Nicholas Harty
Maker Portfolio

MAKER
MAKER
MAKER
MAKER
MAKER
MAKER
MAKER

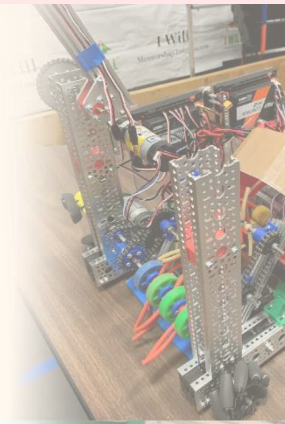
Inspired by the “The Worm” cover, which was inspired by NASA’s “Graphics Standards Manual”:
A masterclass on bringing ~~uniformity~~ to design, unseen before in government

But for me, ~~uniformity~~ stops at the cover.

☆ Projects shown in **Videos** or **Maker Essays**

Maker Directory

Robotics

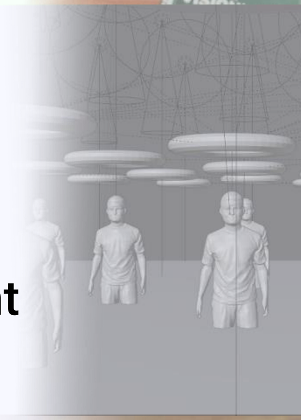


- π FIRST Tech Challenge - Java The Hutts #14725 ☆
  - “The Rookie / The Grabber” • “The Tall One” • “The Launcher”

Software/Product Development ☆

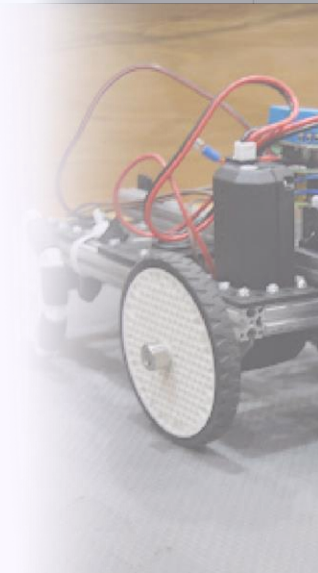
- π Pathways: A White-Cane Navigation Attachment
- π Medibound: An IoMT Management App
- π VisionBound: A Medibound Device for Classifying DR

A Lens into My Mind: Film



- π “Again”: A VFX film on Anxiety ☆
- π Climate Change Misinformation Satire News Segment

Fun Hands-On Projects

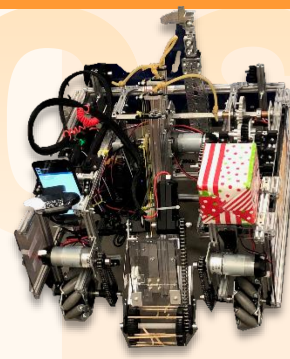


- π The Most Niche BINGO Software
- π Other Important Projects ☆

Worked with Cayden Keene, Aum Dhruv, Max Bazan, and Ishaan Ahmad. **Role:** Lead Programmer & Builder

Robotics – FTC Java The Hutts #14725

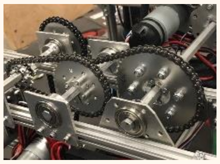
Year One: *“The Rookie / The Grabber”*



Year: 2018—2019 | **Status:** Complete

My Challenges / My Solutions

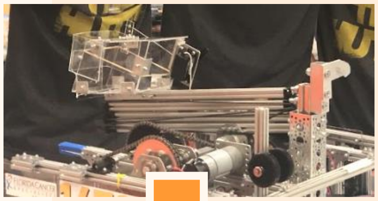
Building a stable Linear Actuator

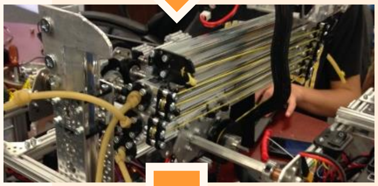


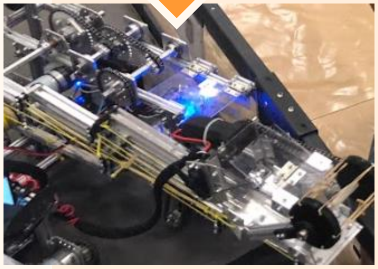
Using direct drive to rotate the collection arm was very unstable (to put it lightly) as the arm would fling from the forward to backward position… nearly breaking the arm on every rotation. The linear actuator itself was held back by surgical tubing (which would wear over time) and stretched forward by a yellow string on a makeshift wooden spool.

