GAMING WIZARD

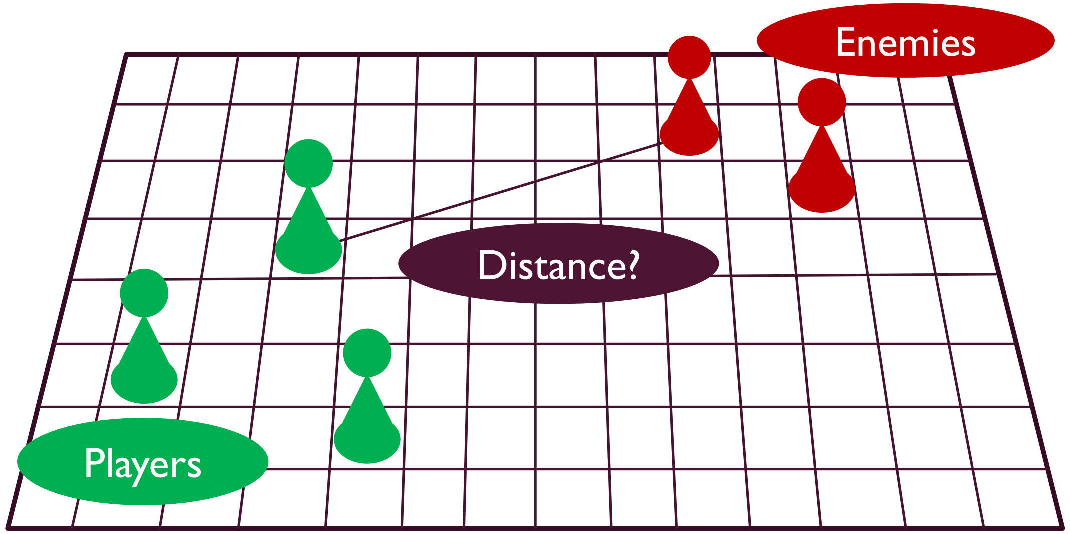
A SMART TABLETOP GAMING SYSTEM

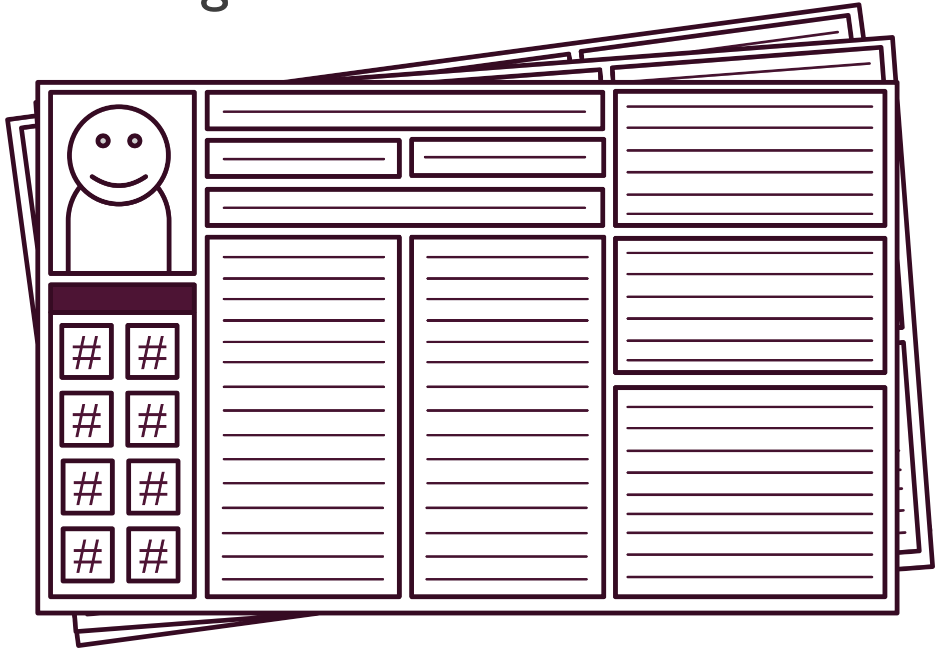
Group 30

Gabriel Holguin	-	Computer Engineering
Daniel Kalley	-	Computer Engineering
Erica Lindbeck	-	Electrical Engineering
Logan Taylor	-	Electrical Engineering

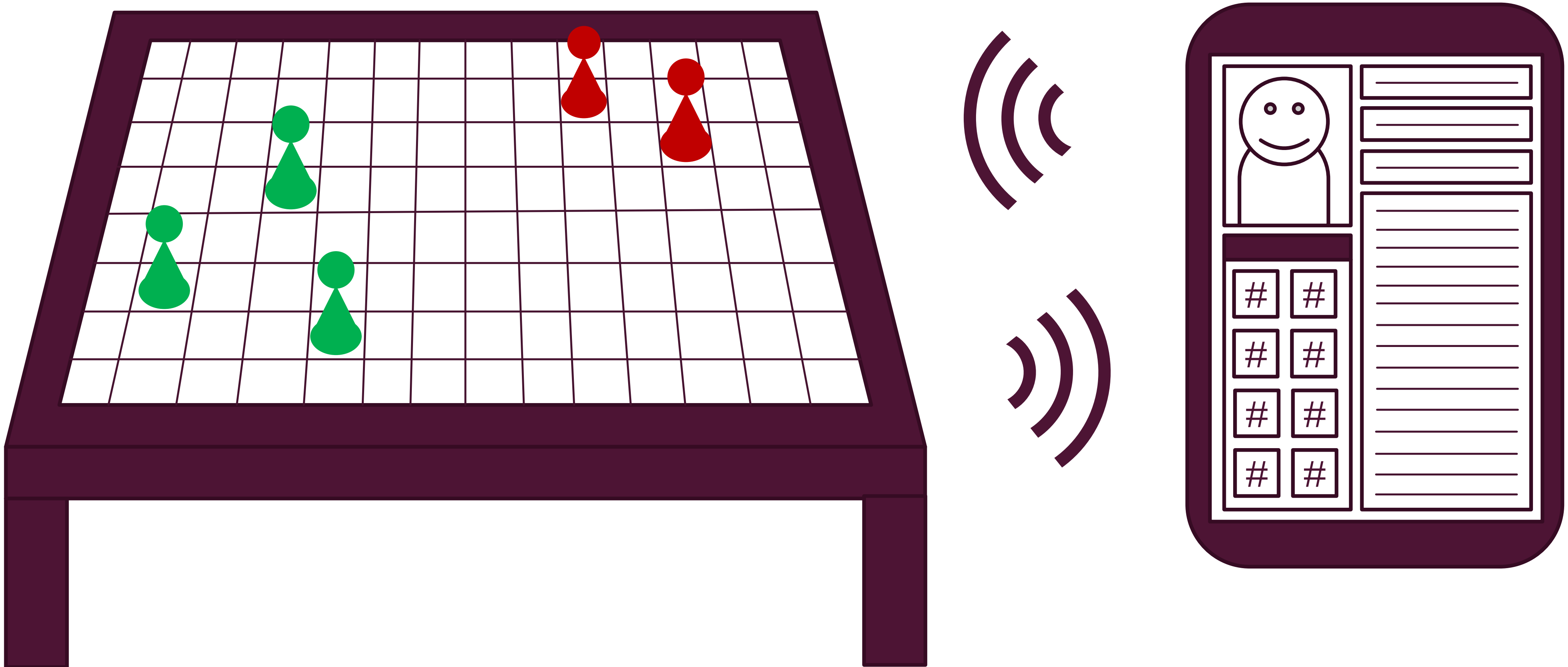
MOTIVATION

- Tabletop roleplaying games are experiencing a resurgence in popularity
- Time is wasted in setup, information lookup, and calculations
- Keeping track of abilities can be complex and confusing





PROPOSED SOLUTION



ENGINEERING REQUIREMENTS

Requirement	Value
Screen Size	20”-36” per side
Table Height	≤ 3’6”
Display Resolution	≥ 720 px / side
Object Size for Detection	≥ 0.5”
Operating Temperature Inside Table	≤ 32 °C
Time to Cool on Start	≤ 15 minutes
Continuous Operation	≥ 6 hours
Device Lifetime	≥ 3 years
People Required to Move Table	≤ 2

Requirement	Value
Simultaneous Touch Inputs	≥ 6
Touch Input Delay	≤ 200 ms
Simultaneous Mobile Controllers	≥ 5
Mobile Controller Input Delay	≤ 1000 ms
Mobile Controller Range	≥ 10’
Object Location Accuracy	± 0.5”
Locations Saved on Exit	≥ 20
Average Time to Set Up Map	≤ 2 minutes
Characters Saved in App	≥ 4

OTHER CONSTRAINTS

Ethical Concerns

- No data is being stored in online servers, since all game and app data is stored locally on a PC and Android device respectively

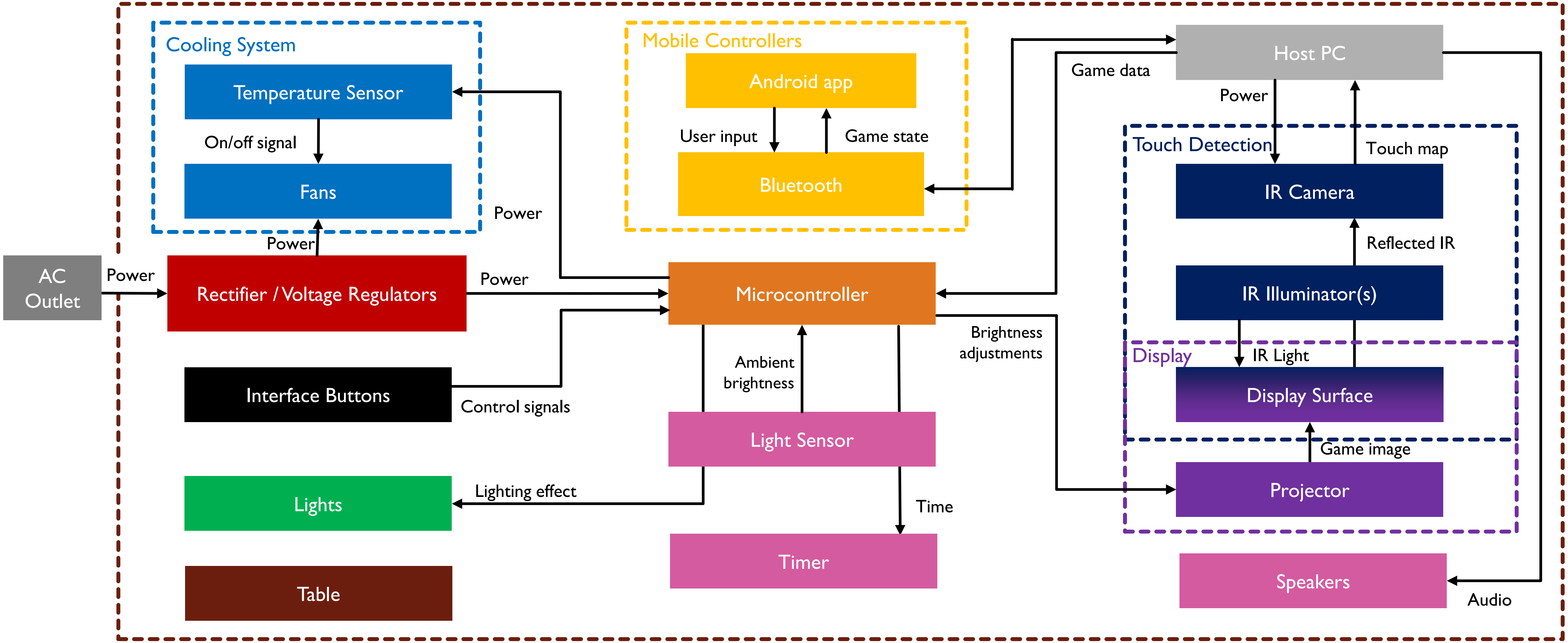
Social Considerations

- Viability of potential users, since product is currently costly and only supports Android OS
- Need to make sure not to infringe on copyright, especially on game material

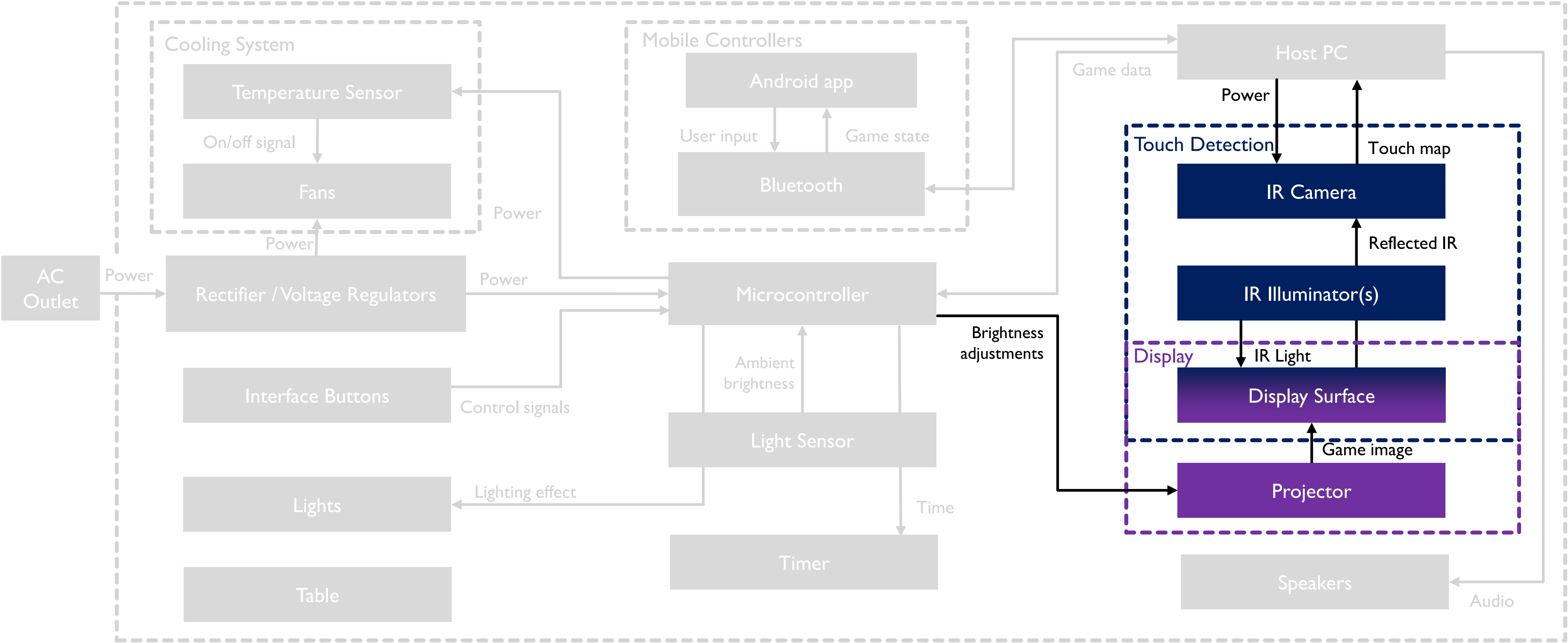
Electronic Safety

- Soldering should be done with a clear head and proper safety equipment on hand.
- Electronics should be in an enclosed environment so users can’t injure themselves with wires and circuit boards
- Projector should always be off when in movement
- Volume and light limits for what players deal with

PROPOSED SOLUTION – HARDWARE



HARDWARE - TOUCHSCREEN



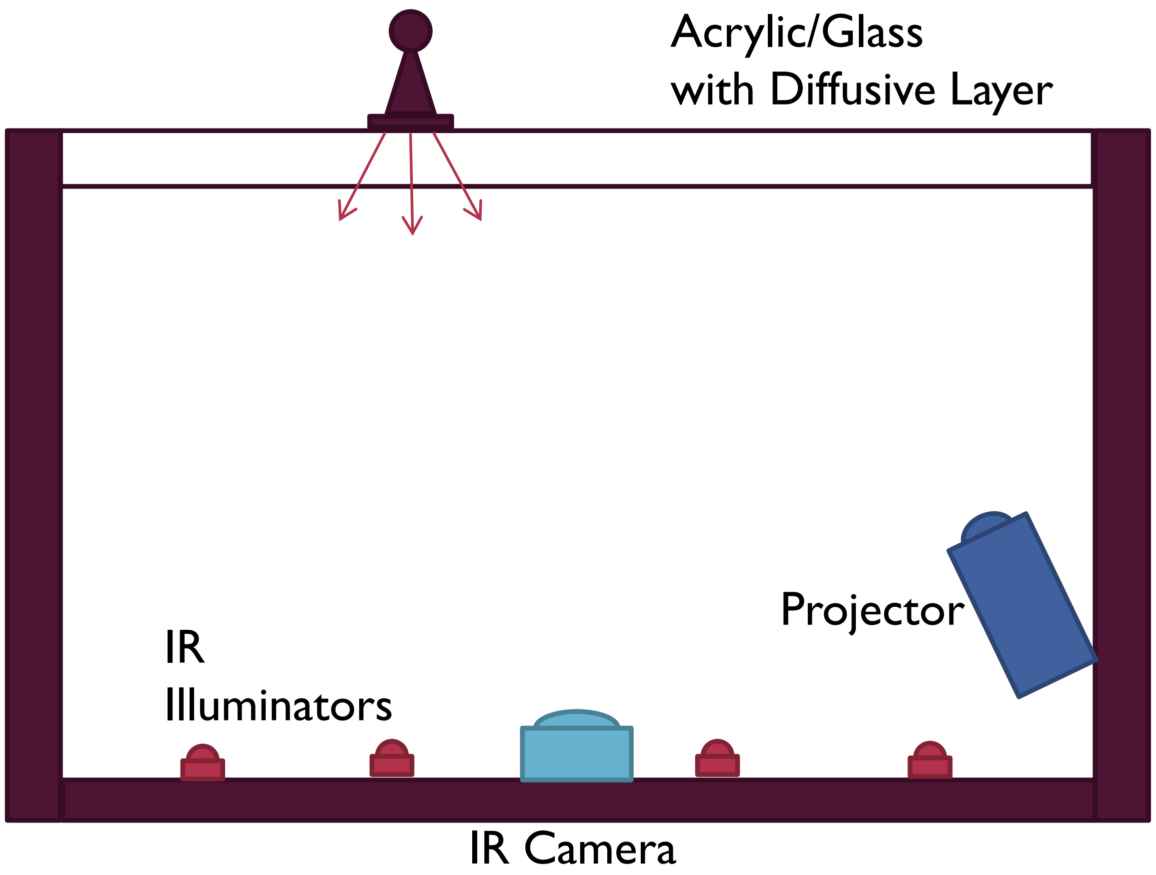
TOUCHSCREEN - METHODS

Requirements

- Distinguish locations on 1" square grid
- Detect objects as well as users' fingers
- $\geq$ 24" per side

