Group 8 S.T.E.P.S.

Style Tracking Expressive Pad System

Arcade Rhythm Game with Pose Detection

Final Presentation

Christopher Solanilla
Blake Whitaker
Kaila Peeples,
Andres Abrams
Jani Jon Lumibao

Meet the Team

Work Distribution

Blake Whitaker

Electrical Engineer

Hardware Lead
PCB Design
Architect Systems & Integration

Christopher Solanilla

Computer Engineer

Project Lead
Computer Vision & Core Software
Game Designer

Kaila Peeples

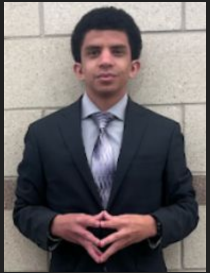
Photonics Engineer

Lens/Design Simulation
Camera Module Integration
Optical/Illumination Optimization
Image Quality Calibration

Jani Jon Lumibao

Computer Engineer

Hardware Assistant
Embedded Programming



Andres Abrams

Computer Engineer

Software Assistant
UI & Game Balance
Development Support

Motivation and Background

- Rhythm games have been of interest to several of our group members for a long time.
- Most notably, the Rhythm game StepManiax featured at UCF’s Knightcade is the most influential one.
- Lack of 9 panel Dance Rhythm Games
- Lack of expression for upper body movement
- StepManiax at UCF’s Knightcade being broken

Goals

Hardware

- Construct a high quality, durable 9-panel dance pad that feels professional
- Ensure instant, accurate feedback for every player step
- Support input with USB HID compatibility for seamless communication within a host PC
- Design for modularity allowing easy maintenance.

Software

- Create a dance rhythm game engine to support our custom 9-panel layout
- Create an engaging rhythm game experience that challenges players of all skill levels
- Implement Computer Vision to track player poses and award them points for doing specific poses
- Implement a robust UI to allow users to make their own charts

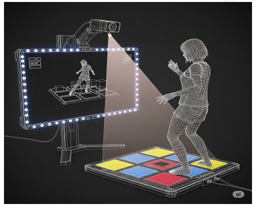
Optics

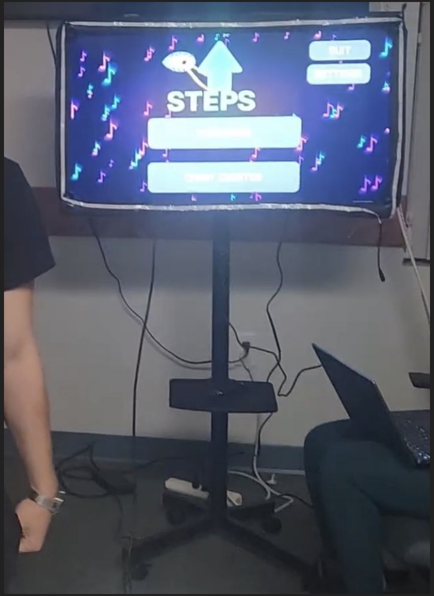
- Design a lighting subsystem to support computer vision in any lighting condition
- Ensure the player will be seen within the field of view
- Have the lens specially designed to aid in computer vision computation

Objectives

- Fabricate a durable 9-panel dance pad using custom PCBs and FSR sensors with input latency within 20-40ms
- Develop a robust rhythm game engine in Godot featuring chart loading, score tracking, and real-time pose tracking
- Design an optical subsystem using a camera and LED array for consistent player tracking in any lighting.
- Track the player’s current pose with MediaPipe to recognize expressive upper-body poses.
- Integrate all hardware, software, and vision components into a cohesive, high quality arcade prototype ready for playing.

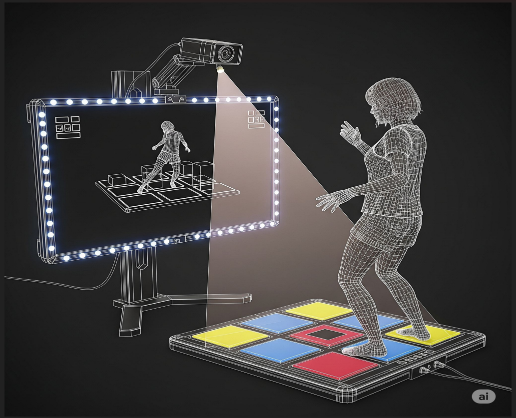
Prototype Design

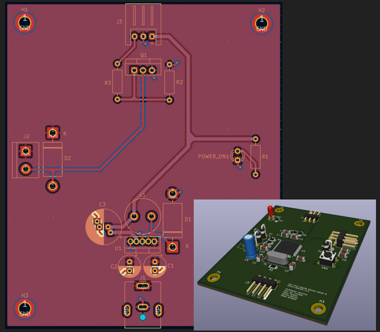


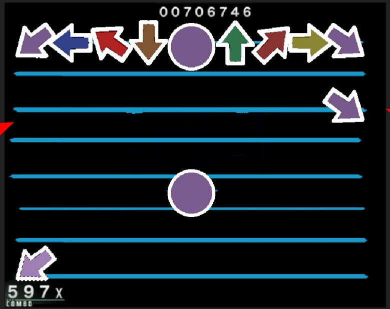


Subsystems Integration

- **Dance Pad** - acts as the main input device for navigation and gameplay
- **Camera Module** - captures real-time player poses during songs
- **Computer Vision** - converts camera data into pose-based inputs
- **Dance Game** - integrates pad and pose inputs to control gameplay and scoring







Engineering Specifications

Parameter	Value
Overall System	
Active power consumption	~20 W
Lighting response time	~5-10 ms
Pose identification accuracy	≥ 95% detection accuracy for body/limb motion
Pose identification response time	~50-80 ms
Dance Pad	
Size	~ 114.3 x 89 x 7.62 cm
Weight	≤ 50 lbs
Cost	≤ $600
Printed Circuit Boards (PCB) Dimensions	
Master Controller	58 x 78.5 x 1.6 mm
Power Hub	80 x 94 x 1.6 mm
Input Tiles	200 x 200 x 1.6 mm
Display	
Size	40 in
Resolution	≥ 1920 x 1080 pixels
Refresh rate	60 Hz