In the short term, to solve the problem with lifting the arm, being naive to the existence of motor gearboxes, I created a larger geared-down system to increase the precision of arm control on a much larger HEX shafting (as you can see, shifting from REV parts to GoBilda parts, in some areas, paid off in the long run).

Fast forward a couple of months, after watching videos from previous years and meeting with a mentor from BlueSparq, a local engineering firm, Cayden and I altered our design by removing the surgical tubing completely (which had been held inside the GoBilda rods) and jumping into Fusion360 (for the first time) to CAD design a new dual-string spool (an addition inverse string for providing pulling tension) for printing on our high school’s 3D printer. This new spool, after a few iterations, lasted us the rest of the season as its wear was minimal (except for the time it snapped after a faulty autonomous period) and it rarely tangled.







Steady motion in Autonomous

Originally relied on timed movement for the first few competitions as we worked out the quirks of arm motion

Eventually, after talking to other programmers in our league, I explored encoder motion, relying upon the built-in GoBilda encoders for navigation.

Encoder motion revealed its inaccuracies during precision turning. As a result, I explored gyroscopic options after scavenging online threads on the subject. Eventually settling on using the built-in IMU and a while loop got us through the rest of the season.

During the offseason, I conducted research on odometry-based robotics which would eventually be the basis for my Odometry-IMU-Vuforia correction navigation system of the third year.

```
public void turn(double powercus,int sleeptime ) {
    front_left.setPower(powercus);
    front_right.setPower(powercus);
    back_left.setPower(powercus);
    back_right.setPower(powercus);
    sleep(sleeptime);
    front_left.setPower(0);
    front_right.setPower(0);
    back_left.setPower(0);
    back_right.setPower(0);
}
```

```
public void turnDeg (  double ang   double speed) {

    // keep looping while we are still active, and not on heading.
    while (opModeIsActive() && !onHeading(speed, angle, P_TURN_COEFF)) {
        // Update telemetry & Allow time for other processes to run.
        telemetry.update();
    }
    lastAngle = getTrueAngle(angle);
    telemetry.addData("Last Angle:", lastAngle);
}
```

1st Year GitHub: https://github.com/jthftc/ftc-teamcode/tree/2018-2019

Summary & Personal Takeaways

My first year as a founding member of Java The Hutts is a time I will treasure forever. After a attending a “Mathletes” camp at FGCU, Cayden, Ishaan, and I set forth to found a robotics team and truly challenge our perception of engineering. In the beginning we walked blindly… first establishing meeting dates, then roaming YouTube and REV for any reasonably priced starter kits, and finally building our first working chassis. The progress in the beginning was slow and could’ve easily fallen apart given the lack of help or significant example within our area. But, powering through and challenging the process made the community we cultivated so much more worth it. Personally, I consider this robotics journey as my formal “Homecoming” into the function of an engineer. My first year taught me how to fundraise, presenting in front of rooms full of local engineer and educators. My first year taught me the importance of trusting initiative and the worth in believing in a long-term goal. My first year taught me the basics of Java as I would later dub the team “Java The Hutts”, combining both my foundational love for programming and sci-fi. However, above all, my first year taught me how to work long hours with others and converge multiple ideas with an open mind.

Worked with Cayden Keene, Aum Dhruv, Max Bazan, Ishaan Ahmad, and Will Givens. **Role:** Lead Programmer & Builder

Robotics – FTC Java The Hutts #14725

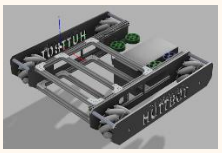
Year Two: *“The Tall One”*

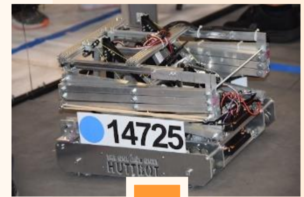
Year: 2019—2020 | **Status:** Complete

My Challenges

My Solutions

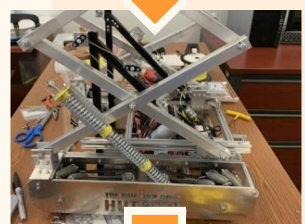
Stacking skystones: “Climbing the mountain” with scissor lifts