Rear Diffused Illumination (Rear DI)

- Advantages: No limit on simultaneous touches, inaccuracy due to camera resolution is small, difficult to illuminate evenly, IR illuminators restrict space for other components
- Disadvantages: Closed box and projector required, difficult to scale



TOUCHSCREEN - COMPONENTS

Projector – Used BenQ MX810ST

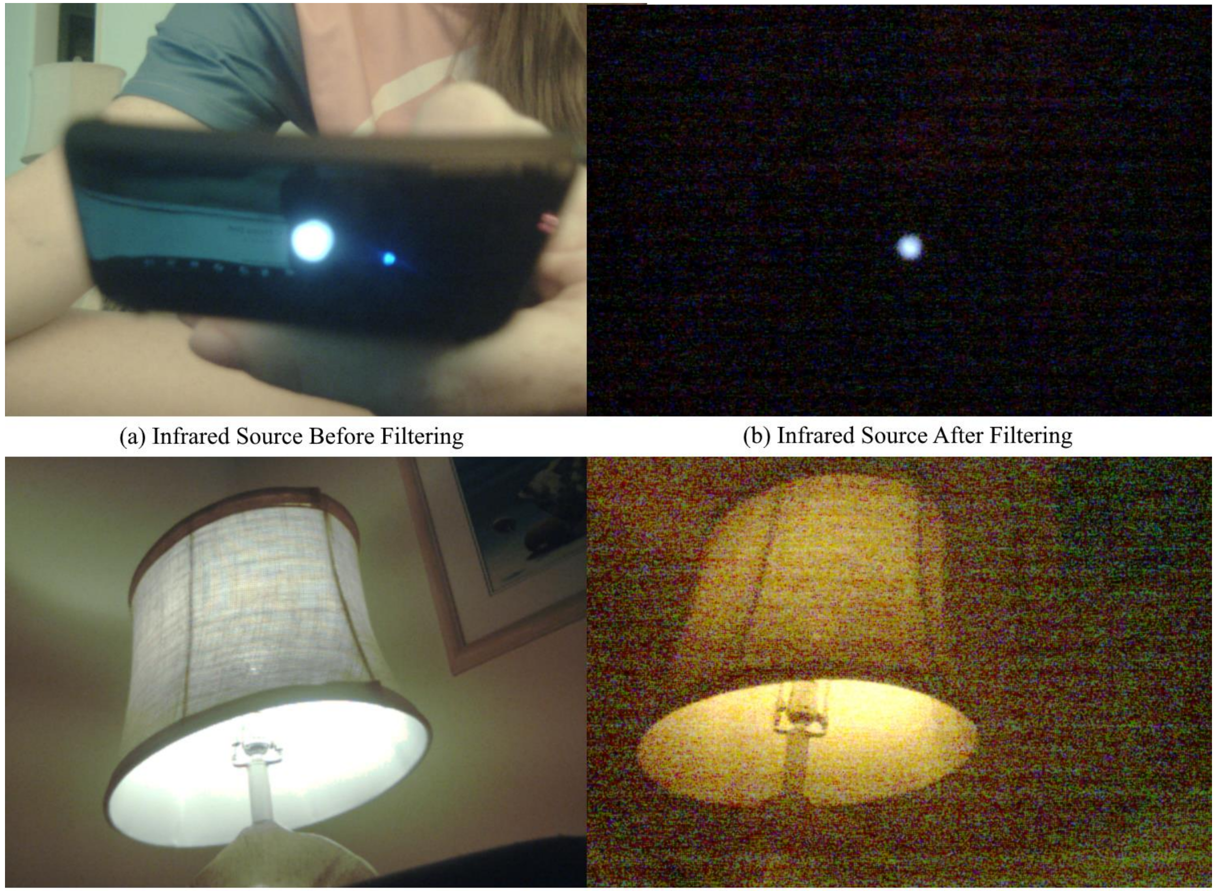
- 1024 x 768 Native Resolution (4:3 Aspect Ratio)
- 0.6 Throw Ratio
  (20" throw distance required for 24" x 32" display)
- 2500 ANSI Lumens
- 4600:1 Contrast Ratio
- 3500 – 5000 hour lamp life (> 1500 remaining)
- Remote and USB control options
- Auto and Manual Vertical Keystoning up to ±30°

Picture TBD

TOUCHSCREEN - COMPONENTS

Camera – PlayStation Eye + Floppy Disk

- 640 x 480
- 60 fps
- USB 2.0 connection
- PC driver available for $3
- Explicitly supported by existing open-source touch detection software
- Removable infrared-blocking filter
- Floppy disk material can be used as cheap visible light-blocking filter



(a) Infrared Source Before Filtering

(b) Infrared Source After Filtering

(c) Incandescent Light Before Filtering

(d) Incandescent Light After Filtering

TOUCHSCREEN - COMPONENTS

Screen Materials

- Cast Acrylic
  - 24” X 32” X 0.25”
    - (25.5" X 33.375" X 0.25” after completion of table)
    - Chosen for strength and rigidity
- Drafting Film
  - 0.003” and 0.005” in testing
    - Chosen for cost, ease of modification, and durability
    - Trade-off between blob contrast and image visibility

Illumination

- Tendelux 80 ft IR Illuminator
  - 850 nm wavelength
  - Intended for use with night vision security cameras
  - Comes with power supply and mounting mechanism
  - Designed to eliminate hot spots

HARDWARE - TABLE

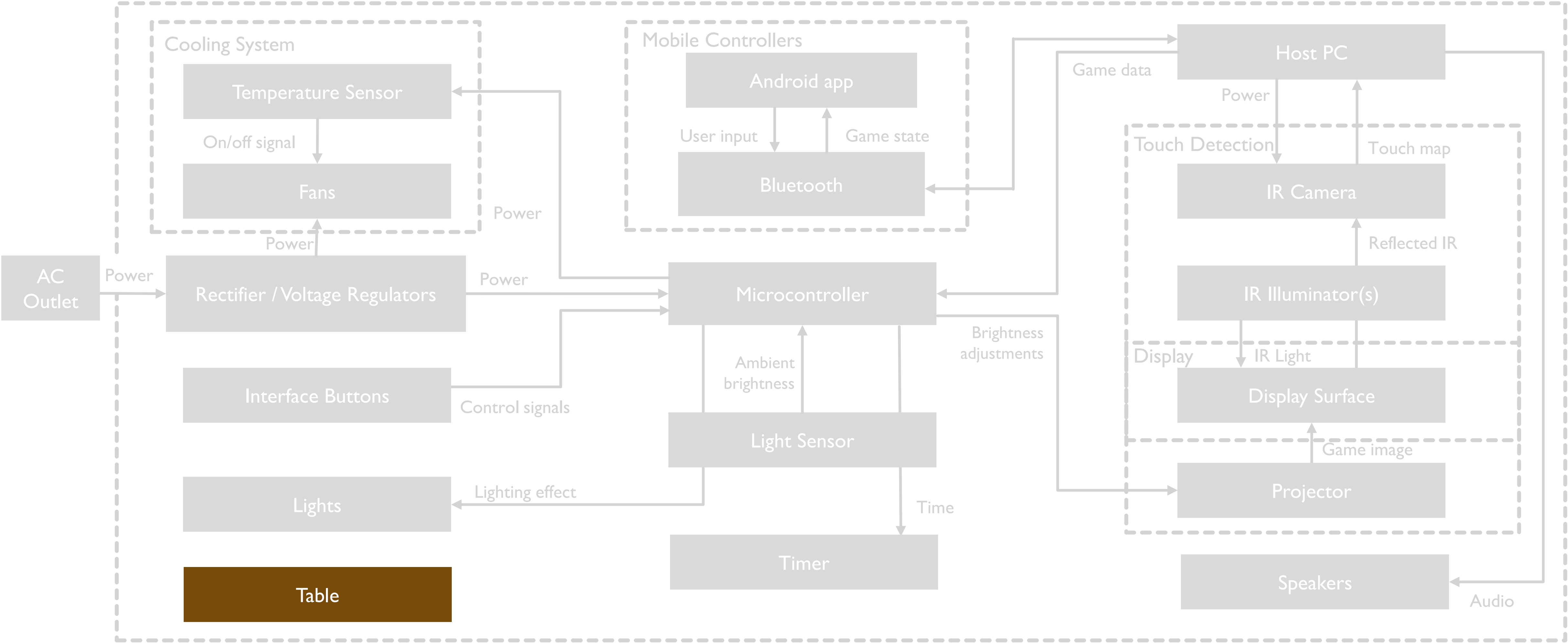


TABLE DESIGN

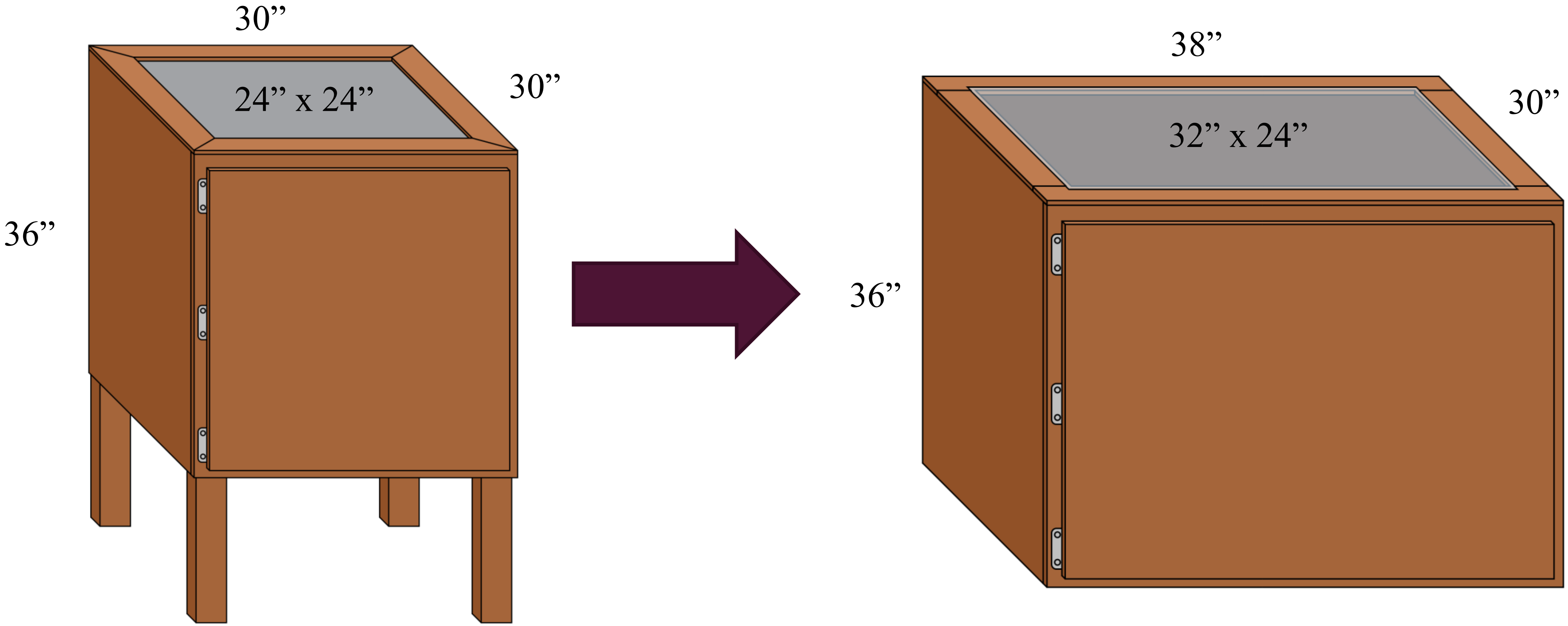
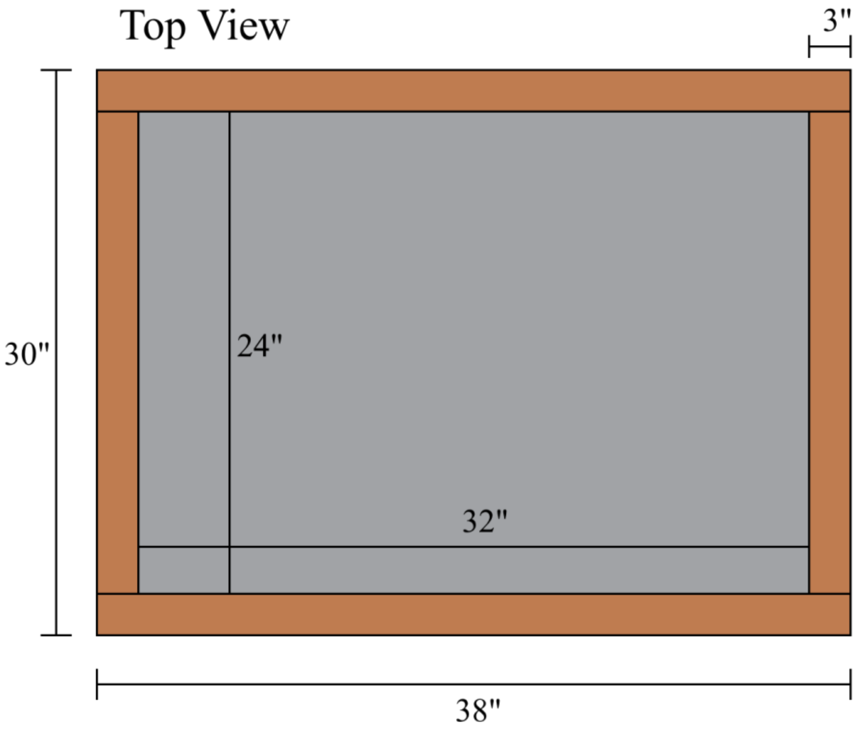
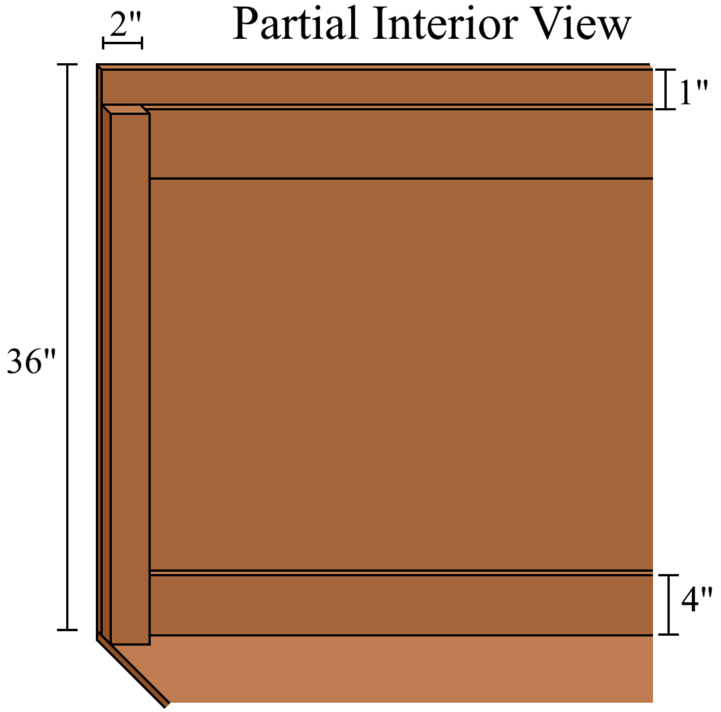