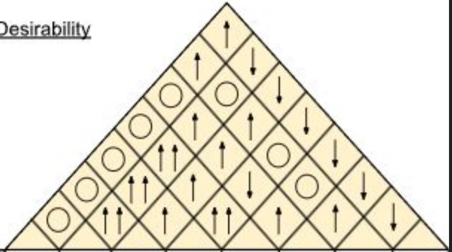
Parameter	Value
Dance Pad Panel	
Size	~254 x 254 x 6.5 mm
FSRs input response time	~20-40 ms
FSRs input detection accuracy	≥ 90%
RGB LEDs PWM duty cycle	~10%-60%
Camera Module	
Full-body coverage area	≥ 2.9 m field width at 1.83 m distance, ensures full-body coverage with no tracking cutoff
Power Supply Unit	
Input voltage from wall power via AC-DC converter	≥ 12 V
Output power	≥ 60W (≥ 5A @ 12V)
Infrared (IR) LED Array	
Illumination sufficiency	Maintain average ROI brightness 75–150 (8-bit) over the play area, achieving ≥95% frame-tracking accuracy across test environments.
Player visibility (shadow coverage)	Full Body at 6ft

House of Quality (HOQ)

- Minimize:
  - Pad Dimensions
  - Pose Identification Input Latency
  - Active Power Consumption
  - Pad Input Latency
- Maximize:
  - Pose Identification Accuracy
  - ROI Brightness
  - Full-body Coverage Area
- Most requirements have either no correlation or strong correlation

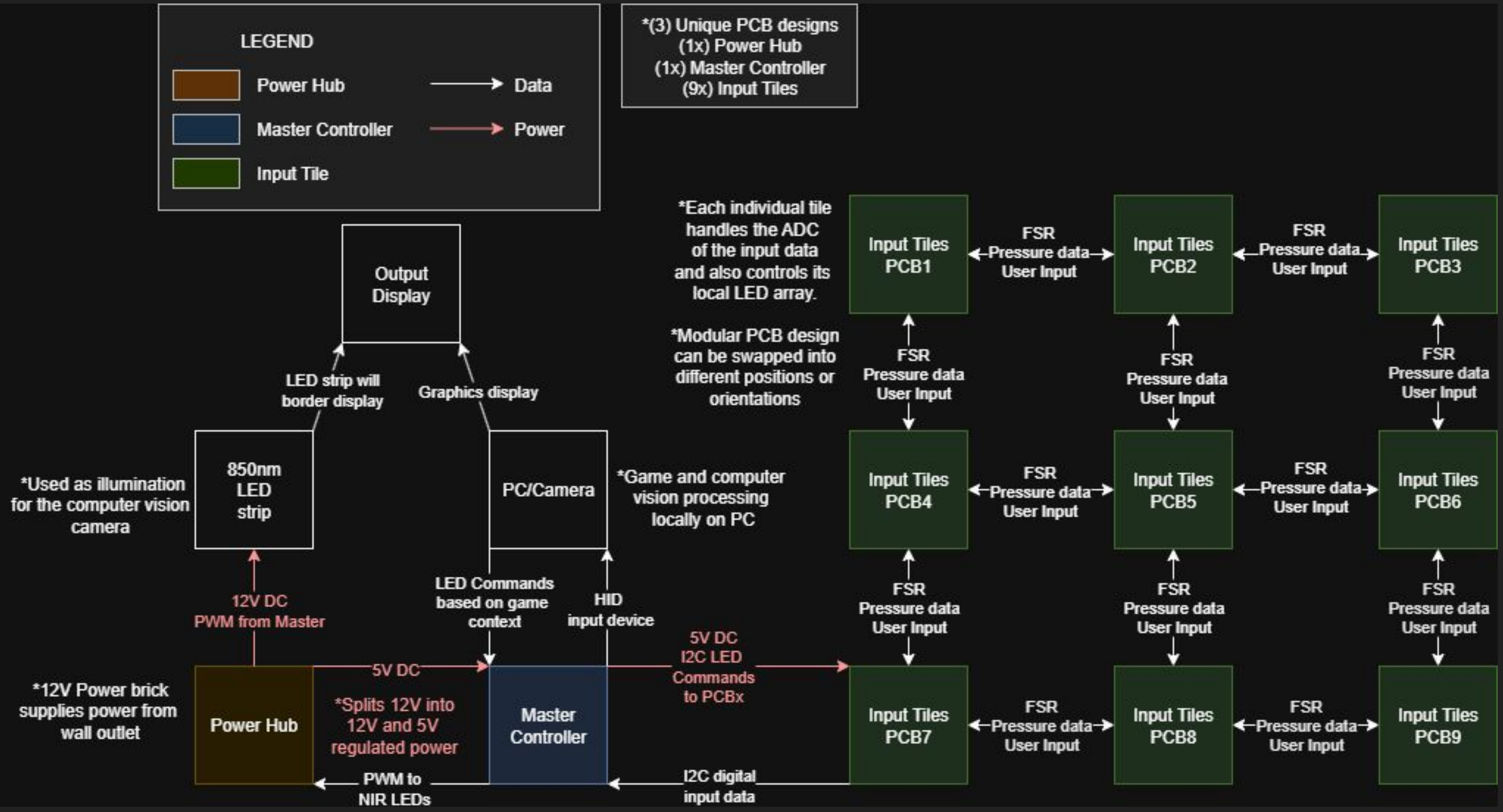
Correlation
- ↑↑ strong positive
- ↑ weak positive
- ○ no correlation
- ↓ weak negative
- ↓↓ strong negative

Direction of Desirability
- ▲ maximize
- ▼ minimize



Marketing Requirements \ Engineering Requirements	Direction of Desirability	Pad Dimensions	Pose Identification Accuracy	Pose Identification Input Latency	ROI Brightness	Full-body Coverage Area	Active Power Consumption	Pad Input Latency	Cost of Materials
Direction of Desirability		▼	▲	▼	▲	▲	▼	▼	▼
Affordability	▲	↑	↓	○	○	○	↑	○	↑↑
Consistency	▲	○	↑↑	↑↑	↑↑	↑↑	↑	↑↑	↓
Durability	▲	↑	○	○	○	○	↑	○	↓↓
Easy Installation	▲	↑↑	○	○	○	○	○	○	↓
Reliability	▲	○	↑↑	↑↑	↑↑	↑↑	↑	↑↑	↓
Maintenance	▲	↑	○	○	○	○	↑	○	↓
Targets for Engineering Requirements		~ 114 x 89 x 7 cm	≥ 95%	≥ 90%	~ 75-150	≥ 2.9 x 1.83 m	~ 20 W	~ 20-40 ms	≤ $1200

Hardware Block Diagram



Development Boards/MCU Comparison

- Analyzed and compared the following development boards series/families:
  - Arduino
  - Teensy
  - Raspberry Pi
  - ESP32
  - STM32
- Selected Arduino Leonardo (ATmega32U4) for master PCB and Arduino Uno (ATmega328P) for slave PCBs (9 total) due to its ease of use and cost, while being sufficient for our design!