The scissor lift was no doubt the most complex and thought-out physical portion of the bot. Our decision to use a scissor lift over any other mechanism had 2 main reasons behind it. One was due to the supposed speed advantage over other designs, the other was its relative simplicity, even though it didn’t turn out to be simple in the end. My first version of the scissor lift used ¼ inch metal tubes as the main chassis of the lift. This was my first mistake, as it was far too heavy for our motors to handle.

Our first course of action was to order and mount some gas springs to try to alleviate some of the pressure being put on the motors. This was nearly perfect, as the dampening on the gas springs would’ve made the lift very smooth, yet quick going up and down. Unfortunately, we found out later that these springs did not meet regulation. Considering this, Cayden and I looked for further solutions.



We decided to order very strong metal springs and design 3D print connectors for them. However, as we found out, this was not optimal because of my poor connections between scissor lift segments. We simply used ¼ inch screws, causing tremendous amounts of friction, which contributed to the overall problem. This created a sluggish design that we would need to be address later in the season.

The major changes I eventually made were the material, which I changed to a thin C-channel with bearing cutouts and using bearings instead of screws as the connections. Our design varied greatly, changing from a string and pulley to a screw drive to lift the mechanism up. These changes enabled the lift to rise in under 2 seconds and improved our stability. The swaying stopped when we implemented 3D printed “X” blocks. Moving forward, I specifically worked on the top grabber that was comprised of a plexiglass base with a three-arm extender that allowed us to lower the grabber while remaining parallel to the ground/block.

Learning from new resources

With the new resources from our partnership with IWMF, I set out to truly incorporate as many home-made elements as possible (standing out in bright yellow filament) into our robot’s design. For 3D printing, these included custom pulleys, “X” blocks for the scissor lift, and rounded hanging-bars for our three-arm grabber. Spending so much time working on internal manufacturing made Fusion 360 a close rival of Android Studio for most of my second season.

In addition to 3D printing, one of the first elements I worked on in Fusion 360 were the side panels. After contemplating the final design for days (ensuring I didn’t expensively screw up the aluminum cut out), I finally sent the design over to be laser cut (provided by our sponsor “Bob Dean Supply”) and, once completed, permanently dubbed our year two robot “The One and Only Huttbot”

2nd Year GitHub: https://github.com/jthftc/ftc-teamcode/tree/2019-2020

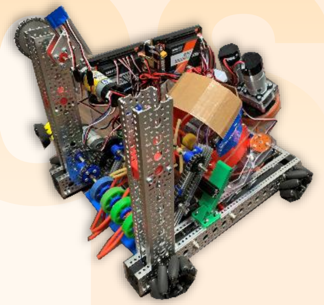
Summary & Personal Takeaways

Concluding my rookie season with Java The Hutts, I was exposed to the variety of tools, designs, and possibilities that consumed my aspirations for the follow season. After partnering with the I-Will Mentorship Foundation, a local non-profit committed to STEM education, we gained access to their inventory of machinery that would later culminate in the makerspace we constructed that summer. This gave us opportunities to experiment with in-house CNC machines (aluminum/wooden), a lineup of MakerBot 3D printers, and a larger space to host team events. These opportunities opened my eyes to wonders of industrial engineering and took me away from the computer, working on fun cutouts to my heart’s delight. Although I remained “Lead Programmer” for entirety of these season (and contributed to that role with equal enthusiasm ultimately culminating in my 3rd year programming advancements), the nostalgia I feel looking back on this second learning year is mostly surrounded in my introduction to CAD and the iteration-on-iteration of parts that became crucial to our SKYSTONE robot’s final design.