TABLE DESIGN

Top View



Partial Interior View



Front Door View

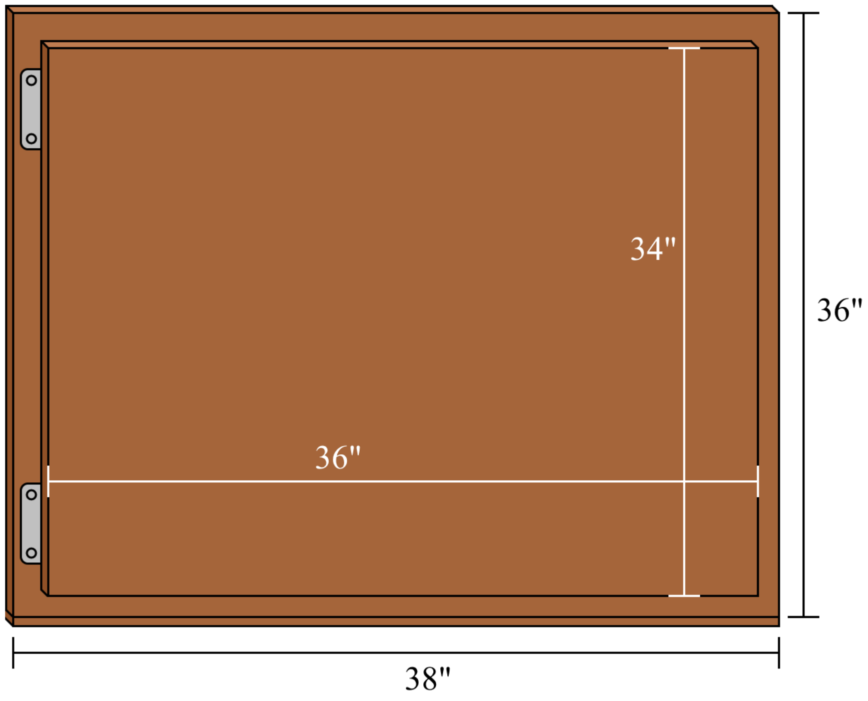
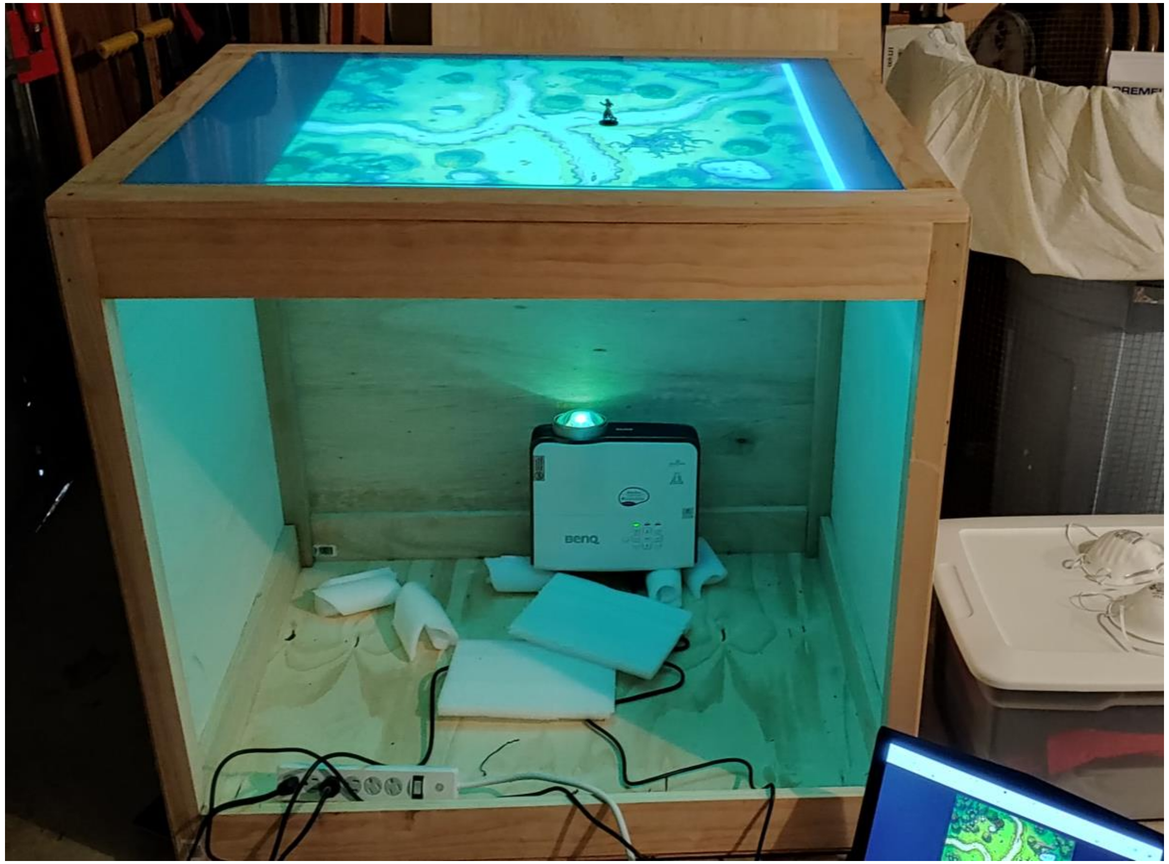
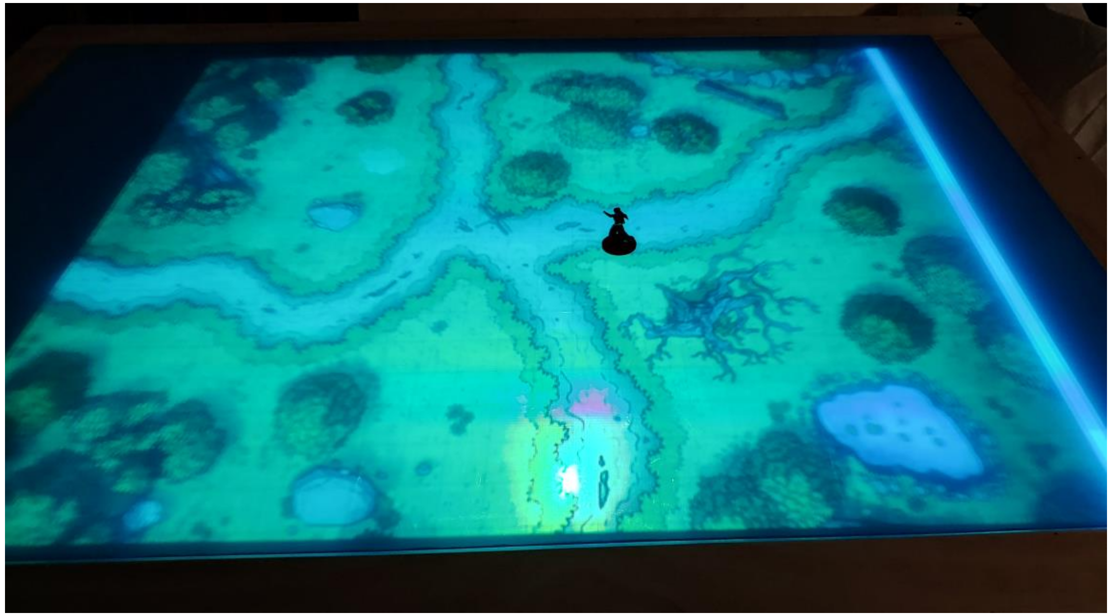
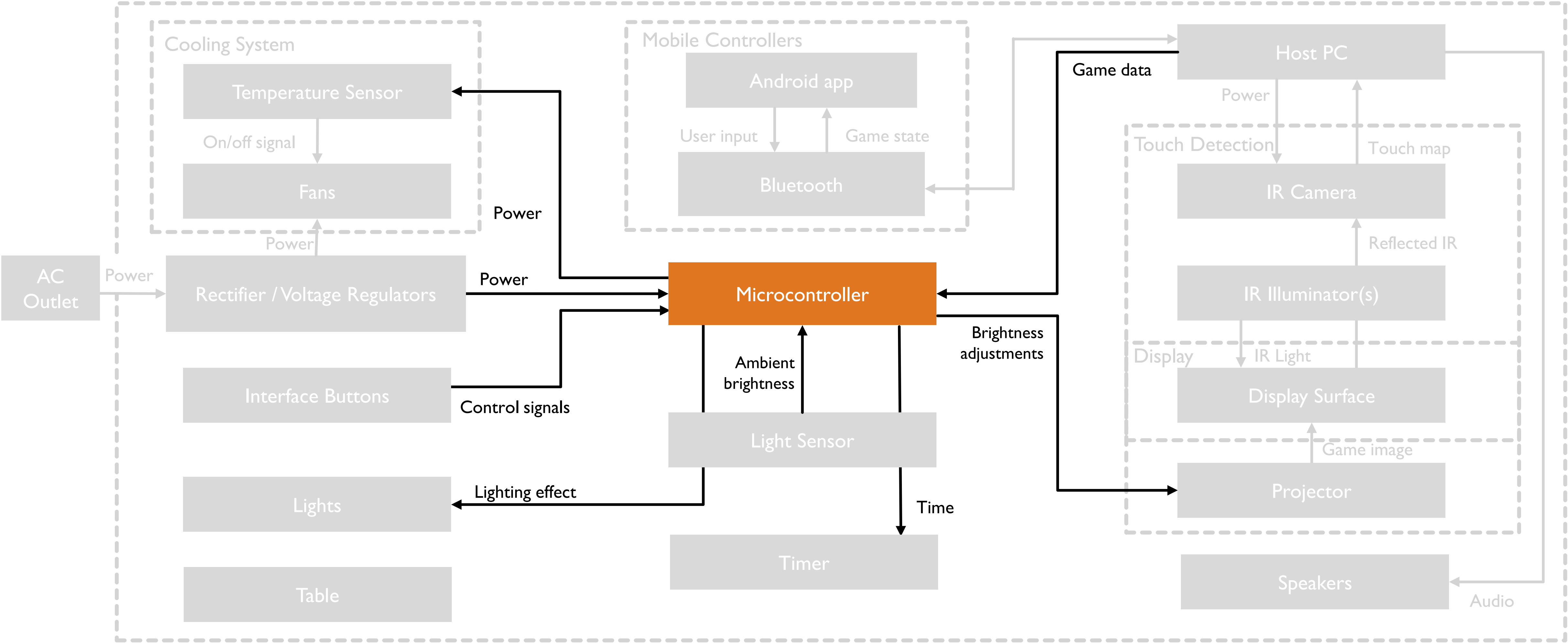


TABLE AND TOUCH SCREEN PROGRESS





HARDWARE - MICROCONTROLLER



MCU – MCU SELECTION

Requirements

- Enough I/O lines for timer, LED drivers, temperature sensor and communication.
- Enough memory for bootloader and software.

Desired

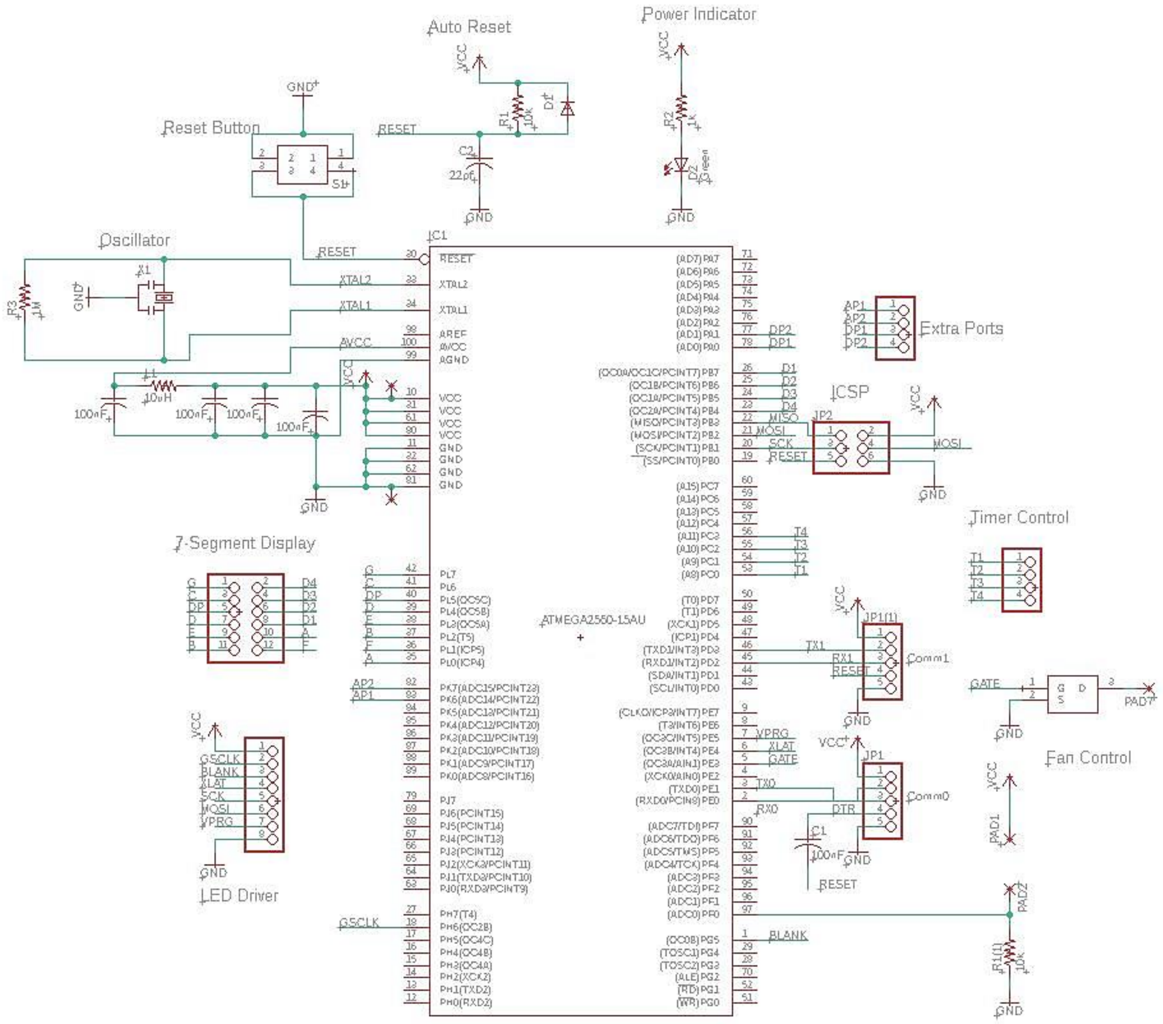
- Large amount of documentation
- Familiar
- Affordable

	ATmega328P	MSP430FR6989	ATmega2560
CPU type	8-bit AVR	16-bit ULP	8-bit AVR
Performance	20 MIPS at 20 MHz	16 MIPS at 16 MHz	16 MIPS at 20 MHz
Flash memory	32 KB	128 KB	256 KB
SRAM	2 KB	2 KB	8 KB
EEPROM	1 KB	0 KB	4 KB
Pin count	28	100	100
Maximum operating frequency	20 MHz	16 MHz	20 MHz
Maximum I/O pins	23	83	86
Cost	$2	$8	$12

MCU – SCHEMATIC

MCU – ATmega2560

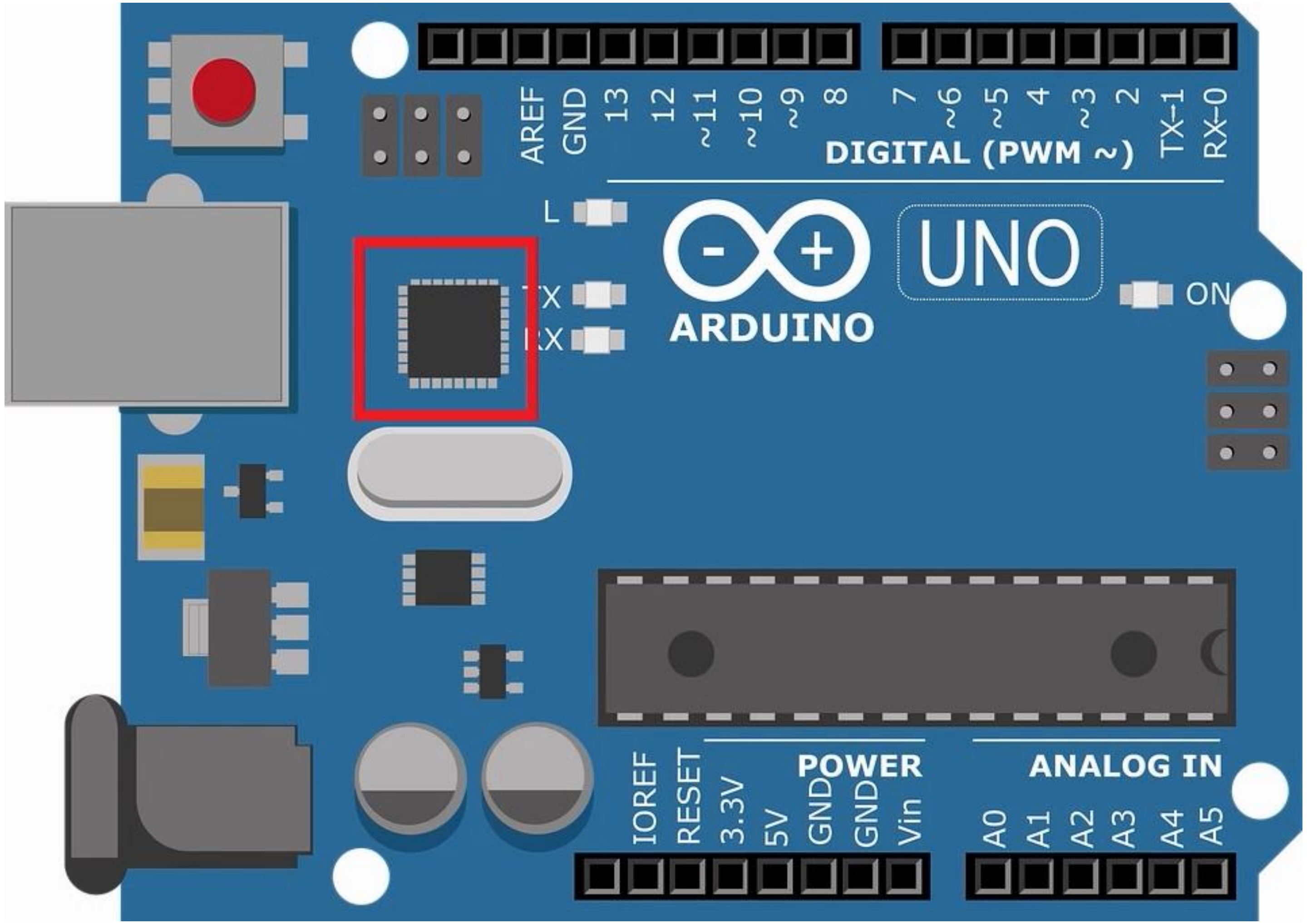
- Two communication headers, one with auto-reset.
- Header for burning bootloader and programming the MCU.
- 12 I/O lines for controlling a 7-segment display. 4 lines for button inputs.
- 6 lines for controlling the LED drivers.
- A reset button and power indicator LED.
- A MOSFET and voltage divider for controlling the cooling system.
- 16 MHz oscillator.



MCU – SERIAL COMMUNICATION

Communicating with the PC

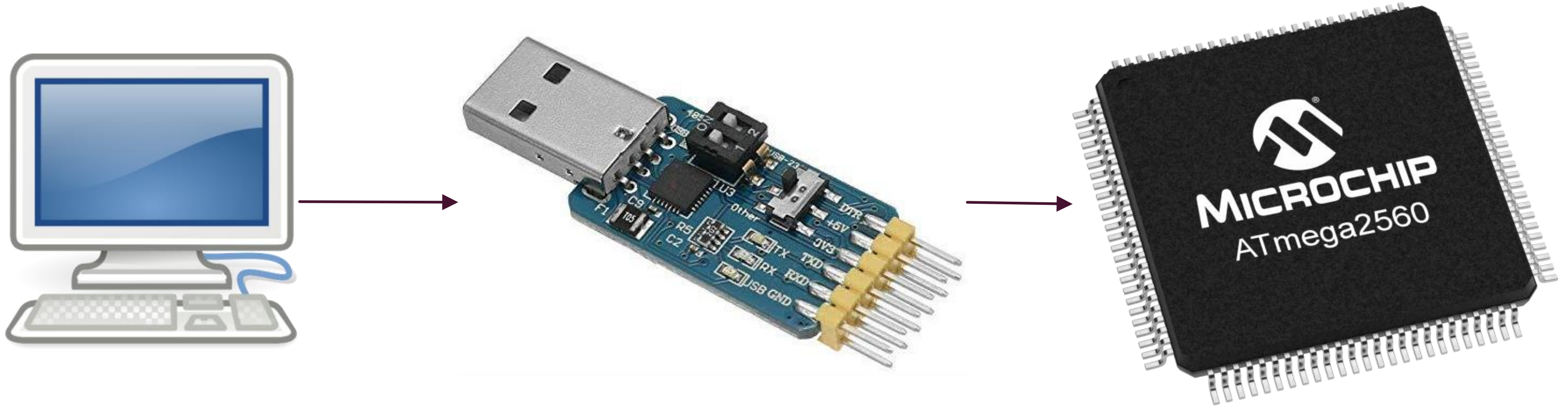
- Signals must be sent from the user's PC to the MCU for controlling LED effects.
- Arduino requires using a second MCU for USB to serial communication.
- This complicates PCB design and requires a second ICSP for programming.