Feature	Arduino Uno	Arduino Mega 2560	Arduino Leonardo
Main MCU	**ATmega328P**	**ATmega2560**	**ATmega32U4**
Clock Speed	16 MHz	16 MHz	16 MHz
Flash Memory	32 KB	256 KB	32 KB
SRAM	2 KB	8 KB	2.5 KB
EEPROM	1 KB	4 KB	1 KB
GPIO Pins	14	54	20
PWM Channels	6	15	7
ADC Inputs	6	16	12
USB Communication	serial-to-USB	serial-to-USB	native USB
USB HID Support	none	none	included
Serial Ports (UART)	1	4	1
Active Power Consumption	Moderate	Moderate-High	Moderate
MCU Chip Cost	~$3	~$8	~$4

Communication Protocol

Feature	UART	SPI	I2C
Pin Count	2 (per device)	4 + (N slaves)	2 (total)
Speed	Low	Very High	Moderate
Multi-Device Support	No (Point-to-Point)	Yes (High pin count)	Yes (Address-based)
Complexity	Low	Moderate	Moderate
Project Suitability	Unsuitable	Possible but not ideal	Optimal

- I^2C Communication was selected primarily due to its ability to connect all input tiles while minimizing the pin count on the master to only 2 pins (SDA and SCL)

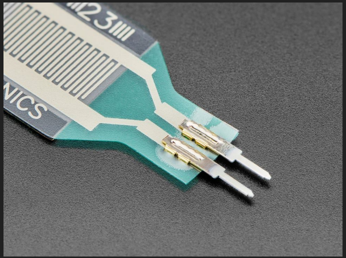
Dance Pad Sensor Comparison and Selection

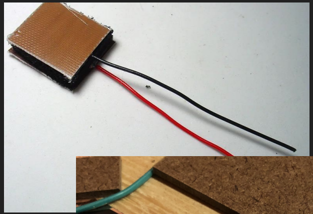
- Force-Sensing Resistors (FSRs) were selected mainly for the following reasons:
  - Affordability
  - Decent Response Time
  - Ease of use
  - Can be used to send both impact and sustain input data

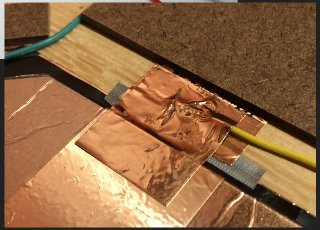
Feature	FSRs	Load Cells	Strain Gauges (Raw)	Piezoelectric Sensors	Break Beam Sensors
Size	↓↓	↑	↓↓	↓	M
Cost	↓↓	↑	↓	↓	↓
Response Time	↑	↑	↑	↑↑ (impact)	↑↑
Accuracy	↓	↑↑	↑↑	↑ (impact)	↓
Complexity	↓↓	↑	↑↑	M	↓
Durability	↓	↑↑	M	↓	↑
Impact Input	✓	✓	✓	✓	✓
Sustain Input	✓	✓	✓		

Force Sensing Resistor Comparison

- FSR Model 408 from Interlink Electronics were extremely affordable and reliable
- FSR Model 408 excels in most features compared to the DIY route







Feature	FSR Model 408 (300 mm)	DIY Conductive Material + Velostat
Cost per Sensor	(≥$3)	(~$3)
Customizability	↓	↑
Assembly Effort	↓	↑
Reliability	↑	M
Sensitivity Consistency	↑	M
Dead Zone Risk	↓	M
Durability	↑	↓
Ease of Replacement	M	↑

RGB vs. Monochrome Sensor Comparison

- Monochrome:sharper, more sensitive, but not AI compatible
- RGB: Lower sensitivity, larger data, but required by MediaPipe
- RGB enables background/limb separation
- Chosen: RGB sensor for AI Pose tracking

Attribute	Monochrome Sensor	RGB Sensor
Image Contrast & Sharpness	Higher (no color filter array)	Lower due to color filter array (CFA)
Light Sensitivity	Higher(no CFA, more light per pixel)	Lower (CFA reduces incoming light)
Data Size	Smaller (single channel)	Larger (3 channels: R, G, B)
Computational Load	Lower (less data to process)	Higher (more data to process)
AI Compatibility	Not supported by MediaPipe (requires RGB input)	Fully supported and optimized for MediaPipe
Background Differentiation	Poor (no color distinction between limbs, background)	Good (color cues help separate limbs and surroundings
Suitability	Not suitable due to MediaPipe incompatibility	Selected for pose tracking and AI model compatibility

Camera Comparison

- Pi HQ : high resolution, rolling shutter, low FPS
- Arducam AR0234: global shutter, high FPS, costly
- SVPORO AR0234: same performance, lower price
- Chosen: SVPRO AR0234 for best value

Camera	Pixel Pitch(μm)	Frame Rate (FPS)	Price (USD)	Shutter Type
Raspberry Pi HQ (IMX477)	1.55	20	$53.78	Rolling
Arducam AR0234 Global Shutter	3.0	60	$109.99	Global
SVPRO AR0234 Global Shutter	3.0	60	$76.99	Global

Illumination Method Comparison

- RGB LEDs: visible, glare, inconsistent
- White LEDs: bright, discomfort/glare
- 850 nm IR LEDs: little to no discomfort/glare for player,efficient, reliable
- Chosen: 850nm IR LEDs for comfort & performance

Attribute	RGB LEDs	White LEDs	850nm IR LEDS
Visibility to Player	Fully visible	Bright and visible	Partially or mostly invisible
Comfort / Glare	Moderate (color shifting may distract)	Potential discomfort in dark environment	Comfortable (no glare)
MediaPipe Compatibility	Inconsistent under varied RGB output	High contrast but could saturate camera	Reliable for pose detection
Power Efficiency	Lower (color mixing requires more power	Moderate	High (simple constant voltage)
Ease of Integration	Moderate (requires careful color control)	Easy	Easy 12 V strips
Camera Compatibility	Compatible (RGB input)	Compatible	Compatible (requires no IR-cut filter)
Beam Shaping/ Directionality	Flexible with lenses or domes	Fixed	Slightly less flexible but evenly distributed
Cost and Availability	Moderate to High	Low	Moderate and widely available
Chosen Option for S.T.E.P	Rejected due to inconsistency	Rejected due to glare	Selected for performance and comfort

LED Strip Comparison

- Compared three IR strip options
- 120° beam angle for wide coverage
- 360 Digital Signage: best balance of cost, power, and density
- Chosen: 360 Digital Signage 850 nm strips