Worked with Cayden Keene and Aum Dhruv. Role: Lead Programmer, Builder & Mentor

Robotics – FTC Java The Hutts

Year Three: *"The Launcher"*



Year: 2020—2021 | Status: Complete

My Challenges

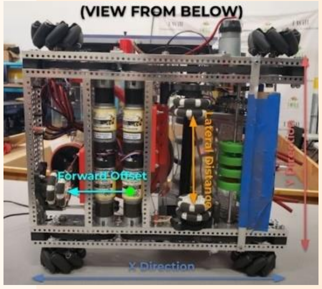
My Solutions

Improving localization during the Autonomous Period (using three-wheel odometry)

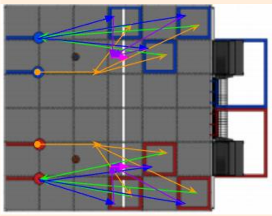
Custom designed and printed odometry modules

Spending the previous summer learning and implementing a modified pure pursuit algorithm (an arc formation/following algorithm) the year previous, an option discovered through research building a motion simulator/viewer in Electron.JS, I was disappointed to discover it's inaccuracy as it relied upon a constant point mass and reliable velocity measures. Neither of these would be viable on the new robot due to the rings' necessary concentration on one side of the robot (as opposed to the empty alternative) and the inconsistency of velocity readings coming from odometry as we made arc movements.

Moving quickly to provide any benefit to localization after postponing our pure pursuit build, we turned to a mature library titled Roadrunner for FTC. After making modifications to support three-wheel odometry systems and appropriate PID curves that could provide necessary arc movements that maintained fairly accurate (~10cm of error at the end of the period) for the entire autonomous period. These measures would ultimately become less important over time as the following ring-shooting alignment algorithm provided proper mid-autonomous/tele-op correction.



Autonomous plan



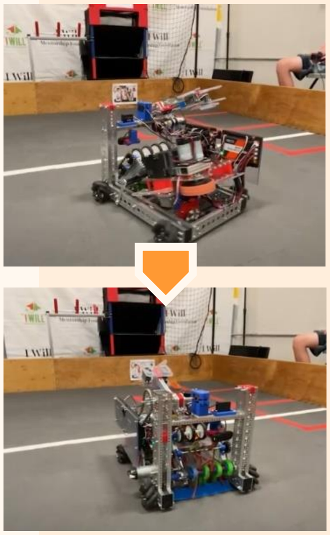
Shooting the rings with proficient accuracy during both the autonomous and tele-op (playing) periods

Originally, we relied upon the value of our IMU gyroscope for angle positioning to align with the top goal. However, this process was tedious and required that we begin the automatic motion of turning to a particular angle at the specific position in front of the goal.

We were met with similar failure when we implemented constant odometry position during tele-op. This system's coordinate positioning would often erode over time with the driver having to slam against the far wall of the field to re-orientate the coordinate/gyro variables.

Finally, after a late-night longshot program, I was able to achieve near-perfect accuracy by relying on a constant: vision. By maintaining a background Vuforia program that would provide angled dimensions from the Robot's position to an image played underneath the goalpost, I was able to use a responsive angle that would provide the basis for a delta turn that could adjust for any position (and any distance given the relative size of the image) on the field… very basic Pythagorean math that consumed our whiteboard and made my fellow team members think I was crazy for attending an issue seemingly "fixed" through odometry integration. But after tweaking the program for hours, Aum (our driver) and I were in awe at the simplicity of tapping a button and automatically shooting three rings perfectly into their netted capture across the field. Truly one of my most magical moments in robotics.

Turning to match the angle of the high goal:



3rd Year GitHub: https://github.com/jthftc/ftc-teamcode/tree/2020-2021

Summary & Takeaways

Going into my third year, I wanted to challenge myself by going back to square one of robotics: Navigation. Up until this point, our autonomous program relied upon rigid/perpendicular movements and lacked vision as a key to motion correction. After working to design odometry modules within the previous season, I set my sights on spline movements that would target specific coordinates on the field as opposed to the unreliant delta movements of the past. In addition, the new season lacked significant y-maneuvers (bumps, hills, etc.), and fixed goal targets almost required a refinement of autonomous actions during the playing period in order to have any consistency in shooting rings into the highest goal. A large problem that plagued the development period of our third season was our limited active pandemic members. Being a team of three (all founding members), we had to be concise in our plans for the season and focused on our set responsibilities. These goals and challenges set the stage for my last season and the end of a trilogy that, looking back, continues to evoke my innate passion for problem-solving and crawling around on foam tiles trying to view the world through the robot's perspective.