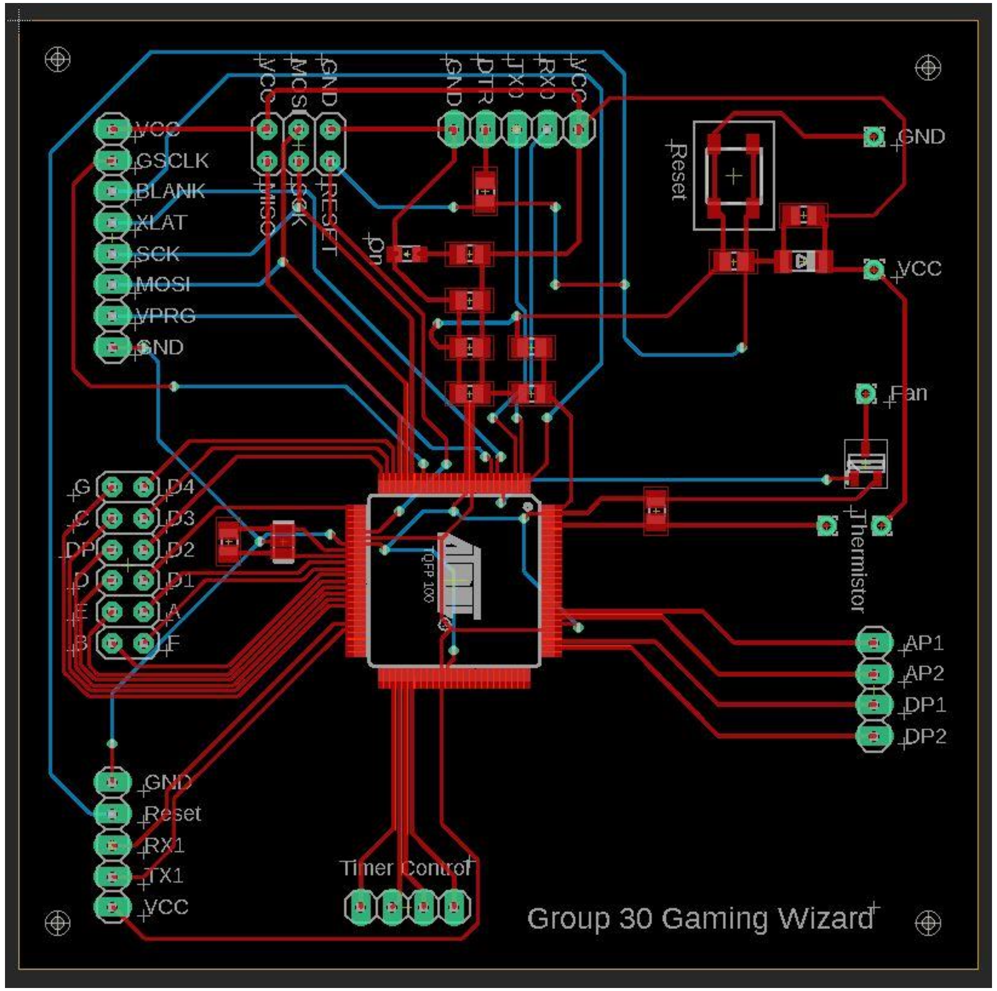
MCU – SERIAL COMMUNICATION

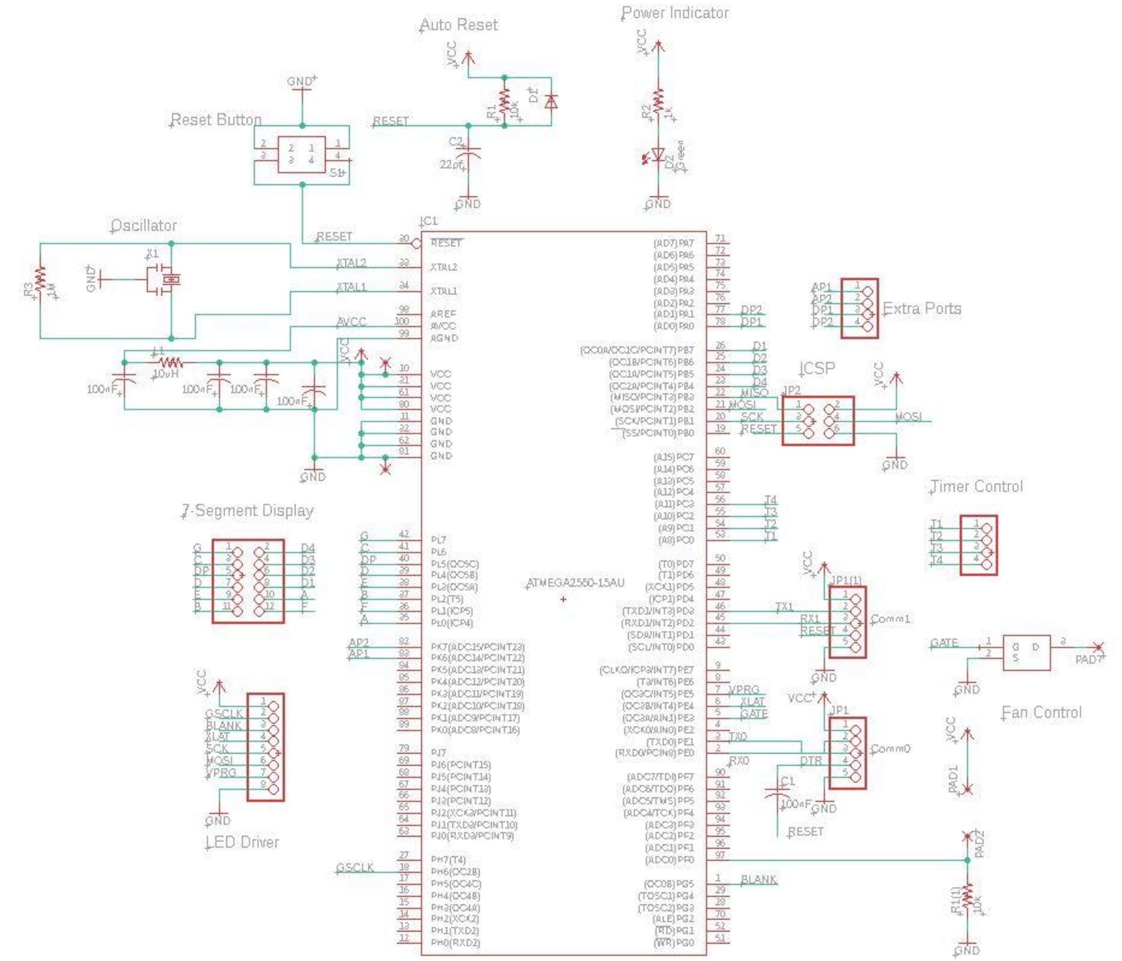
USB to Serial Adapter – FT232RL

- Allows communication with MCU from host PC.
- Allows programming of MCU via USB.
- Simplifies PCB design

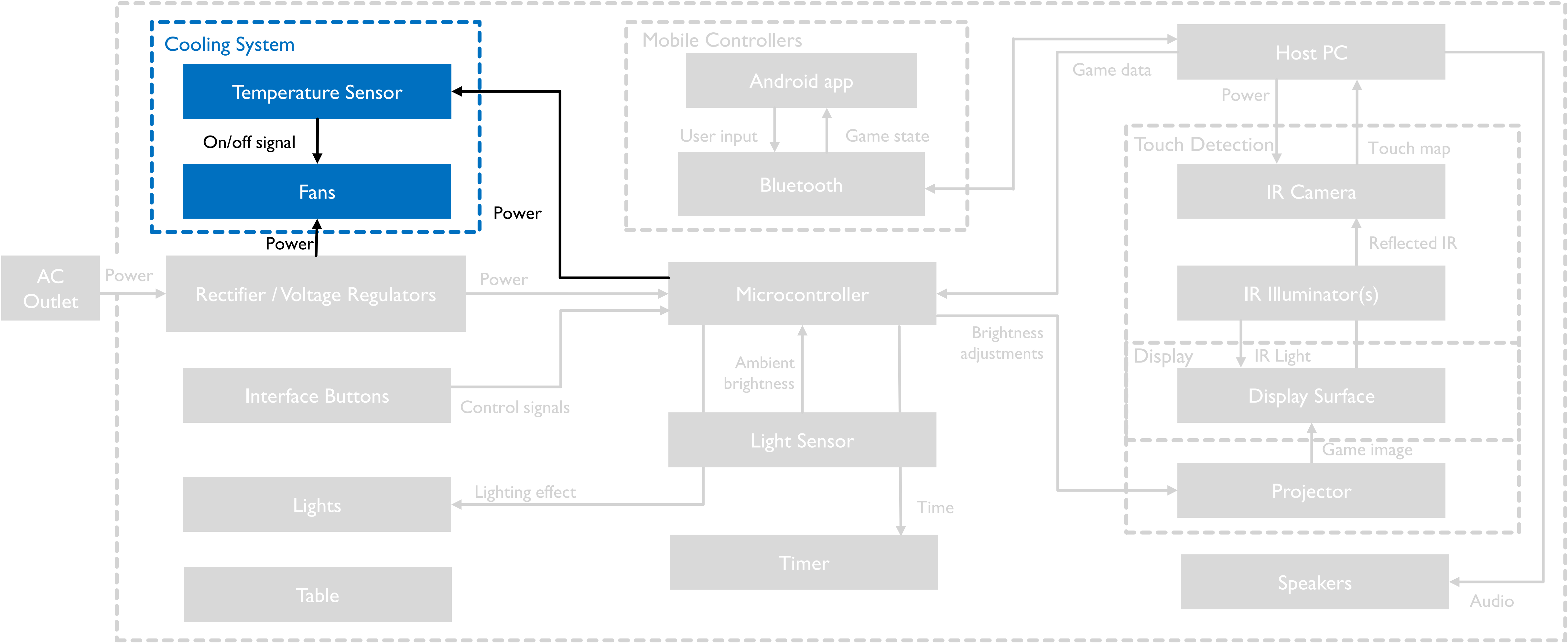


PCB DESIGN – MCU BOARD

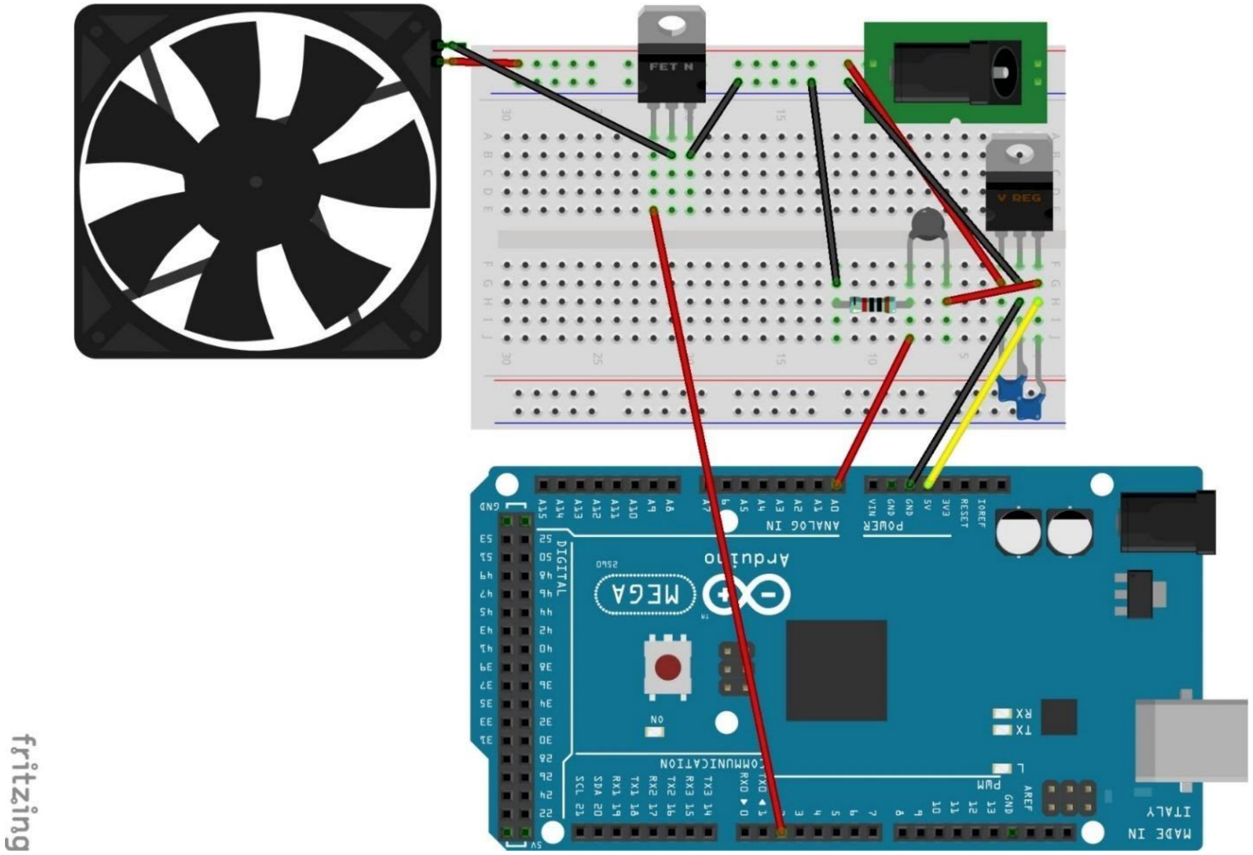




HARDWARE – COOLING SYSTEM

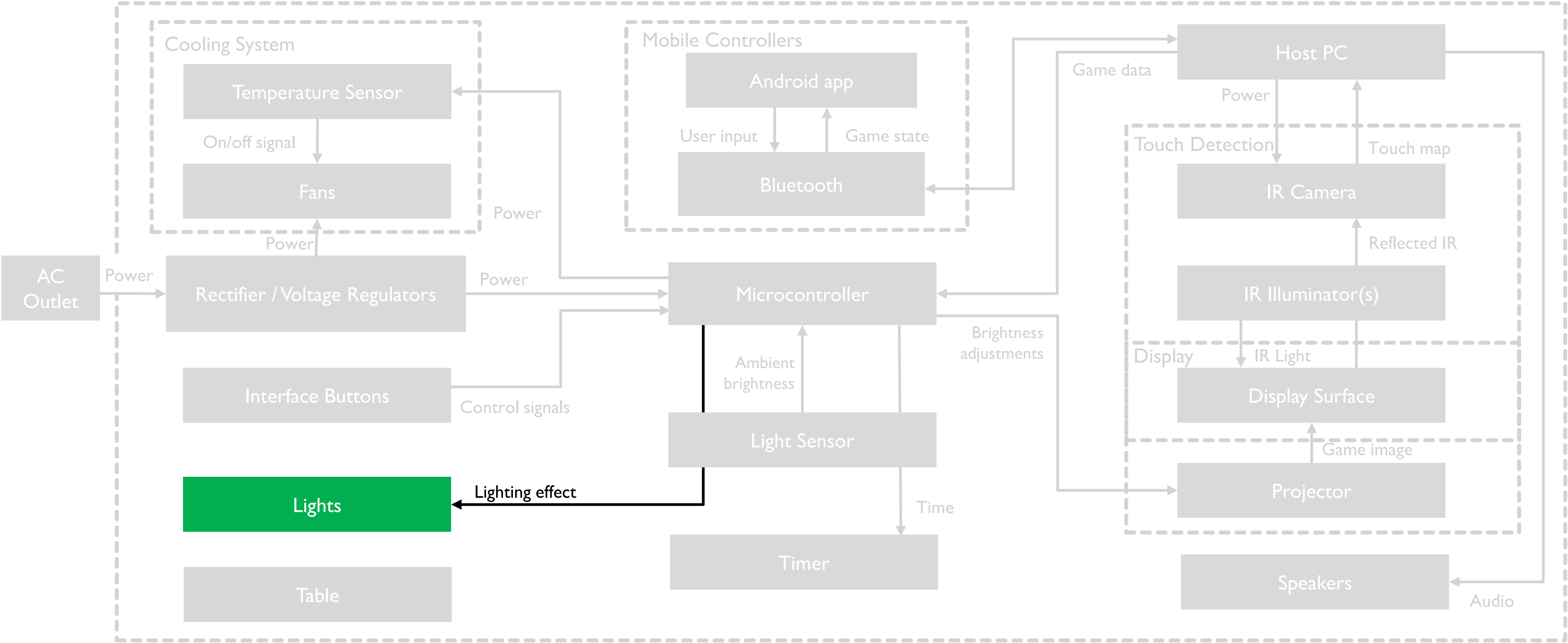


COOLING SYSTEM – FAN CONTROL



Prototype of fan control

HARDWARE – EFFECT LIGHTING



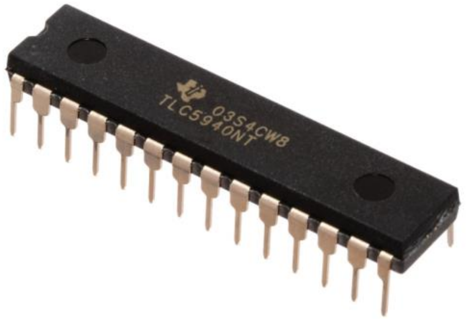
EFFECT LIGHTNING – LED DRIVER SELECTION

TLC5940 16-channel LED driver

- Advantages: Uses standard SPI interface (can be daisy chained), Larger PWM depth (12 bits vs 8 bits).
- Disadvantages: More complicated to implement. Requires six output lines from MCU.

