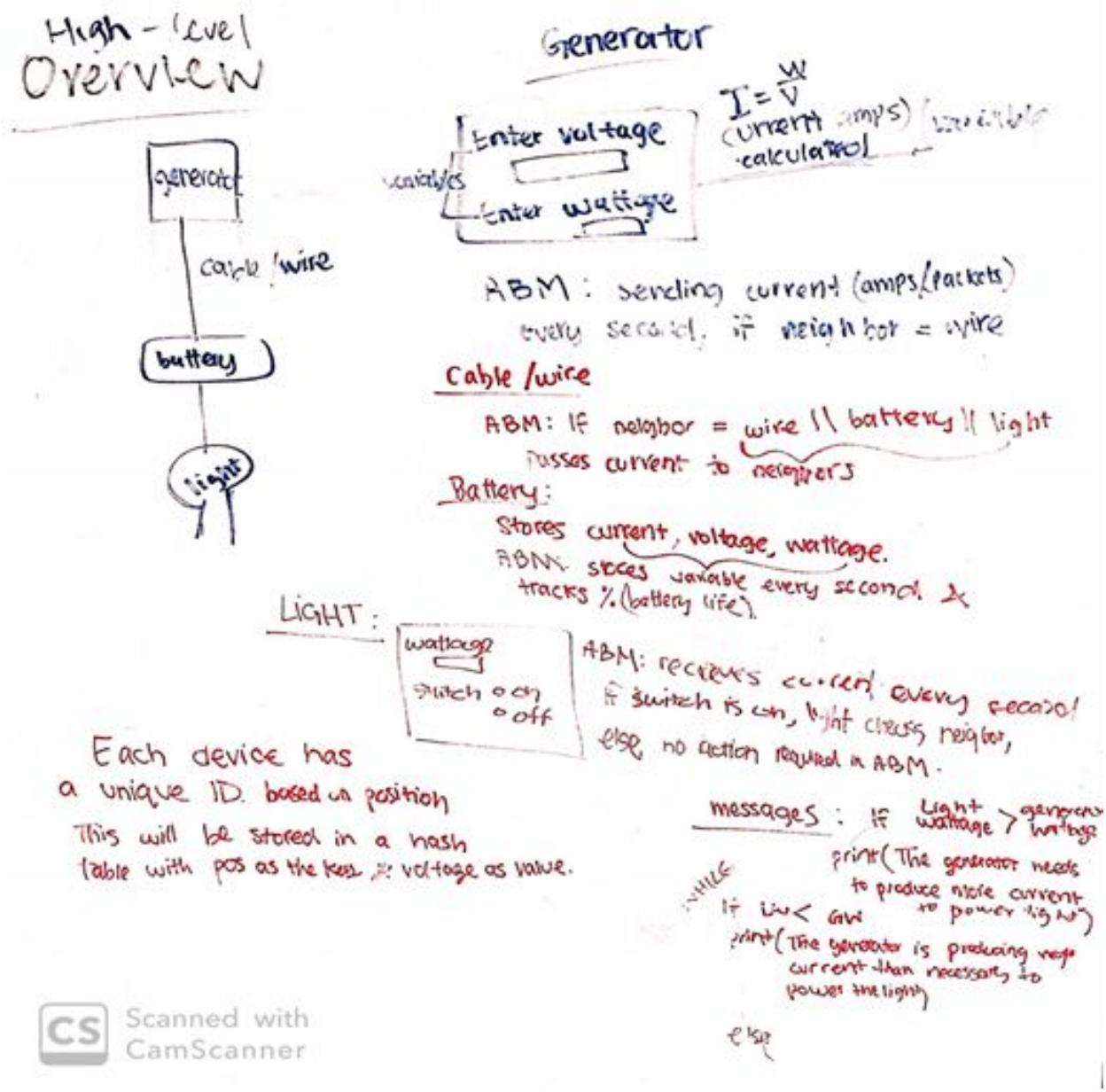
Each class has an Active Block Modifier (ABM) function which checks whether a block is connected to the appropriate block and based on that, transfers the values and determines whether or not the light will glow.

<u>Class Diagram</u>



DESIGN EVALUATION

Analyzing the electrical concepts, flow of the program, and user-side display for the tool.



FUTURE WORK

1. Distributing current
    a. Adding 2 or more generators
    b. Adding 2 or more lights
2. Implement a battery to keep track of the current per second
3. Implement new nodes as power-consuming devices (i.e. light, toaster, fan, etc)

APPENDICES

Project Tasks and Schedule:

https://docs.google.com/spreadsheets/d/1QfW50gxsSqs0vQPZQllp6JdAHNMIzpHoxQI1UJVS4J4/edit?usp=sharing

DFMEA Worksheet:

https://docs.google.com/document/d/1IEl0eBuIXowOa6ESaHF581pVlKUr5KqWAzXG5XmOTPw/edit?usp=sharing

PowerCraft Google Drive Folder:

https://drive.google.com/drive/folders/1UrrutJn-0ptW0uJ9E9Ohs5tF9NRCrV1d?usp=sharing

PowerCraft Github Link:

https://github.com/Chiranjeevi97/PowerCraft

Demo:

https://drive.google.com/open?id=1bB0EULerB8wOQreXjX5-ng055XPaDf_s