Engineering Project Portfolio

Lucas Flowers

Phoenix, AZ | (602) 784-2043 | contact@lucasflowers.net
linkedin.com/in/lucas-flowers-phx | lucasflowers.net

This portfolio highlights selected engineering projects that demonstrate systems thinking, technical depth, leadership, and human-centered design.

About Me

Over the past four years, I have explored engineering through hands-on, interdisciplinary experiences at the intersection of software, hardware, and systems design. My work has been shaped most strongly by robotics, where I served as the lead programmer across multiple FIRST Robotics Competition teams and ultimately founded my own team, Team Epsilon. Through these experiences, I learned how technical decisions, team structure, and real-world constraints combine to shape successful engineering outcomes.

Beyond robotics, I have pursued opportunities that emphasized both technical depth and human impact. I studied kinematics, haptics, and assistive robotics through a pre-college program at Wentworth Institute of Technology, contributed to an assistive technology project as a state finalist in Makers of Change, and gained hands-on fabrication experience through my internship at Phoenix Forge. These projects pushed me to move beyond individual components and think in terms of complete systems, integrating software, electronics, mechanics, and user needs.

This portfolio highlights selected projects that reflect how I approach engineering: learning by building, iterating through failure, and designing systems that are both technically rigorous and purposeful. Together, these experiences have shaped my interest in systems engineering and autonomous systems, where thoughtful integration and responsibility-driven design are central to meaningful innovation.

FRC Team Epsilon

May 2025 - Present
Founder, President & Lead Programmer

Project Overview

Team Epsilon is a student-founded, student-led FIRST Robotics Competition team based in North Phoenix, Arizona. I co-founded the team in 2025 to create a small, hands-on, and technically rigorous engineering environment that more closely mirrors real-world engineering teams.

After multiple seasons on school-based robotics teams, my co-founder and I identified limitations in design freedom, purchasing timelines, and student leadership opportunities imposed by district systems. We decided to take full responsibility for building a team from the ground up. Starting out of a friend's garage, we recruited members across multiple schools, developed a business plan, and formally registered Team Epsilon as a nonprofit organization.

From the beginning, we structured the team to operate like a startup: students own projects end-to-end, collaborate across disciplines, and are accountable for both technical and organizational decisions. This approach allows members to gain experience not only in robotics engineering, but also in systems integration, project management, fundraising, and leadership.

As a rookie team, we have already built a full practice robot during the off-season, grown to over **12 active members, and raised more than $20,000 in funding.** As we enter our first competitive season, Team Epsilon reflects my belief that engineering education is most effective when students are trusted with responsibility, encouraged to take risks, and supported through mentorship.

Team Epsilon first meeting

Problem & Design Challenge

Our primary challenge was designing an engineering environment that provided meaningful responsibility and technical depth for students across Arizona. In many traditional high school engineering classes and large, school-based FRC teams, students often specialize narrowly or have limited ownership over complete systems due to team size, scheduling constraints, and administrative limitations.

We believe students learn engineering most effectively when they are trusted with real responsibility, held to clear expectations, and exposed to the realities of collaborative, deadline-driven work. To address this, we intentionally designed Team Epsilon to operate like a small engineering startup. Team members are expected to take ownership of projects, collaborate across disciplines, and follow through on commitments that directly affect team performance.

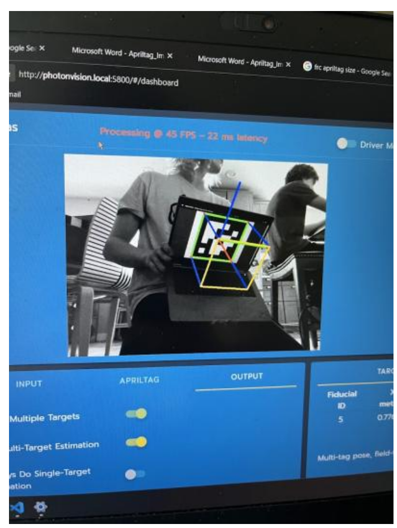
The design challenge was not only technical, building a competitive robot, but organizational: creating a structure that balances rigor with mentorship, accountability with support, and ambition with sustainability. This approach allows students to experience problem-solving, iteration, and responsibility that define real-world engineering while maintaining a learning-focused environment.