WS2812b "Neopixel"

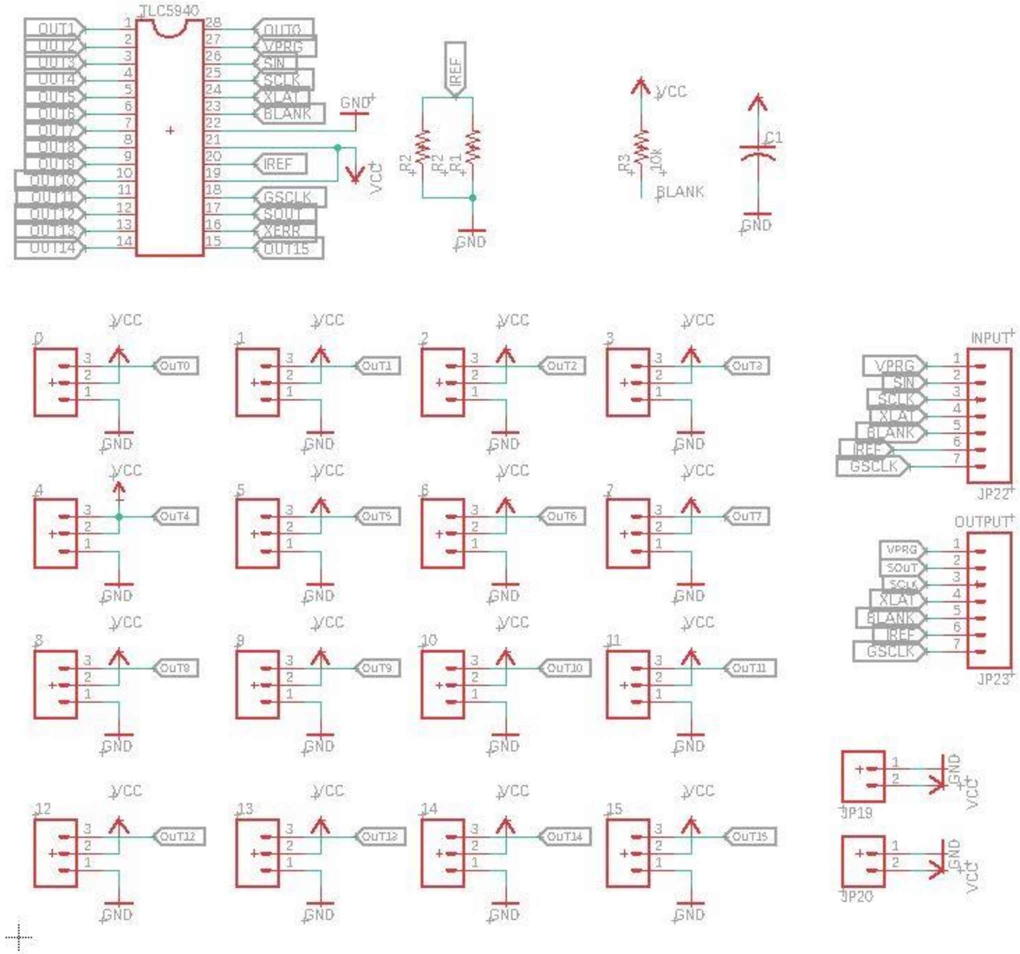
- Advantages: Only requires a single data line, Easy to control large amount.
- Disadvantages: Communication protocol not standard and is handled via software "bit banging".

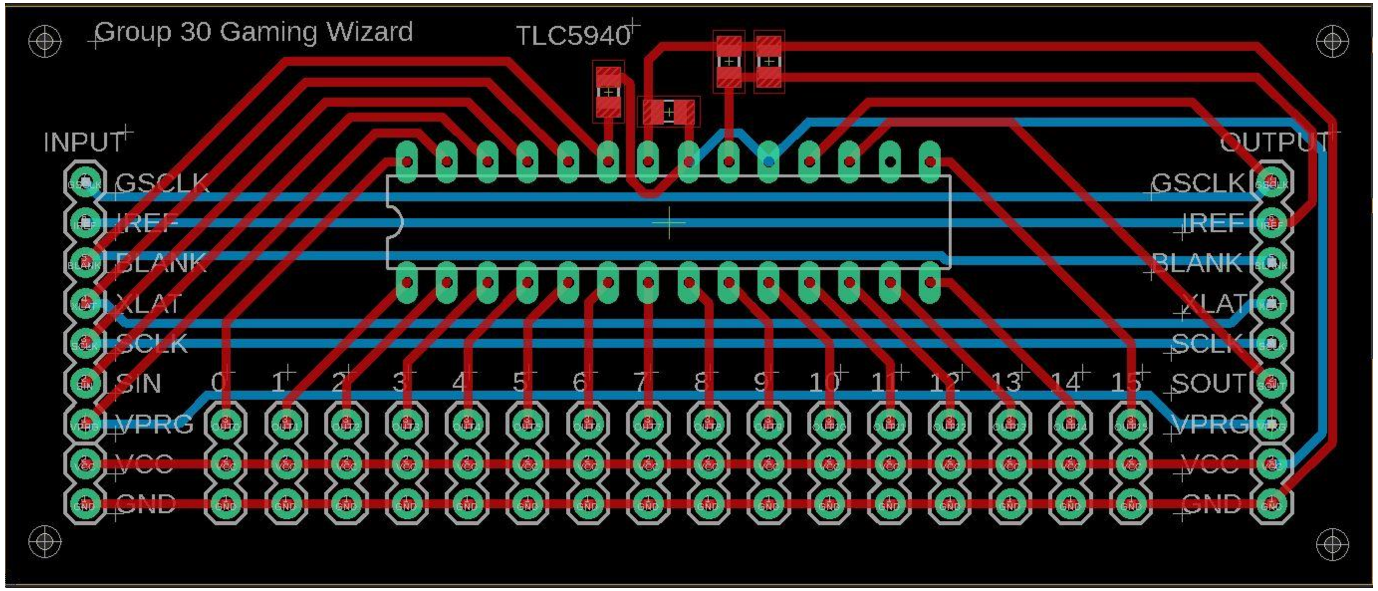


TLC5940

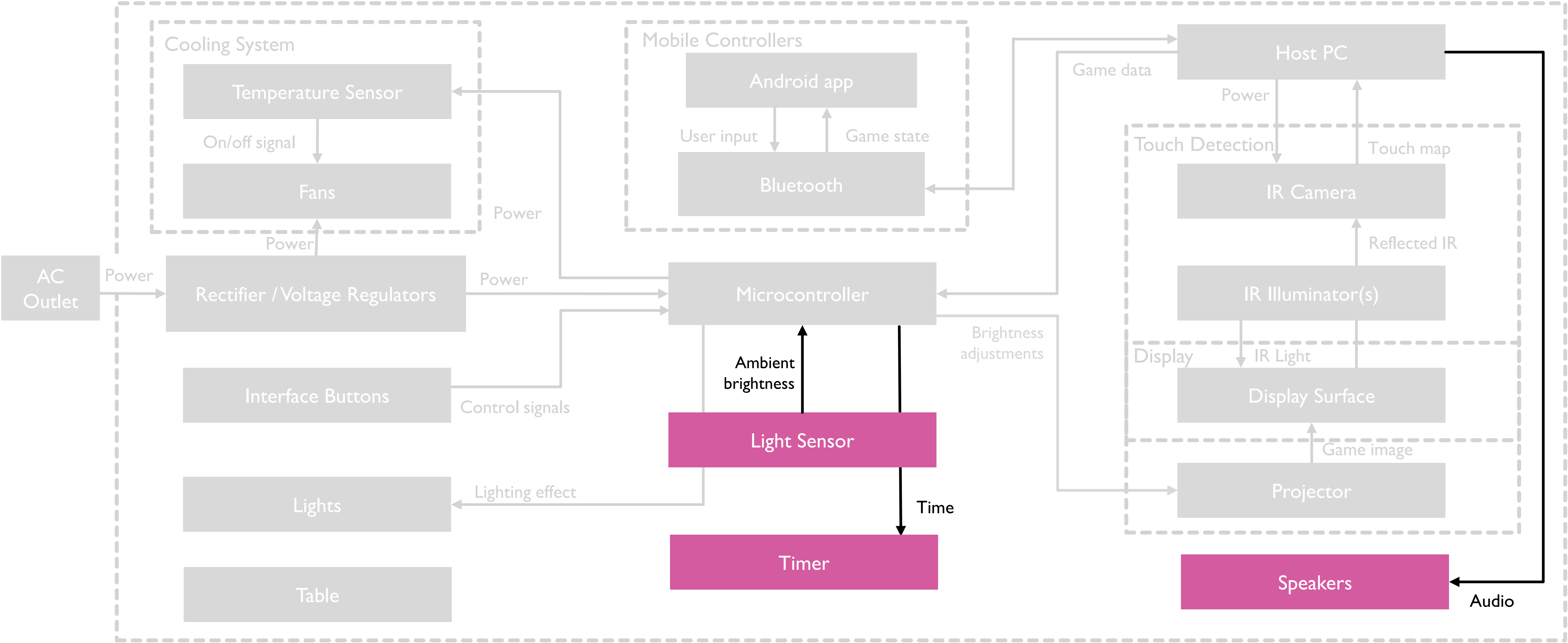
Ws2812b neopixel

PCB DESIGN – LED LIGHTING BOARD





HARDWARE – ADDITIONAL FEATURES



ADDITIONAL FEATURES

Timer

- Limit time per turn
- Set time-sensitive challenges for players
- Quad 7-Segment LED Display
  - Multiplexing for each digit
  - 16 pins
  - Low power usage (40 mW/segment)
- Start/Stop, +Minute, and +Second Buttons
- 4 States of Operation: Off, On, Running, Complete

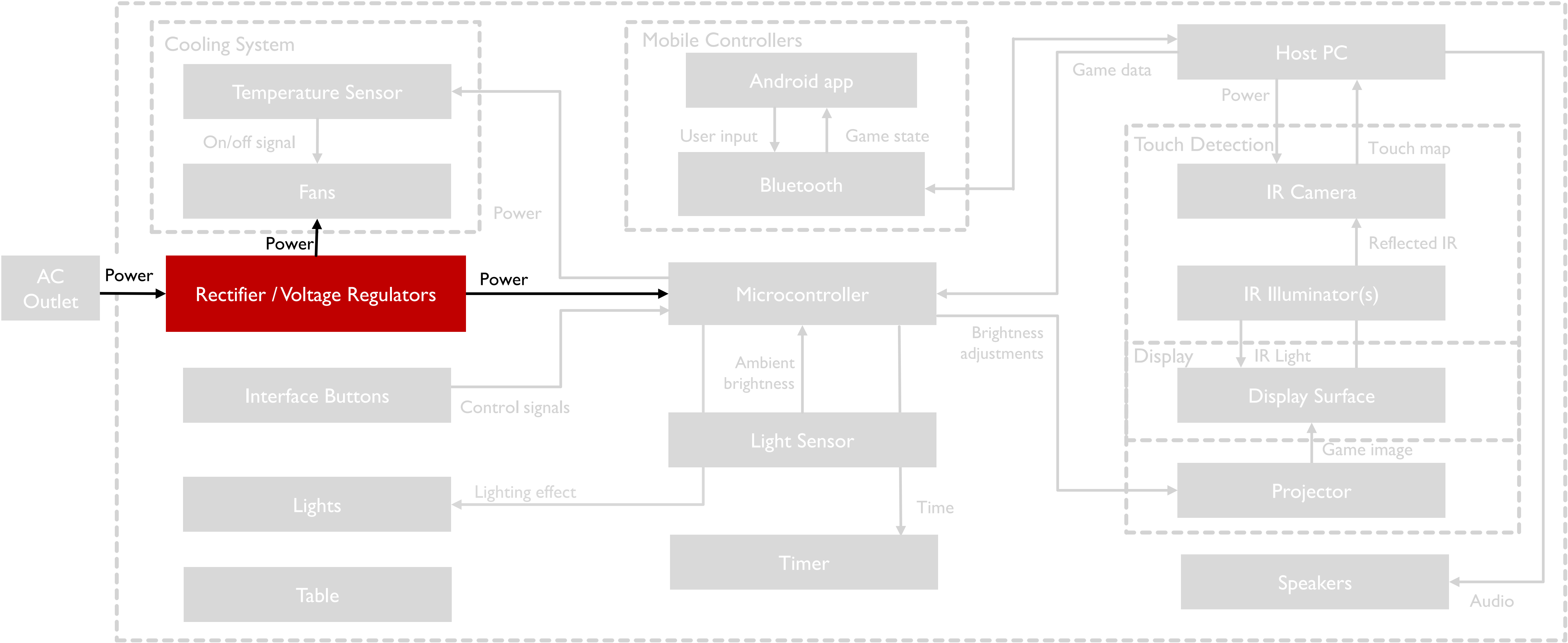
Automatic Brightness Adjustment

- Improve visibility in bright conditions and protect eyes in dim conditions by varying the display brightness automatically
- Estimate brightness using photoresistor
- Compare to current display brightness
- Adjust display brightness based on difference