Feature	Waveform IRFlex 850nm	DC12/24V	360 Digital Signage
LED Type	2835 SMD	5050 SMD	3528 SMD
Beam Angle	120	120	120
LED Density (/m)	120	60	60
Power(W/m)	9.6	14.4	7.2
Price ($)	$55	$28.98	$45.89

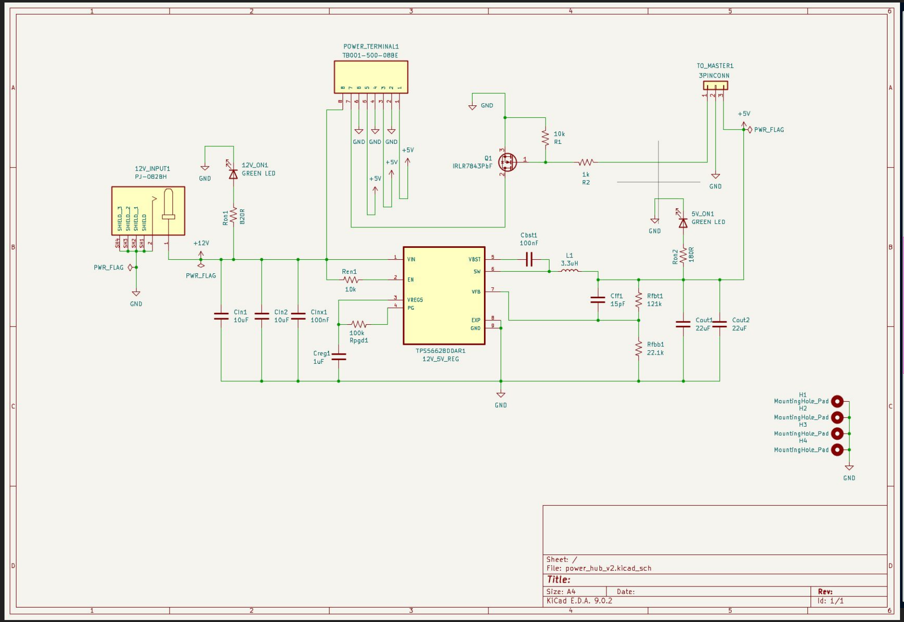
Power Distribution Table

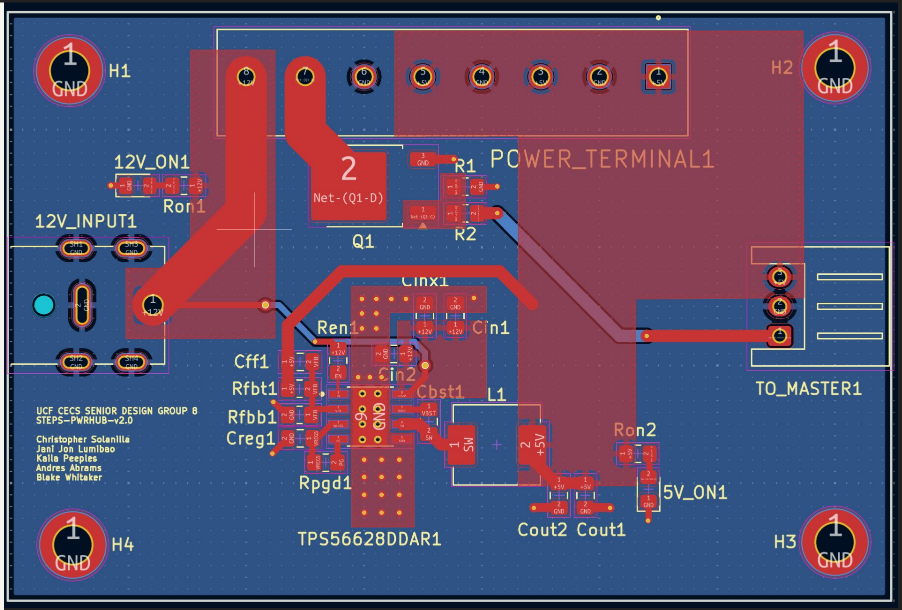
Subsystem	Rail Voltage (V)	Nominal Current (A)	Nominal Power (W)	Notes
NIR LED Subsystem	12	3.8	45.6	PWM-dimmed via MOSFET at ~50% duty
Master Controller (ATmega32u4)	5	0.05	0.25	Main controller logic; relatively constant load.
Column 1: 27 WS2812B LEDs + 3× ATmega328PB	5	1	5	Typical animations and brightness limits
Column 2: 27 WS2812B LEDs + 3× ATmega328PB	5	1	5	Identical to Column 1.
Column 3: 27 WS2812B LEDs + 3× ATmega328PB	5	1	5	Identical to Column 1
System Total — 12 V Rail	—	**3.85 A**	**46.2 W**	NIR LEDs + Leonardo under normal operation.
System Total — 5 V Rails (all columns)	—	**3.00 A**	**15.0 W**	Three separate 5 V supplies at ~1.0 A each in typical use.
Overall System Total (nominal)	—	**6.85 A**	**~61 W**	Real-world power based on typical duty cycles and brightness.
Overall System Total (peak)	—	**8.00 A**	**~64 W**	Peak of ~8A total due to Dance Pad Tiles changing brightness upon user input.

Power Supply Unit

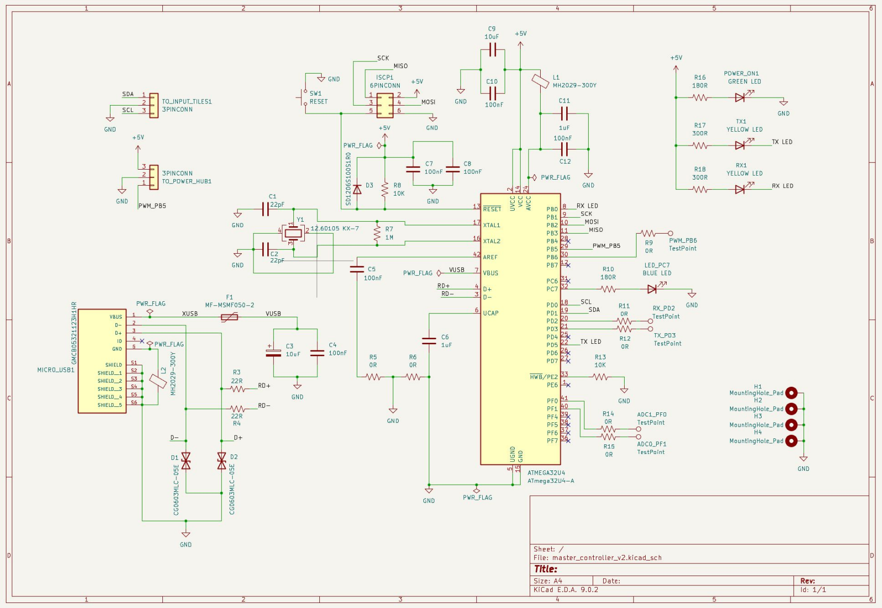
The maximum current draw at full brightness of both the NIR LED subsystem and the Dance Pad LEDs is just under 10 A. With this in mind, we chose a 12 V, 10 A power supply to accommodate the combined load with some overhead. While the theoretical maximum current is ~10 A, normal operation at a 50% duty cycle for both subsystems reduces the expected draw to roughly 7 A with a peak of 8 A due to Dance Pad Tiles changing brightness upon user input.