My Role & Responsibilities

As founder, president, and lead programmer of Team Epsilon, I have taken primary responsibility for both the technical and organizational development of the team. I created the team's business plan, programmed and maintained the team website, and developed its branding and merchandise. I led grant writing and fundraising efforts to secure startup funding and assisted in developing and managing the team budget.

I manage team operations by maintaining meeting agendas, communicating announcements and schedules, and coordinating regular team meetings to ensure alignment across sub teams. In addition, I oversee project planning and delegation, ensuring each member has clearly defined responsibilities and that technical work progresses on schedule. This role requires balancing engineering leadership with communication and financial planning, closely mirroring the responsibilities of leading a small engineering organization.

So far, my most significant technical contribution has been developing the vision-based auto-alignment system for our off-season robot. The system uses two cameras running on a dedicated coprocessor to dynamically detect fiducial markers and estimate the robot's pose in real time. When the operator activates the system, the robot automatically aligns itself based on these pose calculations, improving accuracy and consistency during operation.



Fiducial apriltag being detected by the coprocessor

Implementing this system required careful calibration, tuning, and integration with the robot's control software. To work efficiently despite limited physical access to the robot (particularly while the mechanical team was still assembling the chassis) I leveraged a simulation environment to develop and test the vision pipeline and control logic. This approach allowed me to iterate on the system early and significantly reduce integration issues once on-robot testing began.

Results & Impact


2025 off-season robot fully assembled

During the off-season, our team built a fully functional practice robot to allow new members to develop technical skills before the official competition season began. Beginning in the summer, we designed and competed with this robot in the Arizona Robotics League (ARL), a series of off-season competitions hosted by local high schools across Arizona. The league includes monthly events followed by a championship tournament in October that uses a draft-based alliance selection system.

Team Epsilon competed in three ARL events and was drafted to compete in the championship tournament—the only rookie team to be selected. At the championship event, **we received the Breakthrough Award**, recognizing Team Epsilon as the top emerging rookie team in Arizona.

Reflection & Next Steps

As the competition season approaches, I am excited to lead Team Epsilon through a full engineering design cycle under competitive constraints. This year, we are registered to compete at the North Arizona Regional and the Utah Regional, where our focus will be on applying what we learned during the off-season to achieve higher performance and reliability.

Seeing the idea of creating an independent, student-led team grow into a functioning and competitive organization has been especially meaningful. Looking ahead, I plan to develop a sustainability and leadership transition plan to ensure that Team Epsilon continues to grow and operate successfully after I graduate.

Makers of Change — Assistive Technology Design (State Finalist)

The Bluey Jamboard
State Finalist (Top 9 of 95+ Teams)

Hosted by Southwest Human Development
October 2025
Role: Embedded Systems Specialist

Project Overview

Makers of Change is a statewide assistive technology design challenge focused on creating custom solutions for individuals with disabilities. Our team designed and built an interactive game board for Levi, a child with cerebral palsy and related motor and processing challenges. The goal was to create an engaging, accessible device that encouraged interaction, fine motor practice, and inclusive play for Levi and his siblings.

Our project was selected as a **state finalist (Top 9 out of 95+ teams across Arizona)**, recognizing both its technical execution and its impact-driven design.

Problem & Design Challenge

Levi experiences cerebral palsy, hypertonia in his legs, hypotonia in his torso, and challenges with fine motor control and language processing. Many off-the-shelf games are not accessible to children with these needs, either requiring precise motor control or failing to sustain engagement.

The design challenge was to create a device that balanced accessibility, durability, and engagement within a limited time and budget. The solution needed to be easy to use, visually appealing, and adaptable to Levi's physical abilities, while also being safe and enjoyable for shared play with siblings.

My Role & Responsibilities

As the lead programmer and a primary technical contributor, I was responsible for both software development and key hardware integration, including:

- Programming interactive games on the Raspberry Pi with accessibility and engagement as core design goals
- Wiring physical button inputs using the Raspberry Pi's GPIO pins
- Managing power distribution and system startup for the Raspberry Pi
- Integrating and configuring the display, including troubleshooting screen compatibility and resolution issues
- Collaborating with teammates to ensure alignment between the physical enclosure and system behavior

Process

Our team began by researching Levi's diagnoses and understanding how his physical and cognitive needs affected everyday interactions. We brainstormed multiple concepts and used CAD tools to design the physical layout of the device.