Worked with Aum Dhruv and Cayden Keene **Role:** Programmer, CAD/Photoshop Designer, Speaker

Software & Product Development

Pathways: A White Cane Navigation Attachment

Year: 2020-2021 | **Status:** Partial Prototype; Inactive Project after I left Florida

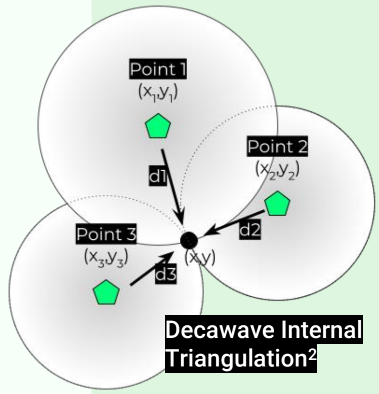
Why?

Although there have been many innovations in technical assistance when it comes to disability, many modern exercise utilities, either physical or mobile, don't provide proper technology to suit people with visual disabilities. With the introduction of mobile positioning and UWB (Ultra-Wideband) technology[1] in the early 21st century, many existing and emerging fitness brands invested in technology that would track the well-being of users as well as display performance data. However, one target market mostly untouched by this overflow of innovation has been specifically visually impaired individuals who often experience the majority of issues when it comes to maintaining their health. This became ever clear upon meeting with the SWFL Council of the Blind and refining our product through first-hand accounts that put the experience of blind individuals into perspective. Through our innovation, an adaptor to the original white cane design (titled the "GEOCane"), we believed we could alter this status quo and truly change the confidence of all partially-sighted individuals in their ease of navigation. All of the basic underlying features of this product contribute to our impactful goal of increasing the confidence of our users by adopting methods that protect their safety.

MDEK1001 Beacon[1]

How? & Goals

The design of our product/solution works on existing technology called UWB (Ultra-Wideband). UWB is a radio wave-based technology that's purpose is to transfer data and determine location through radio waves. In our case we are using UWB technology in combination with triangulation, similar to the triangulation used in GPS satellites, to determine location. With this in mind, we present to your pathways. Pathways is a system using UWB radio beacons and receivers to track the exact positioning of a visually impaired user. Beacons consist of the MDEK1001 circuit board (UWB beacon/receiver) connected to a constant power source like crosswalks, streetlights, and traffic lights. The beacons have a radius range of 150 Meters allowing the beacons to be placed at least 300 meters apart.



Decawave Internal Triangulation[2]

$$(x_1 - x)^2 + (y_1 - y)^2 = \boldsymbol{d_1^2}$$

$$(x_2 - x)^2 + (y_2 - y)^2 = \boldsymbol{d_2^2}$$

$$(x_3 - x)^2 + (y_3 - y)^2 = \boldsymbol{d_3^2}$$

$$\boxed{\{y\}} = \frac{(x_2-x_3)\,[\,(x_2^2-x_1^2) + (y_2^2-y_1^2) + (d_1^2-d_2^2)\,] - (x_1-x_2)\,[\,(x_3^2-x_2^2) + (y_3^2-y_2^2) + (d_2^2-d_3^2)\,]}{2[\,(y_1-y_2)\,(x_2-x_3) - (y_2-y_3)(x_1-x_2)\,]}$$

$$\boxed{\{x\}} = \frac{(y_2-y_3)\,[\,(y_2^2-y_1^2) + (x_2^2-x_1^2) + (d_1^2-d_2^2)\,] - (y_1-y_2)\,[\,(y_3^2-y_2^2) + (x_3^2-x_2^2) + (d_2^2-d_3^2)\,]}{2[\,(x_1-x_2)\,(y_2-y_3) - (x_2-x_3)(y_1-y_2)\,]}$$

The users were tracked with our device called the Geocane. For our triangulation system[2] to work properly you have to be within the range of 3 beacons. By taking the distance from each beacon to the receiver we are able to calculate the exact positioning of the user by triangulation. The receiver (MDEK1001) for the Geocane is placed inside of a lightweight housing that simply clamps onto a white cane. A white cane is a specialty type of cane design for the visually impaired to feel the environment around them. After meeting with a visually impaired person named Will Alexander who is a part of the SWFL Council For The Blind, we got great feedback on the design and functionality of our cane. The cane needed to be lightweight so it doesn't strain the user's wrist. This is essential because the user relies on tapping the cane back and forth (a wrist movement) to feel the environment around them. So, when designing the cane we heavily took into consideration the weight of our module to ensure the user that our device will have the physical feedback of a normal white cane. Within our Geocane module, we have integrated features like haptic and audio feedback to guide the user. The haptic feedback is provided to the user through a vibration motor as well as the audio feedback being provided to the user through a Bluetooth module.