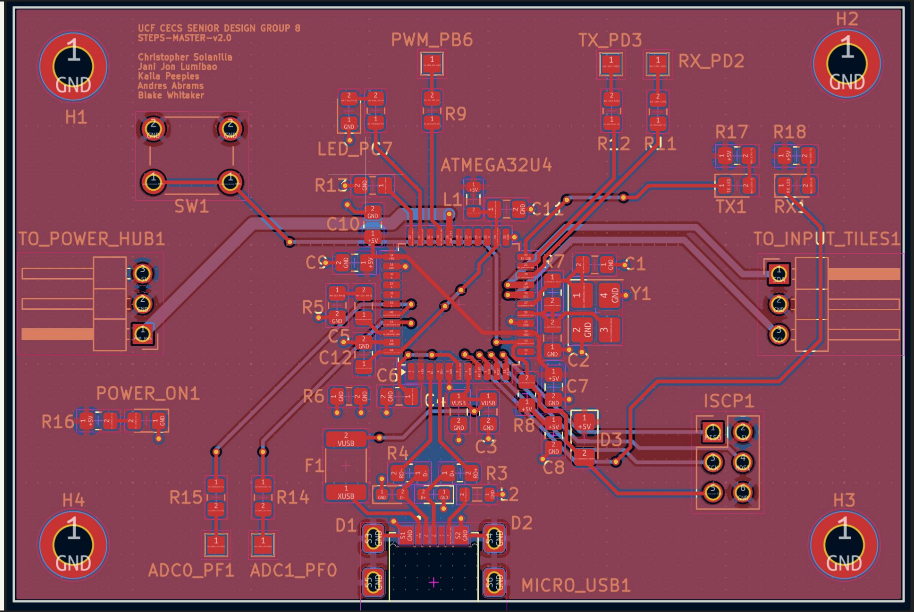
PCB Design - Power Hub



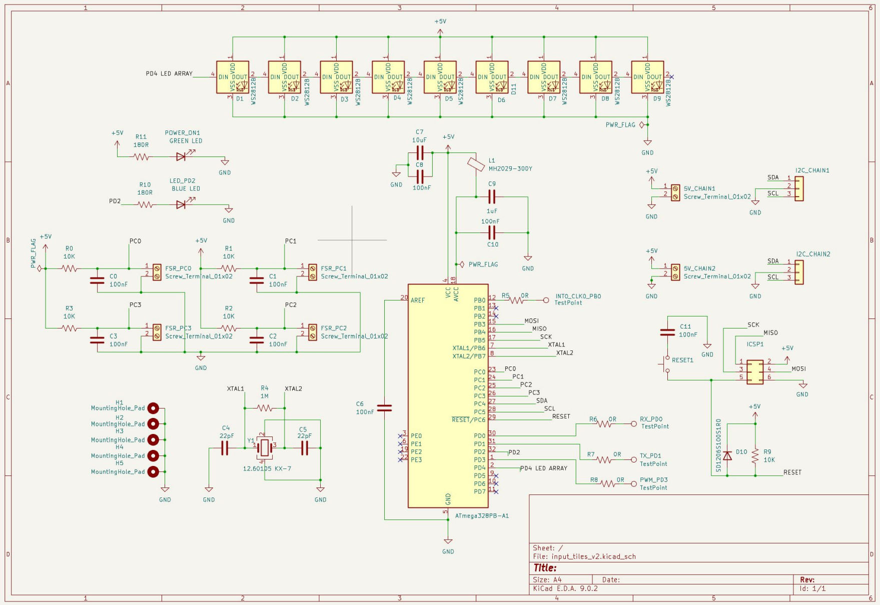


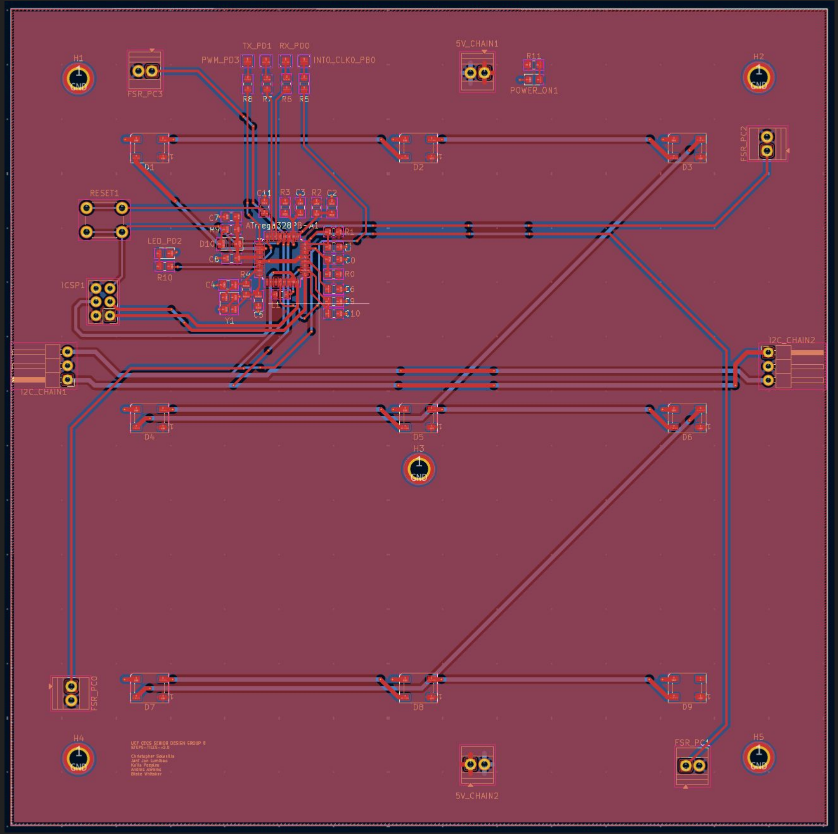
PCB Design - Master Controller





PCB Design - Input Tiles





Computer Vision Comparison

- For our project, MediaPipe was selected for our main Computer Vision Framework.
  - Perfect for our use case in getting single target poses fast accurately
  - OpenPose a potential contender but still largely in development, would be good for 2 player mode
  - Based on BlazePose