Due to time constraints, we pivoted from building a full table to creating a tabletop interactive game board. I initially programmed a balloon-popping game, but after testing, we realized it lacked sustained engagement. I then reprogrammed the system to run a **multi-player basketball game** with large, easy-to-press buttons to better support motor accessibility and cooperative play.

In addition to software development, I handled much of the system integration, including wiring the buttons, managing power to the Raspberry Pi, and configuring the display. Setting up the screen required troubleshooting hardware compatibility and display settings to ensure reliable operation within the custom enclosure. This close integration between hardware and software allowed the device to function as a cohesive, user-friendly system.

Results

The final device successfully met its design goals by providing an accessible, engaging game that Levi could use independently or with his siblings. The system's large buttons, clear visual feedback, and simple game mechanics supported usability while maintaining enjoyment.

Our project was recognized at the state level as a **Top 9 finalist out of more than 95 teams**, validating both the engineering approach and the project's social impact.

Reflection

This project reinforced the importance of designing technology around real users rather than abstract requirements. Small engineering decisions—such as button placement, display configuration, and game pacing—had a significant impact on accessibility and engagement.

If given more time for this project, I would add configurable difficulty levels and additional game modes to better adapt the device to different users and abilities. This experience strengthened my interest in human-centered engineering and showed me how thoughtful integration of hardware and software can directly improve quality of life.

Minecraft Server Wizard — CS50 Final Project

Final project for CS50
December 2023
Demo video

Project Overview

Minecraft Server Wizard is a command-line tool designed to simplify the installation and management of Minecraft servers on Windows. Developed as my CS50 Final Project, the tool abstracts the complexity of manual server setup into a guided, interactive workflow. By automating tasks such as server installation, Java configuration, plugin management, and backups, the project makes server hosting accessible to users without advanced technical knowledge.

Problem & Design Challenge

Setting up a Minecraft server typically requires manual installation of Java, downloading server JAR files, editing configuration files, managing plugins, and maintaining backups. These steps are error-prone and intimidating for many users, especially those without experience working in command-line environments.

The design challenge was to reduce this complexity while maintaining flexibility and control. The tool needed to handle system dependencies, user input, and external APIs reliably, all while providing clear feedback and preventing common configuration errors. Additionally, the project had to be packaged as a standalone executable for ease of use.

My Role & Responsibilities

I independently designed and implemented the entire project, including:

- Designing the command-line interface and user interaction flow
- Automating server installation, Java detection, and memory allocation
- Integrating external APIs to fetch and install the latest server JAR files
- Implementing plugin installation, updating, enabling/disabling, and version tracking
- Creating automated backup functionality with timestamped file management
- Packaging the application into a standalone Windows executable using PyInstaller
- Writing documentation and producing a video demonstration for CS50

Process & Development

I began by mapping the manual steps required to set up a Minecraft server and identifying opportunities for automation. Using Python, I built a modular command-based system that guides users through installation and server configuration.

The tool dynamically retrieves server builds from external APIs and installs Java when necessary. Plugin management is handled through web scraping and version comparison, allowing the tool to detect and update outdated plugins automatically. To improve usability, I added built-in help commands and clear error messages.

Once the core functionality was complete, I packaged the application into a single executable using PyInstaller, allowing users to run the program without installing Python or dependencies. I tested the tool across multiple configurations and documented common troubleshooting scenarios.

Results & Impact

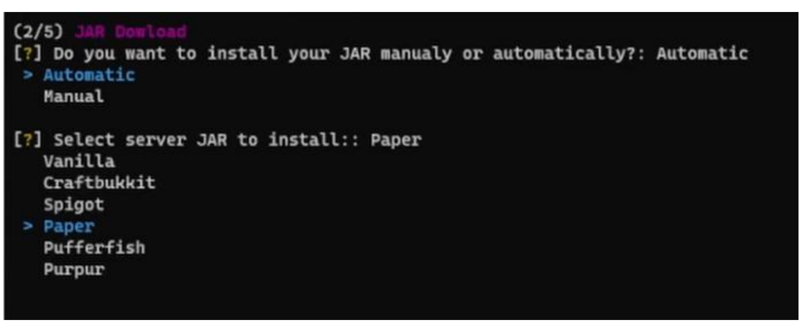
The final tool successfully automates the full lifecycle of a Minecraft server, from installation to ongoing management. Users can install servers, manage plugins, create backups, and start or stop servers using simple commands.