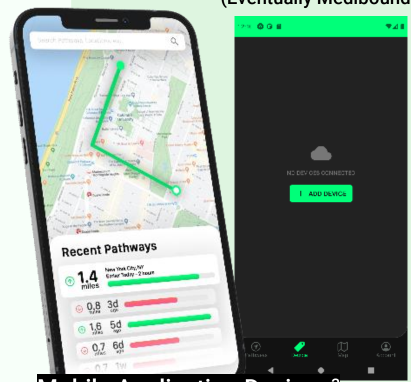
We planned to allow the user to connect headphones (over BLE) to hear audio instructions. A pair of bone-conducting headphones is recommended when using the Geocane due to its ability for the user to hear the audio instructions as well as the environment surrounding them. When the user is tapping the cane in the wrong direction the Geocane will begin to vibrate and the audio will instruct the user to travel either left or right. When the user is traveling in the correct direction the Geocane will stop vibration and the audio will instruct the user to continue traveling forward. We also incorporate an interface that scans for UWB beacons using the device and forms a virtual boundary. When nearing the boundary, the

The Beacon-Module System in Action

device produces a variety of vibrations to alert the user of their distance from the side of their pathway. This is to ensure the safety of users through a padding system within individual pathways. In this sense, the user experience is meant to be intuitive in that our product is very easy to use as a stand-alone solution and its longevity is maintained through the use of additional systems to monitor the progress of the user. Such a system is meant to build confidence within the end-user as they move from simply learning to navigate with this device to use it for more advanced forms of exercise. On top of this, our consumer app (early build of Medibound)[3] allowed for family members and other loved ones to monitor and create routes for the visually impaired individual. This will allow for the sharing of pathways in a unique social aspect.

Photoshopped Design

React Native Partially Functional Build (Eventually Medibound)



Mobile Application Designs[3]
https://github.com/hartytech/pathways-app

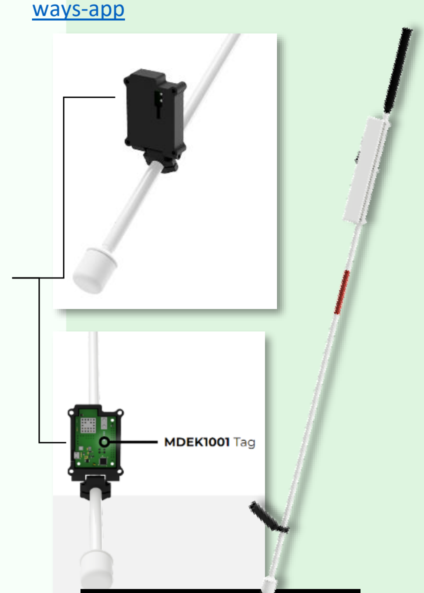
My Challenges / My Solutions

Figuring out localization

RFID: Passive RFID systems, within a considerable production cost often provide limited detection range (<1m in radius). Active RFID modules use up a lot of energy upon constant transmission.

BLE: Only provides double the accuracy of Wi-Fi based solutions with the localization of individuals being within <8 meters of their actual location. Dependent upon Bluetooth and may vary is success based upon the density of other mobile devices.

UWB: UWB has a very high accuracy to less than 30 cm when triangulating a location and within in range of 3 beacons/detectors. A beacon/detector has a max range of 150m. With a longer range than BLE , UWB beacons can be spaced further apart limiting the number of beacons needed triangulate location within in a large area.



Post-Meeting Split Design (Beacon Bottom – Control Top)[4]

Response to feedback and the countermeasures of the lack of initial sample research

When we first were inspired to explore improvements to white cane navigation, we let our ideas run away with us. Before we knew it, we had a solution that was too heavy on the hand, too complex to be taught, and too expensive to implement… a disappointing truth that only emerged upon first meeting with the local Council of the Blind.