Tool	Speed	Accuracy	Hardware Requirement	Multi-person Support	Best Use Case
OpenCV	Fast	Low to Medium	Works on any CPU	No	Basic gesture logic, simple demos
OpenPose	Slow	High	Requires powerful GPU	Yes	Research, motion capture
MediaPipe	Fast	Medium to High	Runs on CPU/mobile	No	Mobile apps, real-time interaction
BlazePose	Very Fast	Medium	Optimized for CPUs	No	Fitness apps, AR, fast input

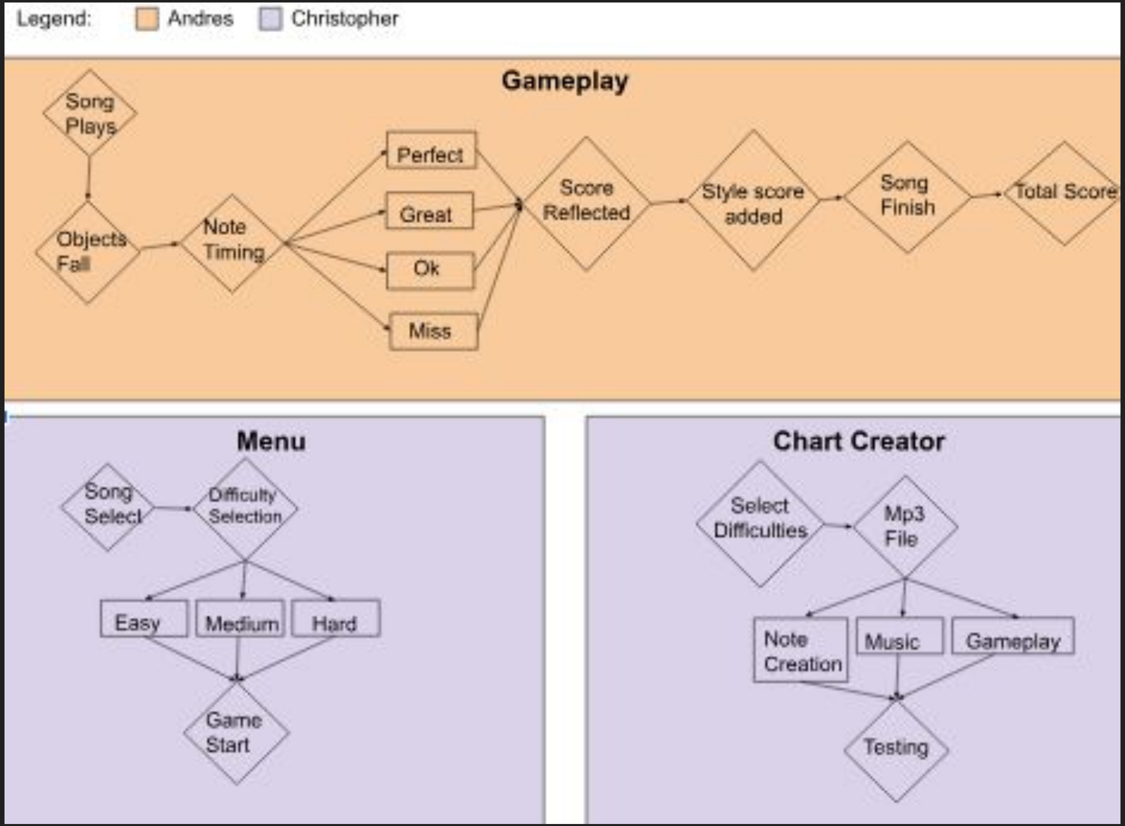
Game Programming

	Game Engine	Custom Coding	Web-Based
Ease of Use	Easy	Hard	Medium
Flexibility	Medium	Large	Small
Learning Curve	Medium	Large	Large
Development Curve	Small	Medium	Medium

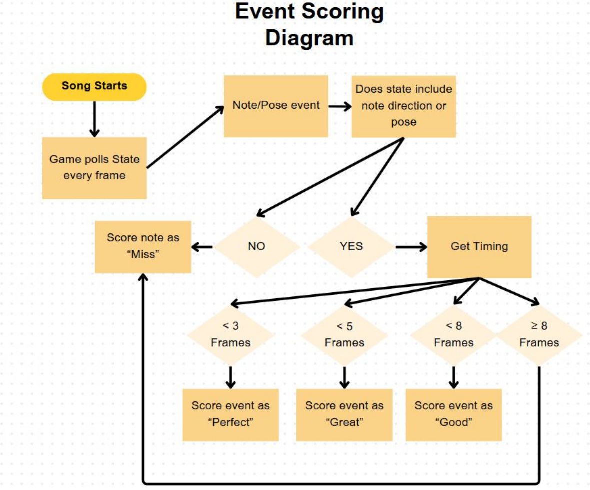
Game Engine Comparisons

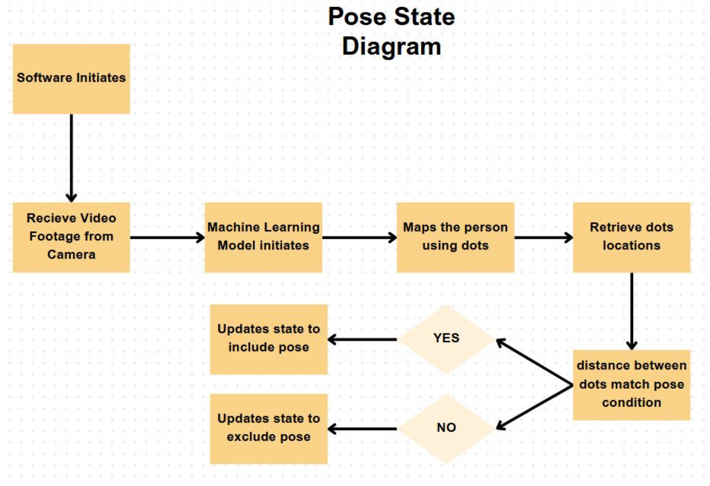
	Unity	Godot	Unreal Engine
2D Capabilities	Primarily 3D, supports 2D	2D & 3D games (strong 2D support)	Primarily 3D, limited 2D via Paper2D
Cost	Paid, university students have a free version	Open Source, Free Download	Free through the Epic Games Launcher
Documentation	Extensive, hard to find, and hard to understand	Open source, easy to understand, has examples	Complicated, precise, and very extensive
Tutorial Availability	Mostly 3D tutorials available, minimal 2D	Extensive Tutorials for variety of tools in 2D	Focused on 3D tutorials, 2D tutorials are scarce
Installation/Startup	Long time to install and set up, has heavy load	Quick and easy both install and setup, lightweight	Very heavy install duration and heavy load
Version Control Support	Built in paid version control	External version control needed	Built in version control free
Microcontroller Friendly	No	Compatibility mode able to be used with weaker hardware	No