The project demonstrates how complex system administration tasks can be abstracted into an accessible interface without sacrificing functionality. It also reflects my ability to design software that prioritizes user experience while managing real-world constraints such as system compatibility and dependency management.

Reflection & Next Steps

This project strengthened my understanding of software abstraction, automation, and user-centered design. I learned how to design tools that anticipate user errors and simplify complex workflows without hiding important functionality.

If extended, I would add cross-platform support for Linux, improve error handling for edge cases, and implement a graphical interface for users less comfortable with command-line tools. This project confirmed my interest in building tools that make complex systems more accessible and reliable.



Easy terminal GUI tool

Physics Projectile Motion Simulator

https://github.com/lflowers01/projectile-sim
https://www.lucasflowers.net/projectile-sim
May 2025

Project Overview

This project is a web-based interactive projectile motion simulator developed as a final project for my Physics course. The goal was to translate the kinematic equations of projectile motion into a real-time, visual simulation that allows users to explore how force, launch angle, height, and mass affect a projectile's trajectory.

Rather than producing static graphs or calculations, I designed the simulator to be fully interactive, enabling users to manipulate parameters and observe the resulting motion, velocity vectors, and displacement in real time.

Problem & Design Challenge

Projectile motion is often taught abstractly through equations, which can make it difficult for students to build intuition about how variables interact. The challenge was to create a simulation that was both physically accurate and intuitive, while remaining responsive and visually clear.

From an engineering perspective, the simulation needed to remain stable regardless of frame rate, support multiple simultaneous projectiles, and clearly communicate both motion and underlying physics concepts without overwhelming the user.

My Role & Responsibilities

I independently designed and implemented the entire simulation, including:

- Modeling projectile motion using kinematic equations and real-time integration
- Designing object-oriented classes for projectiles, aiming mechanics, and vector visualization
- Implementing gravity and time-based updates using frame-rate-independent calculations
- Creating an interactive aiming system that converts user input into force and angle vectors
- Visualizing velocity components and full trajectories for educational clarity
- Designing a UI with sliders and controls for mass, height, and force

Process & Development

I structured the simulation using a modular, object-oriented approach. Each projectile tracks its own position, velocity, and trajectory history, while gravity is applied incrementally using a normalized time step based on deltaTime. This ensures consistent physics behavior even under varying performance conditions.

To improve conceptual understanding, I visualized both the horizontal and vertical velocity components as vectors and rendered the projectile's full trajectory path. An interactive aiming system allows users to adjust launch angle and force by dragging a control point, providing immediate visual feedback.

Throughout development, I iteratively tested parameter ranges and scaling factors to balance realism, performance, and usability.

Results & Impact

The final simulator accurately models projectile motion and allows users to experiment with different launch conditions in real time. Interactive design makes abstract physics concepts more intuitive by directly linking mathematical parameters to visual outcomes.

This project helped reinforce my understanding of physics, numerical simulation, and user-focused software design, and it demonstrated how engineering tools can be used to enhance learning rather than simply calculate results.

Reflection & Next Steps

This project strengthened my ability to translate mathematical models into functional software systems and highlighted the importance of thoughtful visualization when communicating technical concepts.

If extended, I would add features such as air resistance, energy analysis, and data export for quantitative comparison. I would also explore separating the physics engine from the rendering layer to support more advanced simulations.