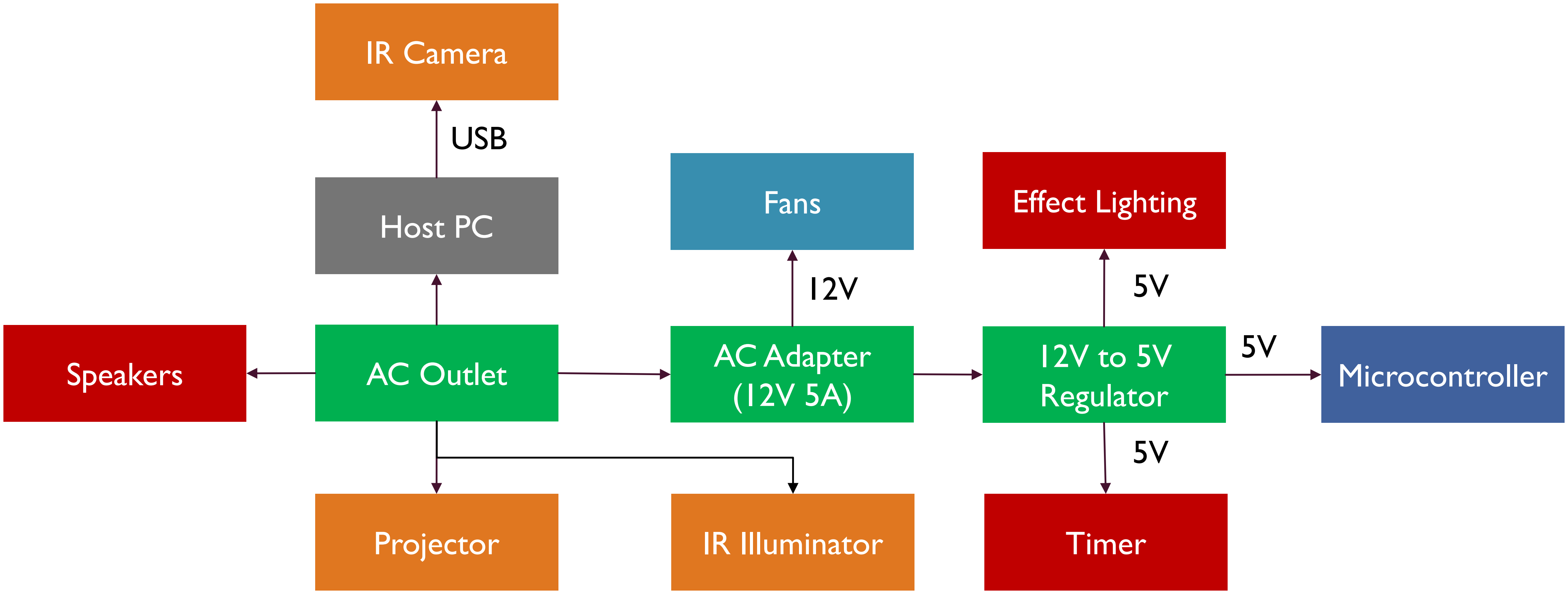
Sound Effects

- Donated PC speakers placed inside table

HARDWARE – POWER SYSTEM



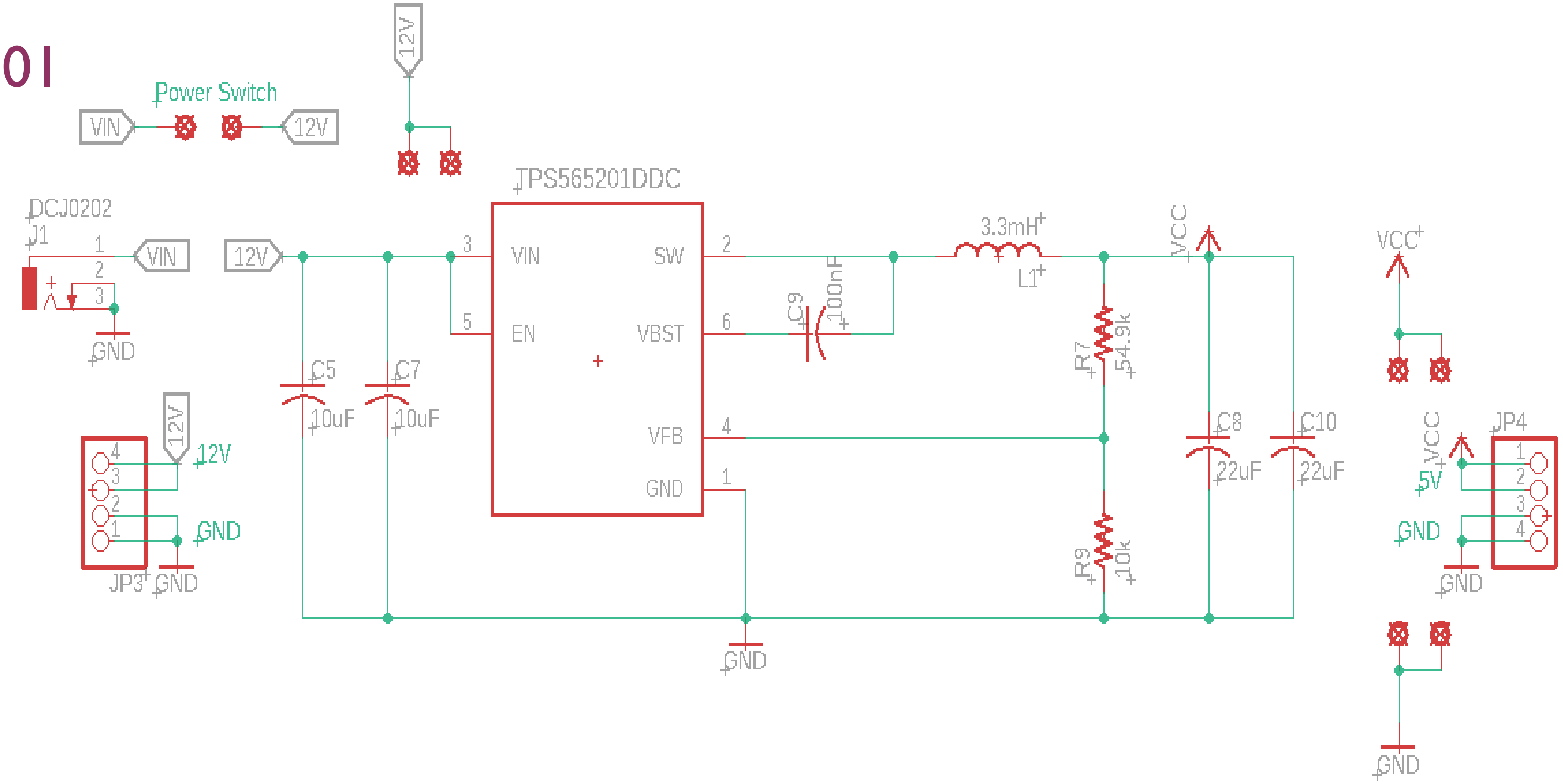
POWER SYSTEM - DIAGRAM



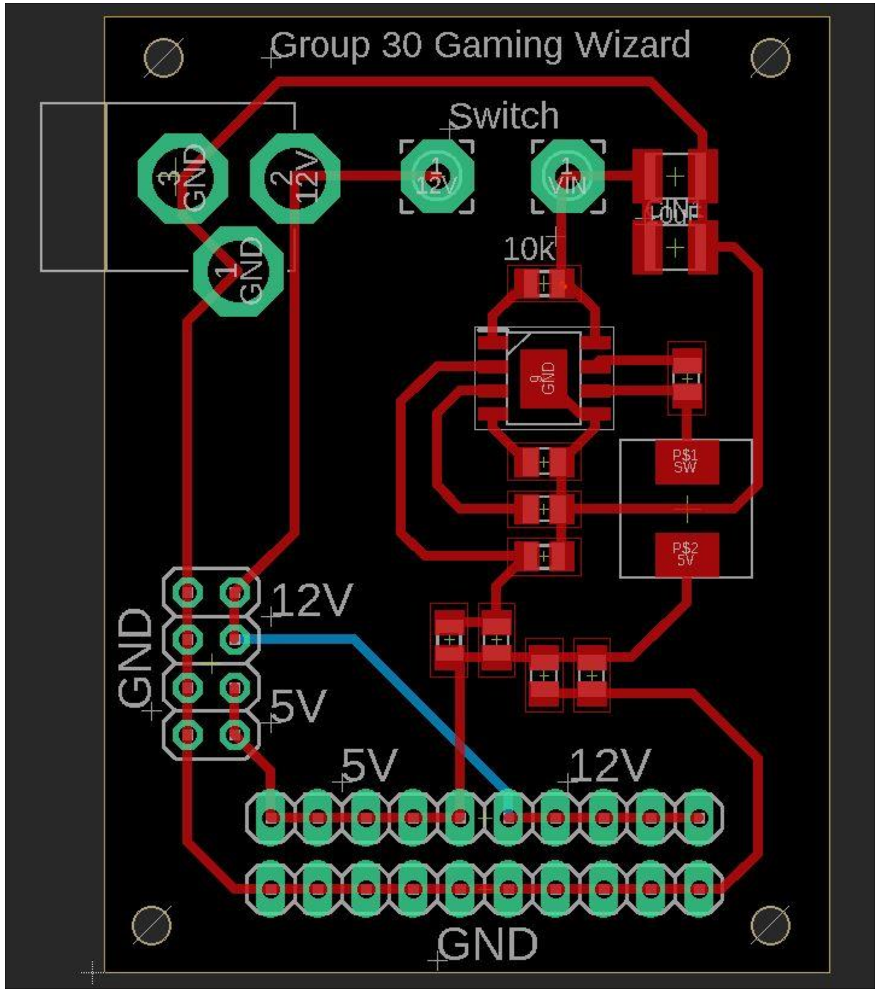
POWER SYSTEM – REGULATOR SCHEMATIC

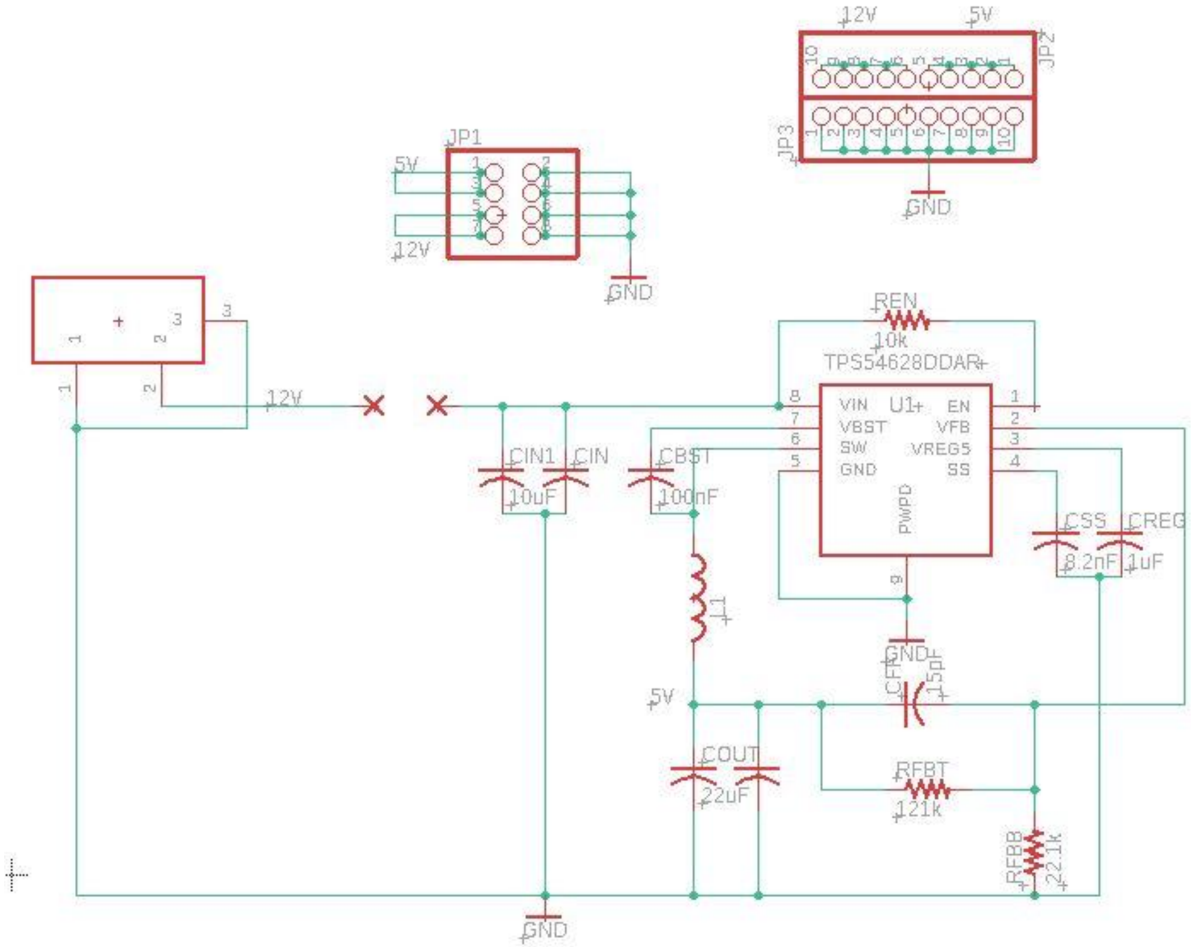
Voltage Regulator – TPS565201

- 12V input and 5V output.
- 5A maximum output current.
- DC barrel jack for 12V input.
- Solder points for power switch.
- Pin headers for easy access to 12V and 5V

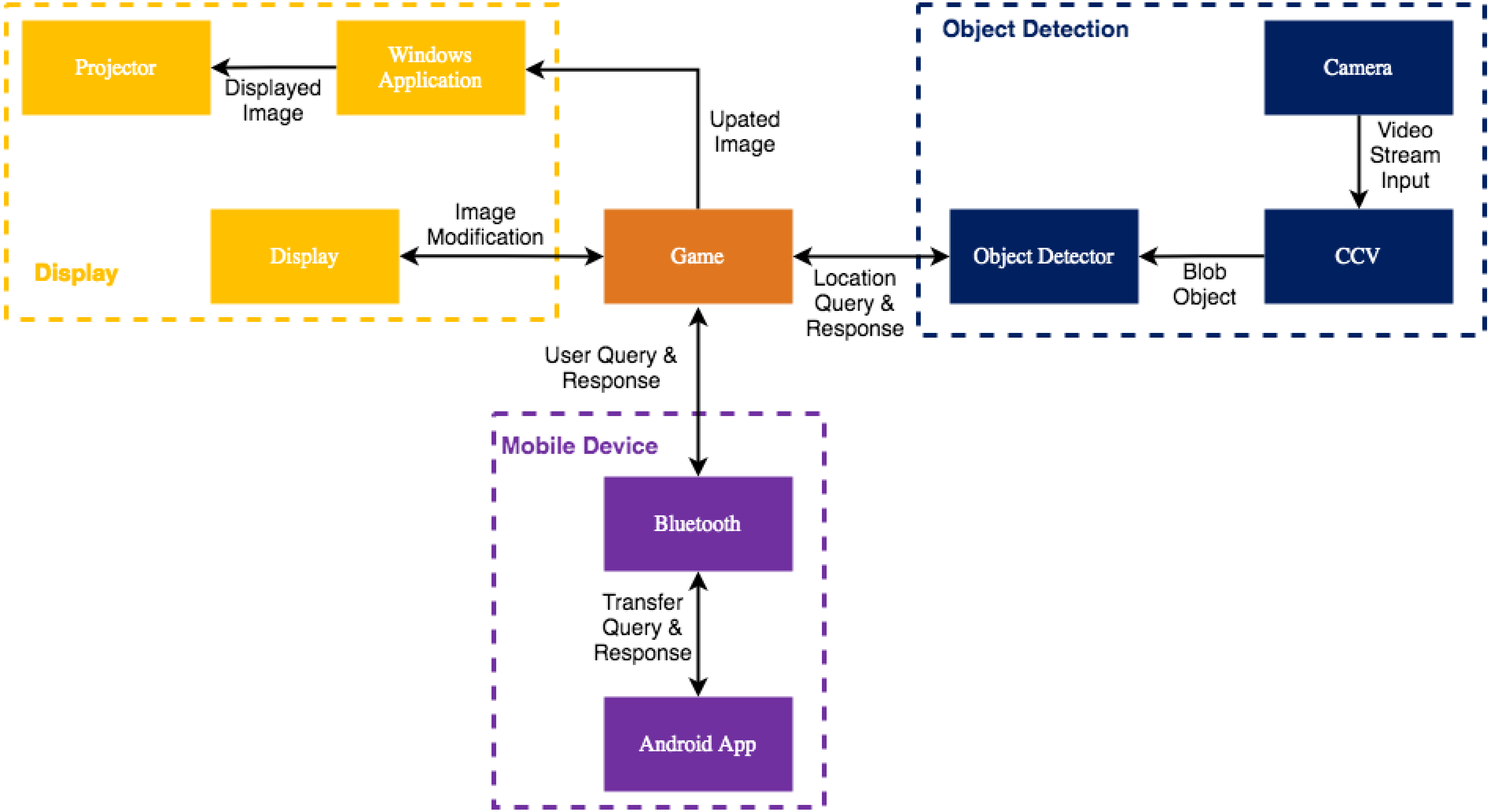


PCB DESIGN – POWER SYSTEM BOARD





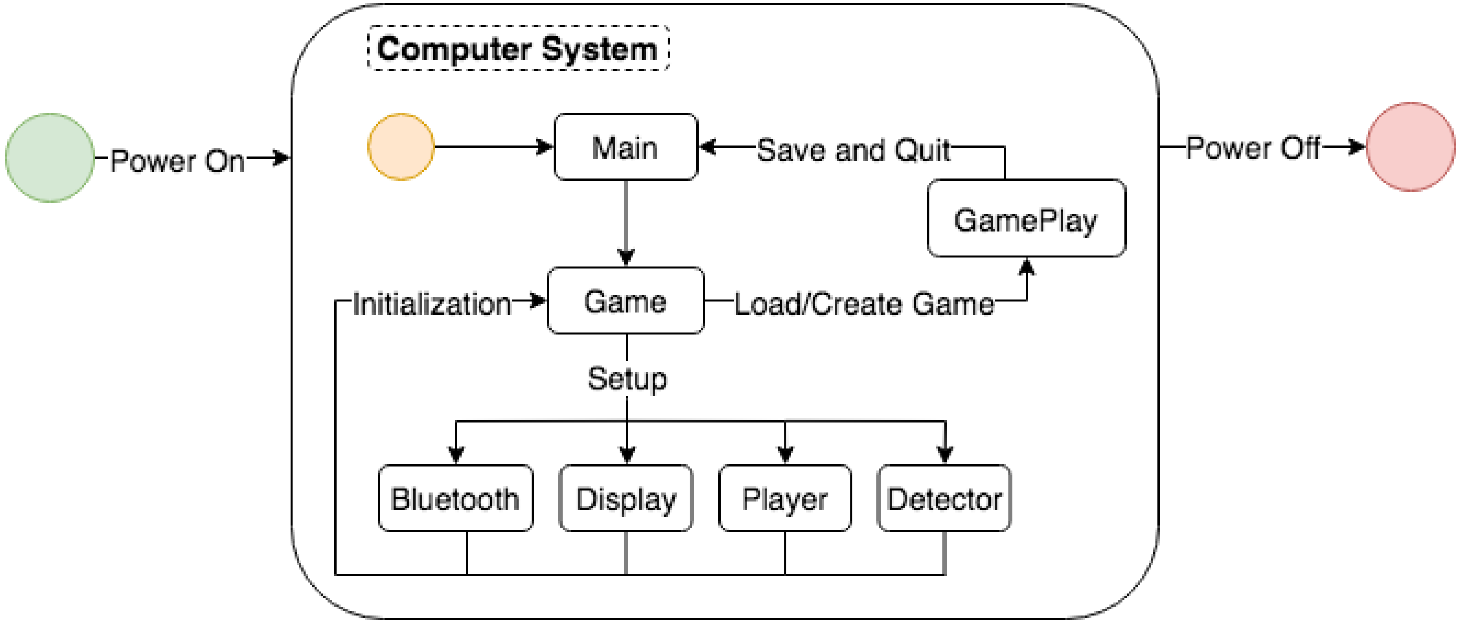
PROPOSED SOLUTION – SOFTWARE



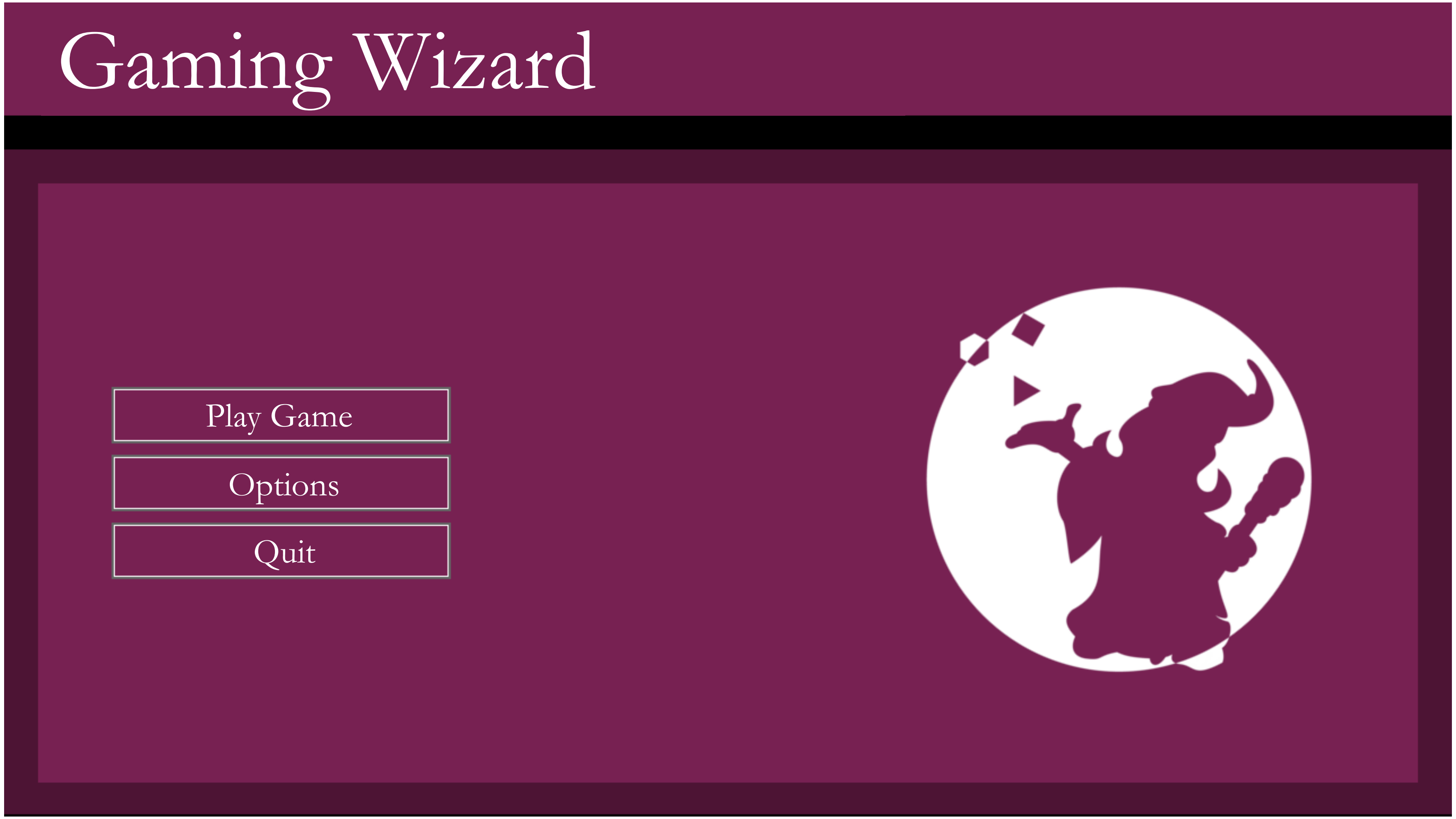
GENERAL SOFTWARE FUNCTIONALITY

General Flow of the Program

- Windows Application is the starting point of the software
- Spawns a thread to begin Game initiation
- Game instantiates Bluetooth, Display, Player(s), and Detector objects
- Game maintains game flow until power off event