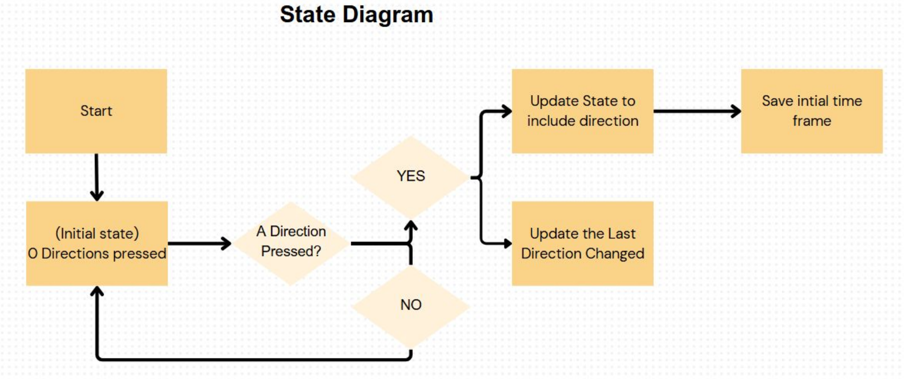
Gameplay Flowchart

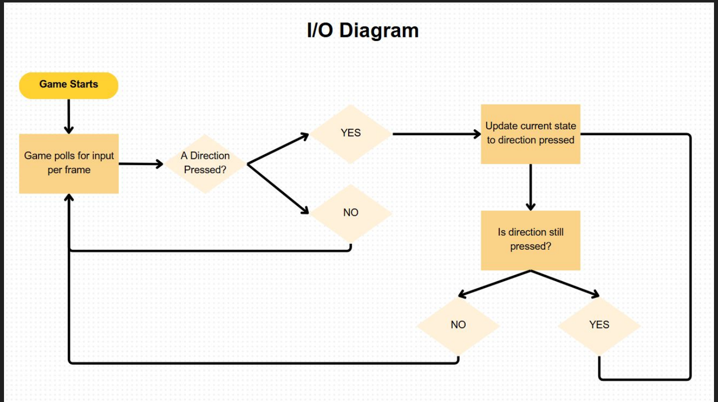


State Flowcharts



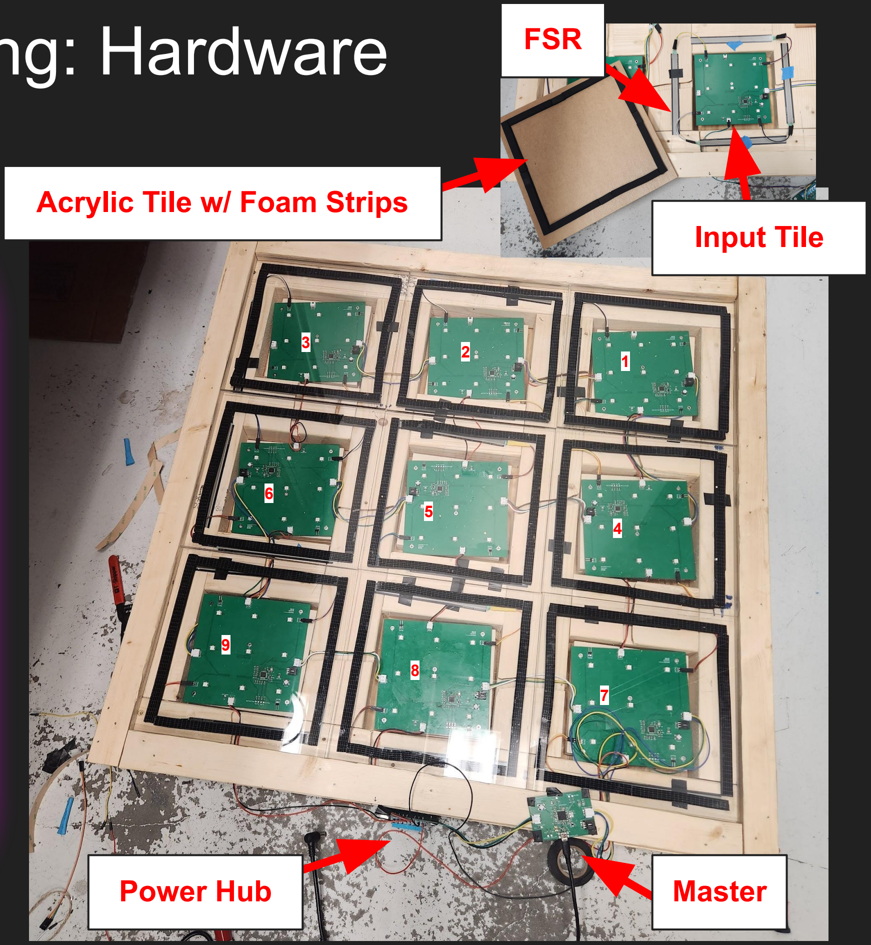






Prototyping and Testing: Hardware

- Each column share their own 5V line from the power hub ([7,4,1], [8,5,2], [9,6,1])
- Each input tile has 4 FSR strips around the edges aligned with the foam strips of the acrylic tile (threshold of FSRs adjusted accordingly so only stepping on the tile triggers input; little to no dead zone risk
- RGB LEDs within each input tile consistently light up as visual feedback that at least one of four FSRs are detecting input
- Master polls through each input tile for their slave address for keyboard functionality (i.e. 0x01='1', 0x02='2', etc.)
- Time it takes for a press on the tile to register as keyboard input is almost instantaneous
- Subsystem Success: Met goals for dance pad input response time of around 20-40ms



Prototyping and Testing: Optical & Illumination systems

- Lens Testing : 2.8 mm, 3.2 mm, 4 mm lenses evaluated → 3.2 mm f/2.4 selected for balance of FOV and sharpness
- Depth of Field: Validated 1.8 m -3 m range → all player zones in focus during motion
- Illumination Spec : Brightness-based (75 - 180 8-bit) achieving ≥ 95% tracking accuracy
- IR Sensitivity: Confirmed at 850 nm for camera and lens combination
- LED System : Bright and effective illumination, minor heat buildup under prolonged use.
- Subsystem Success: Met goals for FOV, DOF, resolution, and brightness specifications.