High-Altitude Weather Balloon Camera System (In Progress)

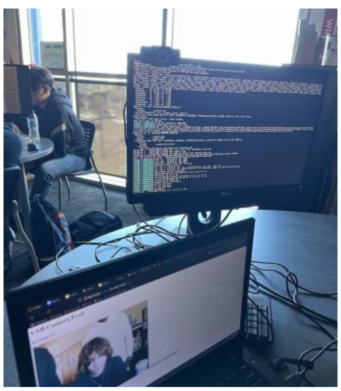
Bioscience E-tech hiking club
October 2025 – Present

Project Overview

This project is an in-progress high-altitude weather balloon camera system developed for my Engineering III–IV course and my school's E-Tech Club. The goal is to capture high-altitude imagery of Earth and near-space using a lightweight, reliable camera payload carried by a weather balloon. Previous launches relied on consumer action cameras, which frequently failed due to power loss, storage issues, or environmental conditions. To improve reliability and control, I am designing a custom camera system built around a Raspberry Pi and multiple lightweight cameras.

Rather than prioritizing continuous high-frame-rate video, the system is designed to capture clear, incremental images over the duration of the flight, producing a timelapse or low-FPS video that maximizes reliability and image quality under extreme conditions.

Problem & Design Challenge



Prototype camera recording working with webserver

The primary challenge is designing a camera system that can operate reliably for **4–6 hours** in a high-altitude environment while remaining under a **strict 4-pound payload limit**. At altitude, the system must tolerate cold temperatures, limited power availability, and the inability to physically intervene once launched.

Additional constraints include ensuring continuous recording without file corruption, minimizing power consumption, and preventing software or hardware failures that could result in data loss. These requirements necessitate careful tradeoffs between image quality, frame rate, system complexity, and power usage.

My Role & Responsibilities

As the camera systems lead, I am responsible for:

- Designing the camera architecture for the balloon payload
- Selecting lightweight, high-resolution USB cameras for the final system
- Developing Python scripts to reliably record images and video on the Raspberry Pi
- Creating a Flask-based web interface for testing, configuration, and data retrieval
- Prototyping and validating camera functionality using a development setup

- Planning transitions from a test-only configuration to an autonomous, flight-ready system

Process & Development

I selected the Raspberry Pi for its low weight balance, processing capability, and USB connectivity. Initial prototyping uses a standard USB webcam to validate software functionality before transitioning to dual high-resolution Arducam cameras in the final payload.

For early testing, I implemented a Flask web server that allows control and monitoring of the camera system from a device on the same network. This interface is used for debugging and configuration; in the final system, the software will automatically begin recording on power-up without user interaction.

Design decisions emphasize simplicity and reliability. Rather than recording high-frame-rate video, I plan to limit frame rate and capture incremental images to reduce power draw, storage usage, and the risk of software instability.

Results & Impact

The current prototype successfully records imagery using a Raspberry Pi and USB camera and supports remote configuration through a web interface. While the system is still in early testing and requires further optimization, it establishes a functional foundation for the final flight hardware.

Reflection & Next Steps

This project emphasized the importance of planning, reliability, and systems-level thinking when designing for environments where failure is not recoverable. Working under weight, power, and environmental constraints has reinforced how early design decisions shape system success.

Next steps include implementing frame-rate limits, conducting extended runtime tests, transitioning to the final camera hardware, and preparing the system for autonomous operation during flight. Balancing technical ambition with realistic timelines has been a key learning outcome as the project progresses toward launch.

Smart Vision Saw Safety System (In Progress)

Project for AZSEF/ISEF

Project Overview

This project is an **in-progress engineering research project** developed for submission to the **Arizona Science & Engineering Fair (AZSEF)** and the **International Science and Engineering Fair (ISEF)**. The goal is to design a low-cost, camera-based safety system that can proactively detect hands near a saw blade and rapidly cut power before contact occurs.

Unlike existing solutions that are typically designed for a single tool type, this system is intended to be adaptable across a variety of common workshop tools, including **table saws, miter saws, and chop saws**. By combining computer vision with a user-configurable interface, the project explores how a single safety device can be flexibly deployed in different workshop environments using affordable, off-the-shelf hardware.

Problem & Design Challenge

The primary challenge was designing a **fast, reliable, and configurable safety system** suitable for multiple types of saws. In safety-critical applications, even small delays can result in serious injury, making system latency a strict constraint. The system must detect both the blade and a hand entering a defined danger zone and trigger a power cutoff within a very short time window.