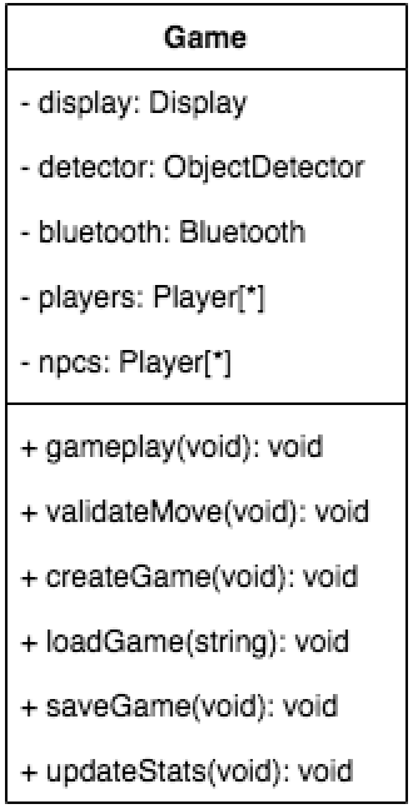
MAIN MENU



GAME

Game Class Purpose:

- Maintain logical control of the game
  - Keep track of turns
  - Maintain mapping of player to mobile device and player piece
  - Update display
- Handle inputs from all software subsystems



Game

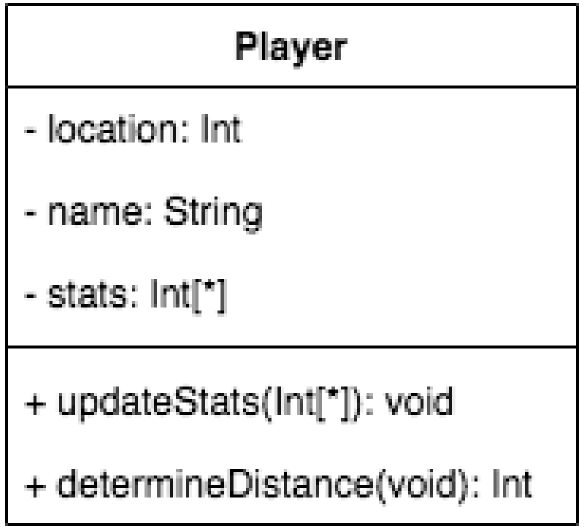
- display: Display
- detector: ObjectDetector
- bluetooth: Bluetooth
- players: Player[*]
- npcs: Player[*]

+ gameplay(void): void
+ validateMove(void): void
+ createGame(void): void
+ loadGame(string): void
+ saveGame(void): void
+ updateStats(void): void

PLAYER

Player Class Purpose:

- Object to create distinct players for the game
- Maintains player stats and location information
- Parent class to NPC’s and human players
  - NPCs: Image path
  - Human Players: Device ID



Player

- location: Int

- name: String

- stats: Int[*]

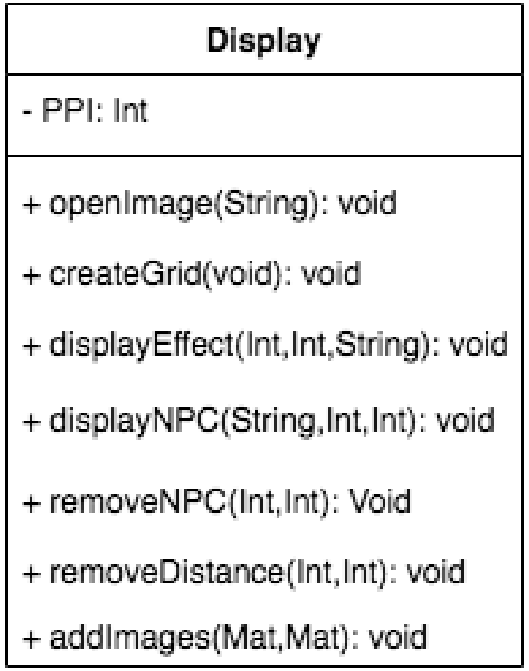
+ updateStats(Int[*]): void

+ determineDistance(void): Int

DISPLAY

Display Class Purpose:

- Image processing in the background
  - Take user provided image, add gridlines, and create map to be displayed
  - Spawn NPC images in desired locations
  - Create special effects for movement and attack phases
- OpenCV framework
- Displaying of the image handled by the windows application methods

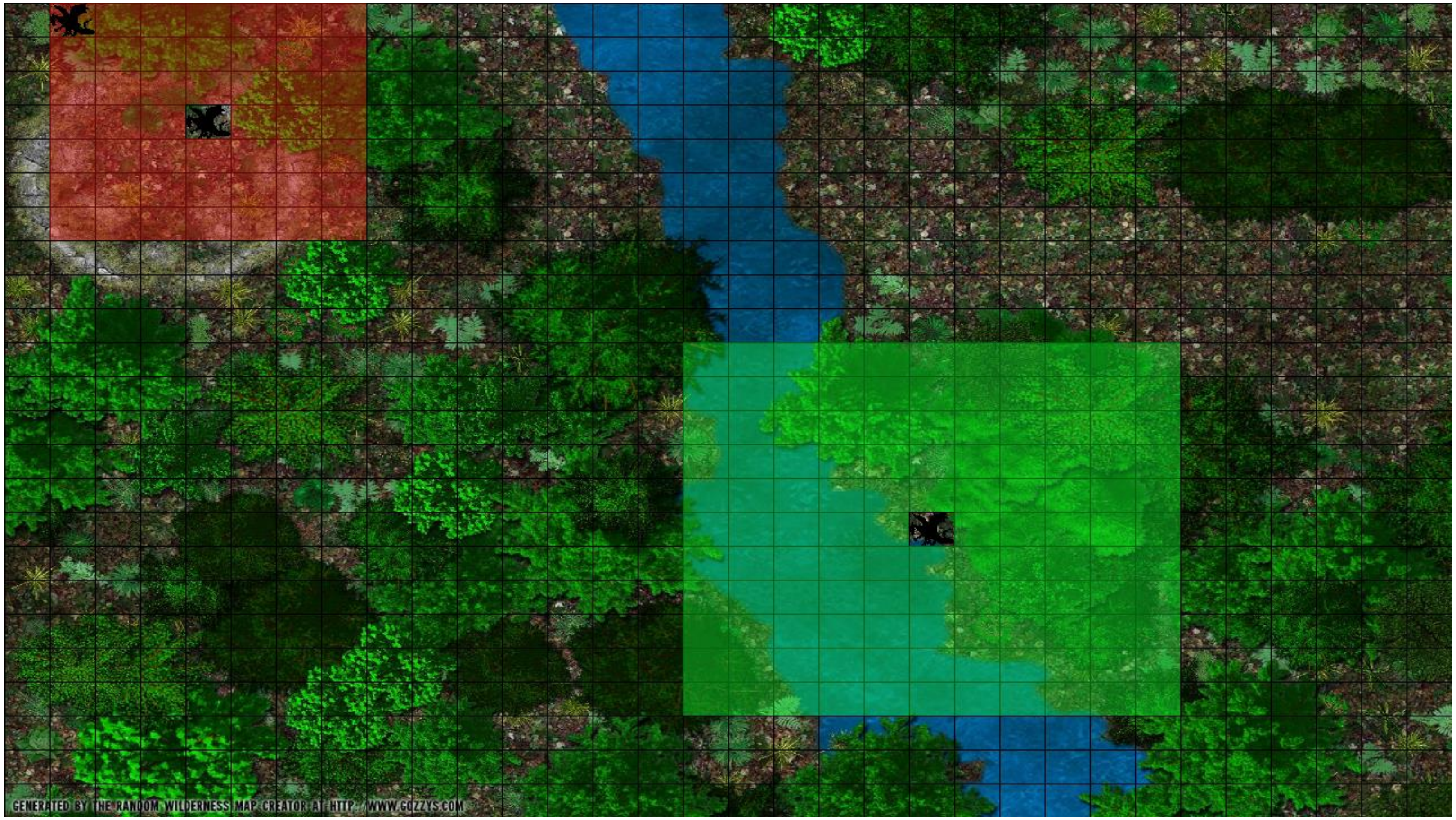


Display

- PPI: Int

+ openImage(String): void
+ createGrid(void): void
+ displayEffect(Int,Int,String): void
+ displayNPC(String,Int,Int): void
+ removeNPC(Int,Int): Void
+ removeDistance(Int,Int): void
+ addImages(Mat,Mat): void

DISPLAY


GENERATED BY THE RANDOM WILDERNESS MAP CREATOR AT HTTP://WWW.GOZZYS.COM

OBJECT DETECTION

Community Core Vision (CCV)

- Blob tracking software with computer vision
  - Used to track player pieces and finger touches
- Supports FTIR, DI, DSI, and LLP
- Open Source
- Uses TUIO API to store and transmit information

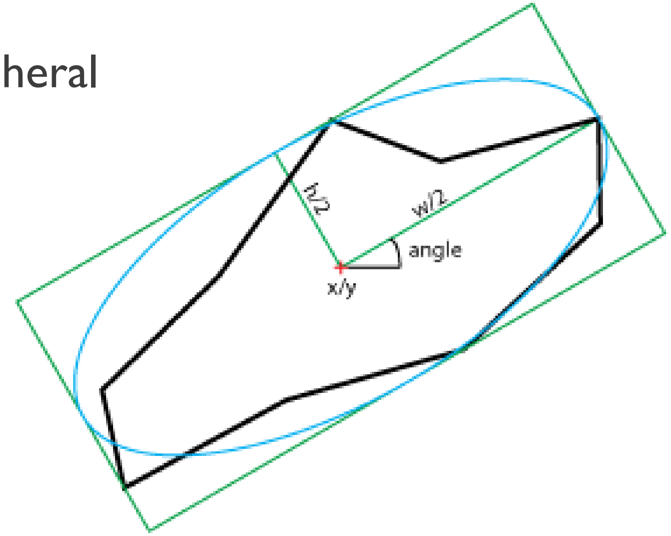
Minimum System Requirements	
CPU	Pentium 4
Ram	512 MB
GPU (Optional)	Modern GPU
Operating System	Windows, Mac, Linux
Peripherals	Camera

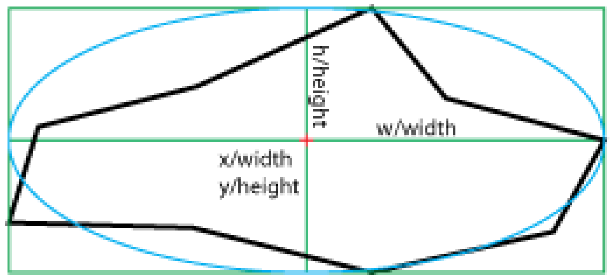
OBJECT DETECTION

Tuio objects

- API provides data structures for Blobs and Cursors
- API provides protocols to transmit information between programs
  - TCP/UDP sockets
- Normalized values based on resolution of peripheral camera

Parameter Name	Parameter Meaning
s	Session ID (temporary object ID)
x, y, z	Position
a, b, c	Angle
w, h, d	Dimension
f, v	Area, Volume

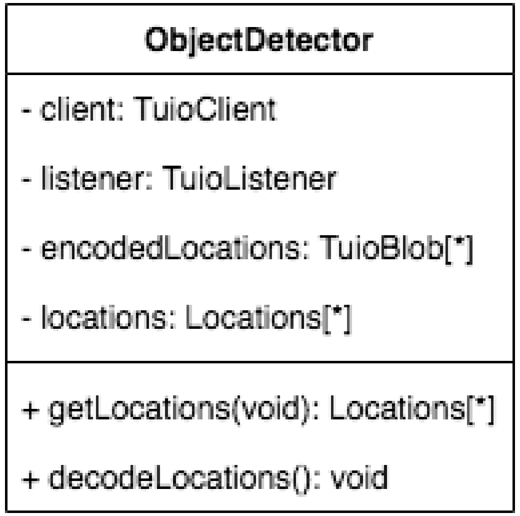




OBJECT DETECTION

ObjectDetector Class Purpose:

- Open, maintain, and handle TUIO connection with CCV
- Receive and decode blob information
  - Translate location to a grid location
- Package and store in shared memory
- Requires its own thread
  - Needs to poll CCV to not miss an event



ObjectDetector

- client: TuioClient

- listener: TuioListener

- encodedLocations: TuioBlob[*]

- locations: Locations[*]

+ getLocations(void): Locations[*]

+ decodeLocations(): void

SYSTEM REQUIREMENTS

Recommended Specifications:

- CPU: Pentium 4 or better
- RAM: 512 MB
- GPU: Any Modern Card (Optional)
- Disk Space: TBD
- OS: Windows 10
- Bluetooth capability

Testing System Specifications:

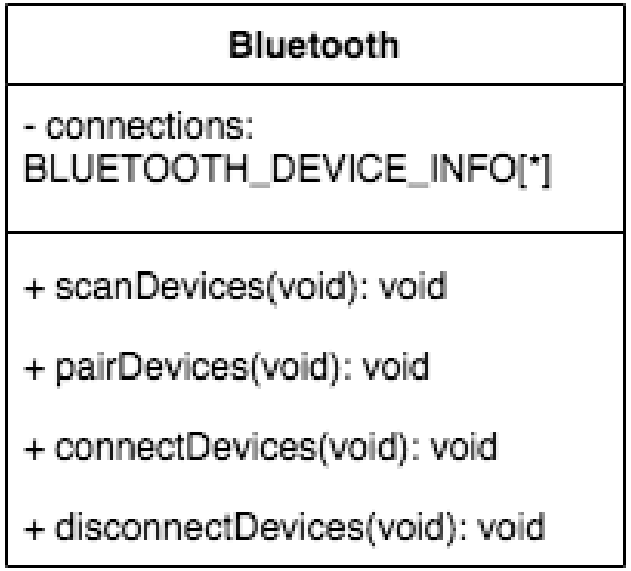
- CPU: Intel Core i5-4210H
- RAM: 8 GB
- GPU: NVIDIA GeForce GTX 965M
- Disk Space: 500 GB Free
- OS: Windows 10 Pro v1909, 64-Bit
- Bluetooth capability

MOBILE APP CONNECTION

- Connection made with a Bluetooth connection
- PC and mobile device must have Bluetooth enabled
- Windows API required to create connection with other devices

Bluetooth Class Purpose:

- Setup and maintain connection to each mobile device
- Handle data transfer and storage



Bluetooth

- connections: BLUETOOTH_DEVICE_INFO[*]