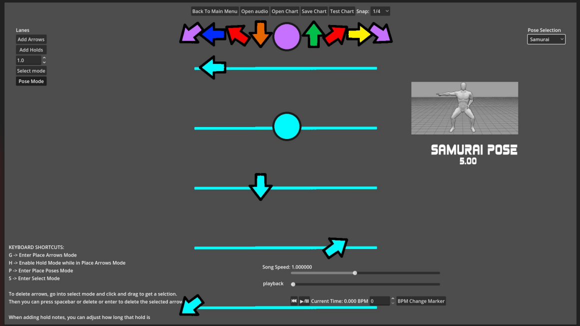
Prototyping and Testing: Software

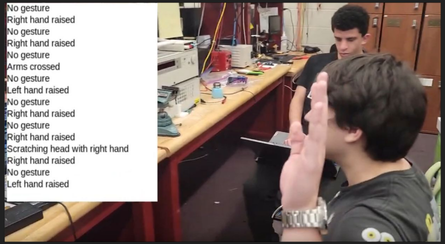
The 2 main aspects of our software include the video game, and the computer vision sub system.

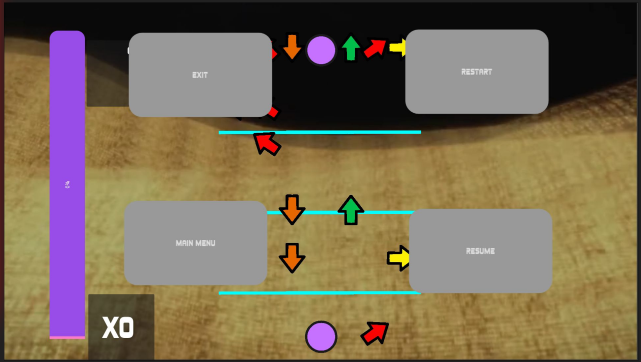
For the video game, we were able to prototype and test rapidly by making our own custom rhythm game engine and testing suite

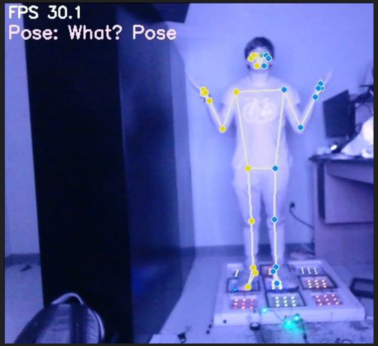
For the Computer Vision software, we were able to prototype quickly using MediaPipes python package as well as making our own GUI for testing.

We converted the logic to C++ and compiled builds once we finished testing.

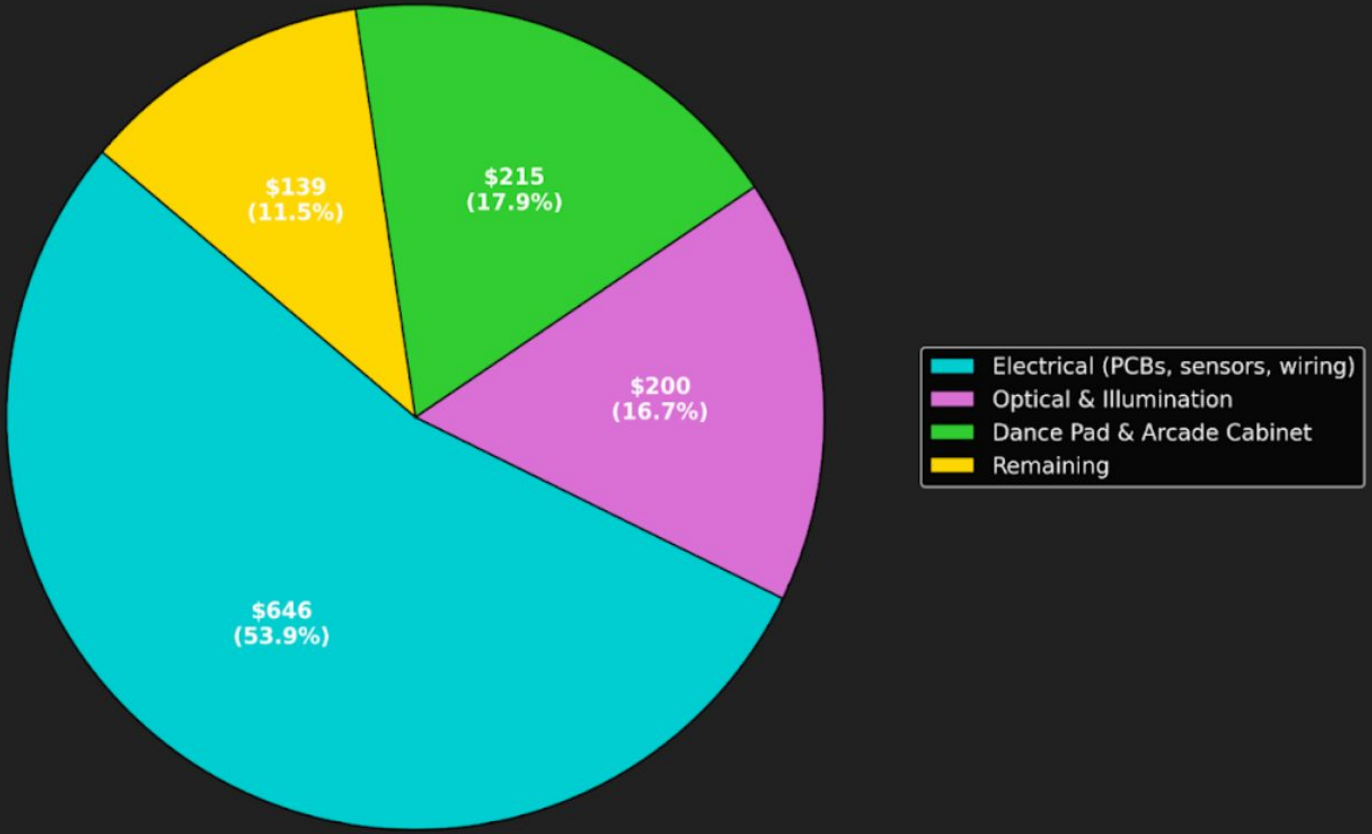








Budget



Total Budget:$1200