A second challenge was usability. Workshop tools vary significantly in geometry and orientation, making fixed, hard-coded detection zones impractical. The system needed to allow users to define blade location and detection boundaries while still relying on intelligent computer vision rather than manual thresholds. Additionally, the system must balance safety and practicality: minimizing false negatives is essential for injury prevention, while excessive false positives can disrupt workflow and reduce trust in the system.

Process & Development

The system is built around a camera and single-board computer mounted near the cutting area. During setup, the user interacts with the touchscreen display to visually define the blade region by drawing a bounding box. This configuration allows the system to adapt to different saw geometries without mechanical modification.

Once configured, the vision pipeline continuously analyzes camera frames to detect the blade and any hands approaching it. Using spatial relationships between detected objects, the system determines when a hand enters the defined danger zone. If this threshold is crossed, the system immediately triggers a relay to cut power to the saw.

To meet AZSEF and ISEF evaluation standards, I developed a structured testing process to measure detection accuracy and end-to-end system latency using timestamped logs and video analysis. These measurements guide iterative tuning of model parameters and system thresholds.

Next Steps

The next phase of this project is focused on system assembly and experimental testing. I will construct a safe mock table saw blade using a low-torque motor and visual blade representation, allowing the full system to be evaluated without introducing cutting risk. The camera-based detection system will be integrated with a YOLO-style hand detection model running on an Orange Pi 5 Pro, with careful optimization to maintain real-time performance on embedded hardware.

Once assembled, I will conduct iterative testing to measure detection accuracy, false positive rates, and end-to-end system latency from hand detection to stop signal. These results will guide refinements to model selection, danger-zone thresholds, and processing pipelines. Through this process, I aim to further develop my understanding of safety-critical engineering, where rigorous testing and responsible decision-making are essential.

Additional Engineering Experience

Wentworth Institute of Technology — Impact Lab (Robotics)

July 2024

I completed a two-week, on-campus pre-college program at Wentworth Institute of Technology focused on robotics and systems engineering. Through hands-on coursework, I studied kinematics, haptics, sensors, and ethical considerations in robotics design. The program emphasized modeling, simulation, and real-world applications of robotic systems, strengthening my understanding of how mechanical motion, control, and human interaction intersect in engineered systems.

Harvard University (edX) — CS50: Introduction to Computer Science

January 2024

I completed CS50, an intensive introduction to computer science that emphasizes problem solving, algorithmic thinking, and software design. Through weekly problem sets and a final project, I gained experience in Python, C, data structures, memory management, and debugging. The course strengthened my ability to design software systems from first principles and directly informed later projects, including my Minecraft Server Wizard, where I applied concepts such as abstraction, automation, and user-centered design.

FIRST Robotics Competition — Programming & Controls Leadership

2019 – Present

In addition to founding Team Epsilon, I have served in programming and controls leadership roles on multiple FIRST Robotics Competition teams:

- **FRC Team Phoenix Robotics (Team 6833):** Lead Programmer and Treasurer
- **FRC Team Dragon Robotics (Team 2375):** Controls Subteam Lead

Across four years and three teams, I programmed autonomous and teleoperated robot systems using WPILib Java, implemented vision-based alignment systems, and mentored new team members in programming fundamentals and electrical integration. These experiences exposed me to different team structures and engineering workflows, helping me adapt quickly and effectively in collaborative, high-pressure environments.

Phoenix Forge — Engineering Internship & Technical Training

September 2025 – Present

I am currently an engineering intern at Phoenix Forge, the largest makerspace in the Southwest. Through structured training and hands-on projects, I have developed fabrication and prototyping skills including welding, woodworking, CNC machining (metal and wood), laser cutting, and 3D printing. I have also supported makerspace members with equipment use, safety procedures, and fabrication workflows. This experience has strengthened my ability to move from digital design to physical implementation and reinforced the importance of precision, safety, and craftsmanship in engineering.

Phoenix Union Bioscience High School — Engineering-Focused Curriculum

2022 – Present

At Phoenix Union Bioscience High School, I have completed an engineering-focused curriculum that includes Honors Engineering & Design I–II and Honors Engineering & Physics I–IV. These courses emphasized project-based learning, iterative design, and the application of physics principles to real-world engineering problems. Through this curriculum, I developed a solid foundation in mechanics, electrical systems, and analytical thinking, which directly supported my work in robotics, simulation, and embedded systems projects.