+ scanDevices(void): void

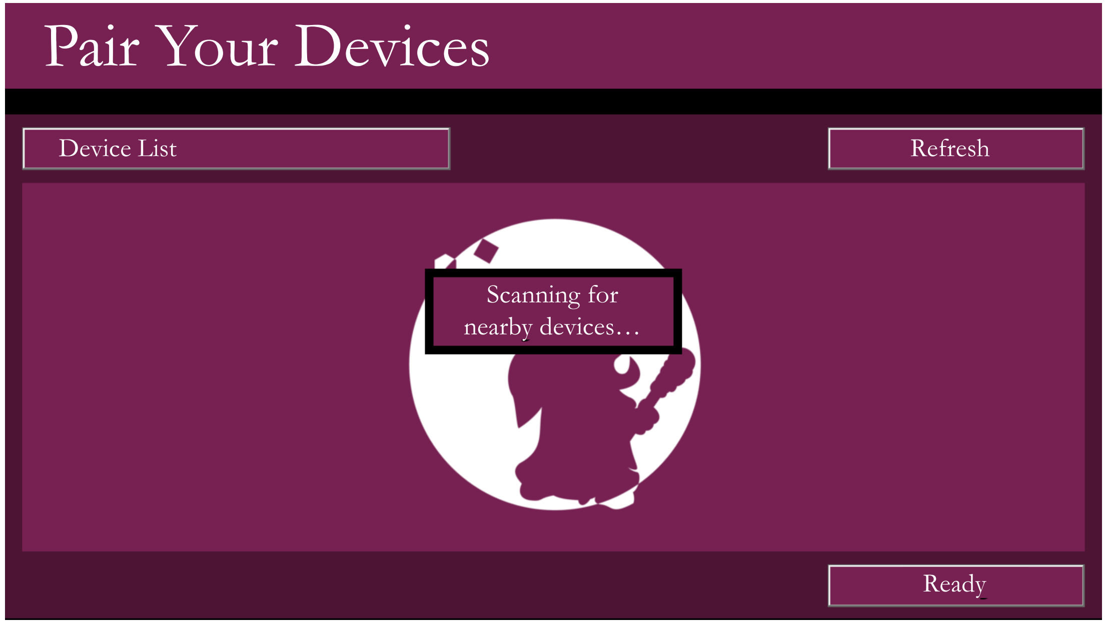
+ pairDevices(void): void

+ connectDevices(void): void

+ disconnectDevices(void): void

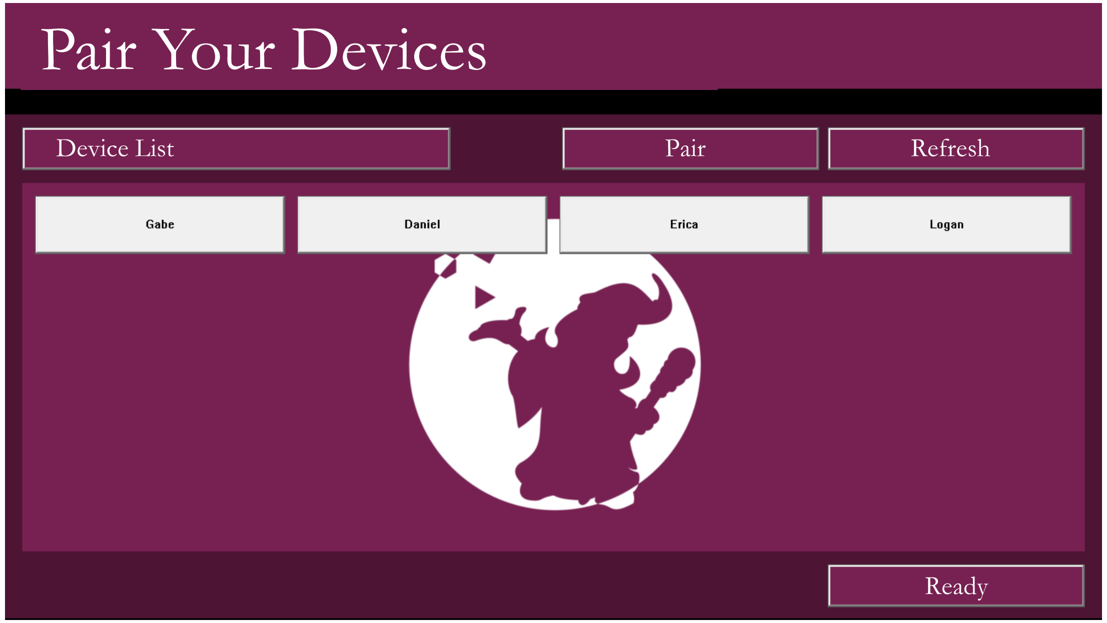
BLUETOOTH

Scanning



Scanning for nearby devices…

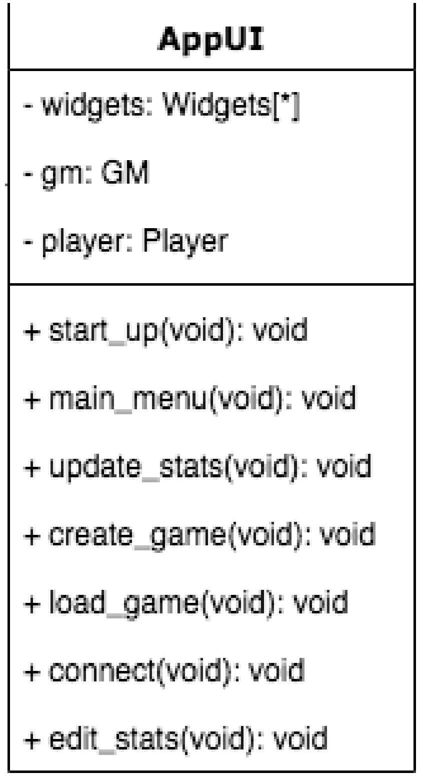
Devices Found



APP SETUP

Initial Structure

- First Person to connect will be maintained as the Game Master (GM)
    - Create game will allow GM to set number of players and name the campaign
    - Load game will grab information from local PC
- Players will then join game and create new characters or select their characters from the saved game
    - When starting an encounter, player will place their pieces on the board for object detection to recognize their characters



AppUI

- widgets: Widgets[*]

- gm: GM

- player: Player

+ start_up(void): void

+ main_menu(void): void

+ update_stats(void): void

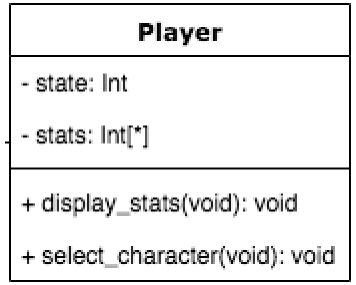
+ create_game(void): void

+ load_game(void): void

+ connect(void): void

+ edit_stats(void): void

PLAYER OPTIONS



Player

- state: Int

- stats: Int[*]

+ display_stats(void): void

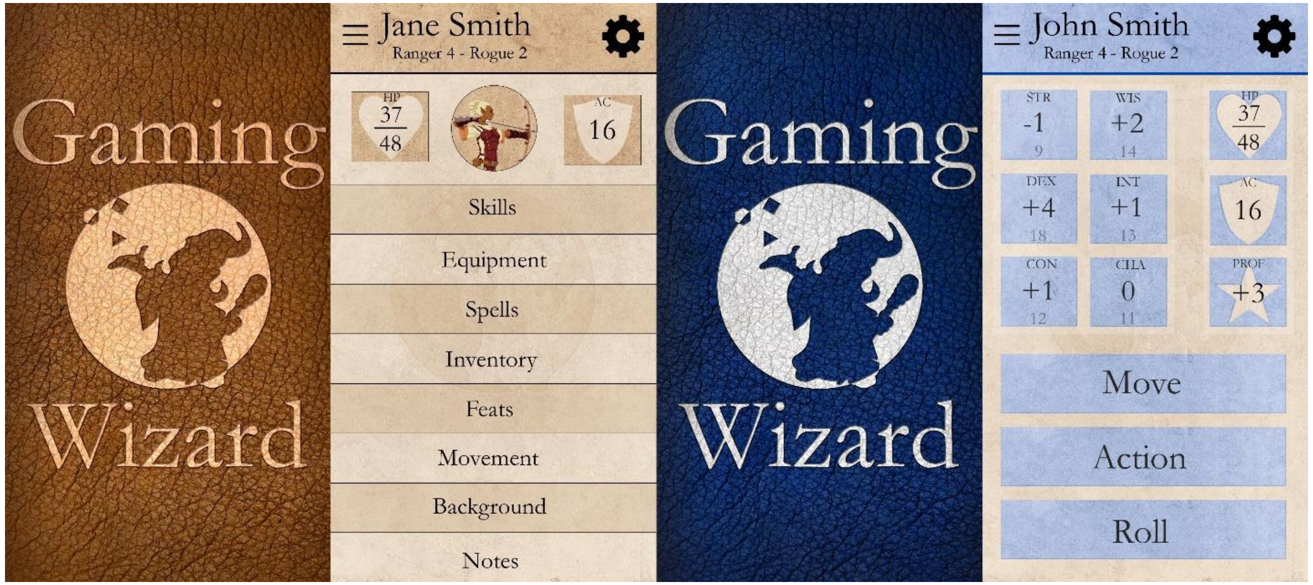
+ select_character(void): void

Not on turn

- Manage actions such as attacks or spells
- Change attribute points for character sheet, character name, and level
- Roll a dice from a list, which will generate a random number appropriate for that dice

On turn

- Select move, attack, or any management option from before



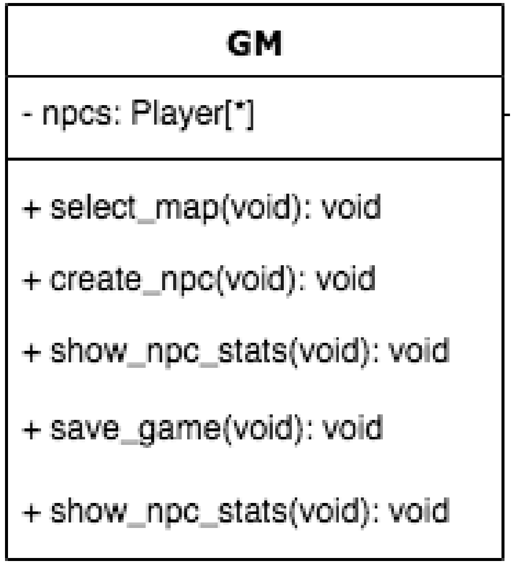
GAME MASTER OPTIONS

GM Controlled Characters

- Same logic as players
- GM has a list of monsters and characters that they control during an encounter.

GM Only Options

- Select map from list of created maps from local PC
- Set turn order
- Add NPCs to board



GM

- npcs: Player[*]

+ select_map(void): void

+ create_npc(void): void

+ show_npc_stats(void): void

+ save_game(void): void

+ show_npc_stats(void): void

DIVISION OF LABOR

Hardware Team

Category	Logan	Erica
Cooling System	**Primary**	Secondary
Display	Secondary	**Primary**
Extra Features	Secondary	**Primary**
PCB Design	**Primary**	Secondary
Microcontroller	**Primary**	Secondary
Power System	**Primary**	Secondary
Effect Lighting	**Primary**	Secondary
Table	Secondary	**Primary**
Touch Detection	Secondary	**Primary**

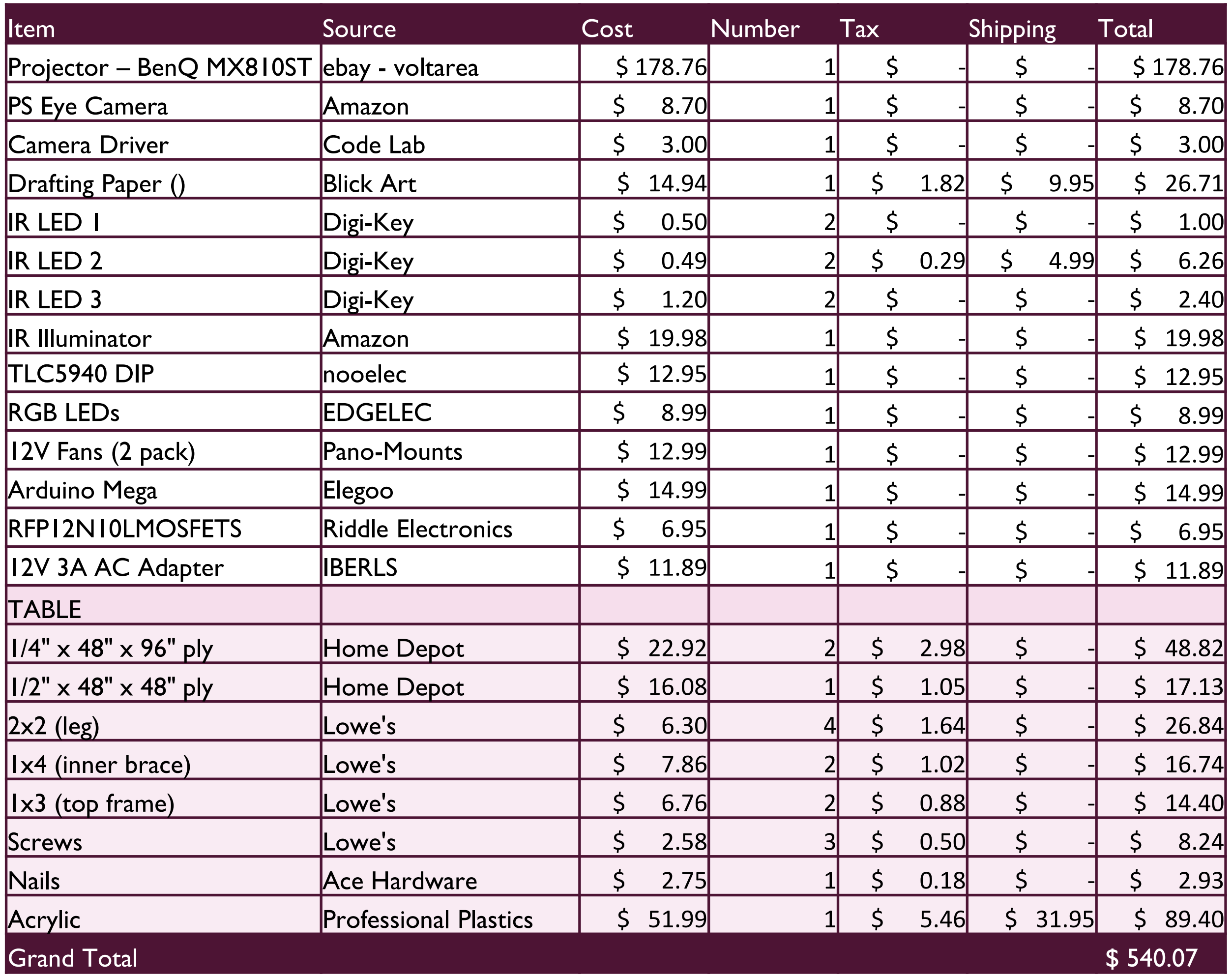
Software Team

Category	Gabe	Daniel
Game	Secondary	**Primary**
Touch and Object Detection	Secondary	**Primary**
Bluetooth	Secondary	**Primary**
App Development	**Primary**	Secondary
Special Effects	**Primary**	Secondary

BUDGET

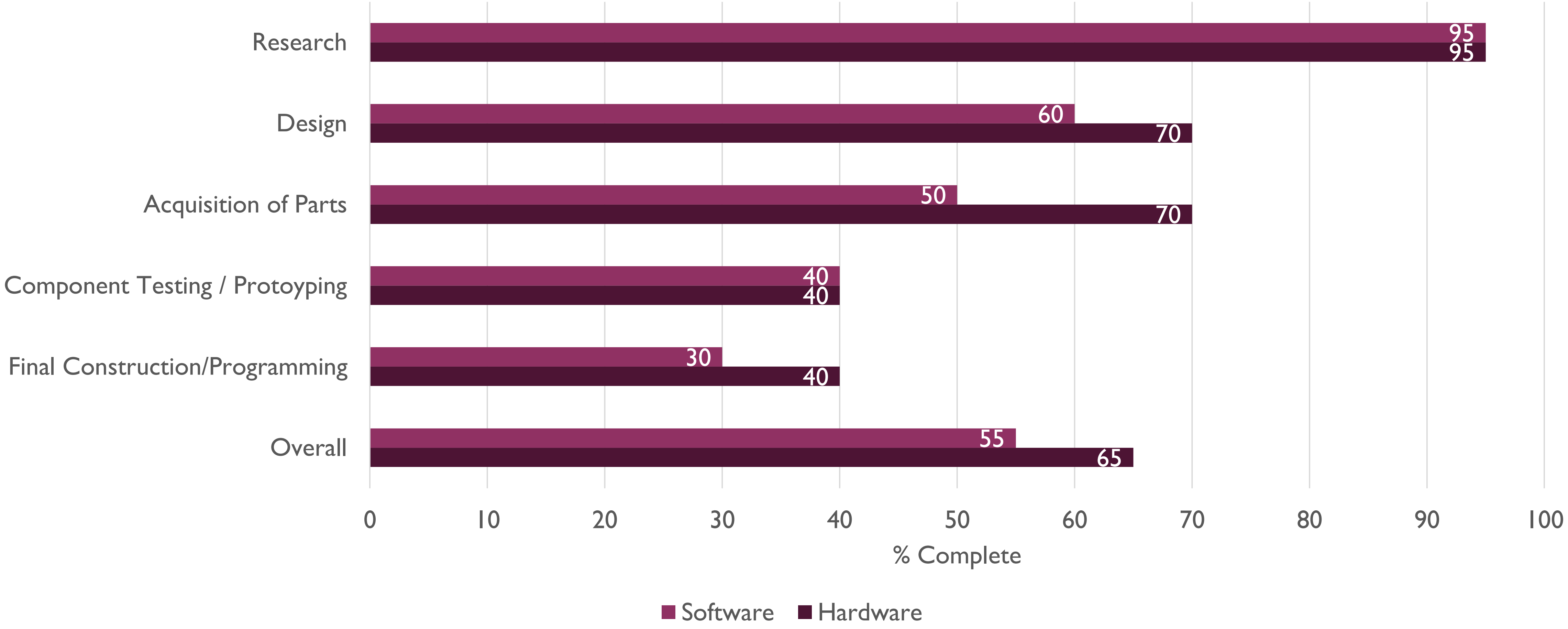
Factors

- No financial sponsor for our project, so all financial burden is placed on the group
- Set budget to be $700, and current cost of materials and equipment purchased is around $540, with only PCB and a few minor electrical components remaining
- Tried to minimize cost of items through deals and use of already owned materials
  - Laptops, breadboards, speakers, etc.



Item	Source	Cost	Number	Tax	Shipping	Total
Projector – BenQ MX810ST	ebay - voltarea	$ 178.76	1	$ -	$ -	$ 178.76
PS Eye Camera	Amazon	$ 8.70	1	$ -	$ -	$ 8.70
Camera Driver	Code Lab	$ 3.00	1	$ -	$ -	$ 3.00
Drafting Paper ()	Blick Art	$ 14.94	1	$ 1.82	$ 9.95	$ 26.71
IR LED 1	Digi-Key	$ 0.50	2	$ -	$ -	$ 1.00
IR LED 2	Digi-Key	$ 0.49	2	$ 0.29	$ 4.99	$ 6.26
IR LED 3	Digi-Key	$ 1.20	2	$ -	$ -	$ 2.40
IR Illuminator	Amazon	$ 19.98	1	$ -	$ -	$ 19.98
TLC5940 DIP	nooelec	$ 12.95	1	$ -	$ -	$ 12.95
RGB LEDs	EDGELEC	$ 8.99	1	$ -	$ -	$ 8.99
12V Fans (2 pack)	Pano-Mounts	$ 12.99	1	$ -	$ -	$ 12.99
Arduino Mega	Elegoo	$ 14.99	1	$ -	$ -	$ 14.99
RFP12N10LMOSFETS	Riddle Electronics	$ 6.95	1	$ -	$ -	$ 6.95
12V 3A AC Adapter	IBERLS	$ 11.89	1	$ -	$ -	$ 11.89
TABLE						
1/4" x 48" x 96" ply	Home Depot	$ 22.92	2	$ 2.98	$ -	$ 48.82
1/2" x 48" x 48" ply	Home Depot	$ 16.08	1	$ 1.05	$ -	$ 17.13
2x2 (leg)	Lowe's	$ 6.30	4	$ 1.64	$ -	$ 26.84
1x4 (inner brace)	Lowe's	$ 7.86	2	$ 1.02	$ -	$ 16.74
1x3 (top frame)	Lowe's	$ 6.76	2	$ 0.88	$ -	$ 14.40
Screws	Lowe's	$ 2.58	3	$ 0.50	$ -	$ 8.24
Nails	Ace Hardware	$ 2.75	1	$ 0.18	$ -	$ 2.93
Acrylic	Professional Plastics	$ 51.99	1	$ 5.46	$ 31.95	$ 89.40
Grand Total						$ 540.07

PROGRESS



Research
95
95
Design
60
70
Acquisition of Parts
50
70
Component Testing / Protoyping
40
40
Final Construction/Programming
30
40
Overall
55
65
0
10
20
30
40
50
60
70
80
90
100
% Complete
Software
Hardware

CURRENT ISSUES

- Creating physical interface/buttons for the table
- Moving the table
- Door vs curtains

- Problems getting Bluetooth to connect (though it does pair)
- Communicating with projector
- Feasibility/usefulness of brightness adjustment

QUESTIONS?