This reality forced improvements such as audio-hapic voice assistance, through experimenting with Google Directions API, and modifications to the split layout[4] of the final design as a means to balance the weight of the device[5]

Disconnecting Decaware's software… connecting the device to our application

Other challenges existed primarily in the software, an area where I held the most responsibility. By using Decawave's (the manufacturer of MDEK1001) hardcoded triangulation software, I was able to modify the firmware in SEGGER to enable BLE mode with, in turn, detached the board from its rudimentary connection to Decaware's mobile app and allowed me to receive serial data on the raspberry pi. Using that data, I would export the terminal stream to a NodeJS program that utilized the library bleno to reproject that data to the React Native mobile app. Although this solution did present some latency, it wasn't enough to provide significant resistance to border detection and it served as a good opportunity to develop BLE technology.

Outer Encasing Of Final Design[5]

Although Incomplete… Takeaways & What's Next

- π Although the product wasn't entire commercially viable, it proved to be a powerful instrument to connect us with a pocket within our community and realize the amount of primary research necessary for successful product development.
- π IoT is **REALLY** cool. The complex data relay system described in the "Challenges" section took a while to construct, but, after it was finished, I was blown away by both the accuracy of the localization technology and the ability to manipulate technology over BLE through multiple consoles/centers of code… an experience unbeknown to me in the world of FIRST Robotics

Worked with Aum Dhruv **Role:** Programmer, UX/Photoshop/Illustrator Designer, Speaker

Software & Product Development

Medibound: An IoMT Management App

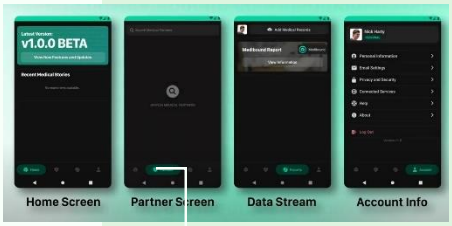
Current Branding Standard:

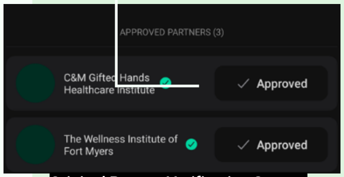
SVG, I made in Illustrator | Gradient Version | App Icon

medibound
Roboto Medium 12px Lowercase
Font/Text Styling Standard

Year: 2021-Now | **Status:** Multiple Prototypes; Still creating iterations with further research
Website: https://medibound.com/

v1



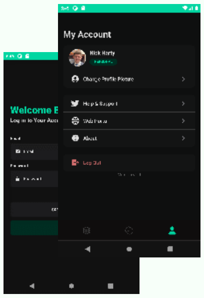


Original Partner Verification System for Sharing Data[1] (currently in the process of being replaced by FHIR)

Why?

After concluding the Pathways project, the mobile application I developed in the process of that research clearly had broader implications beyond its isolated use case within blind navigation. Renaming the application component of the project Medibound, my partner and I redefined our mission by implying we wanted to connect the average user to the larger science of predictive medical health (often found in IoMT NN devices)

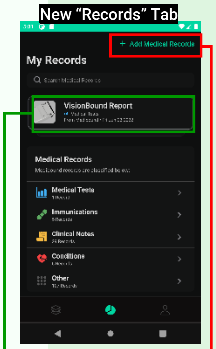
Version 1: Continuing to work on the application, I initially focused on increasing the security of medical users by creating a two-way gate between health providers and patients[1] within the app (Using Firebase Auth for login).



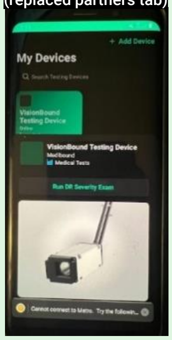
Version 2: However, eventually, we dropped the security component, opting to connect with existing security-based medical data-sharing platforms such as Apple/Google Health, and increased our focus on creating a mobile dashboard for managing IoT devices and introducing simplicity to data presentation. Throughout the entire process of designing this application, a Photoshop canvas became a dear friend in solidifying design ideas and quenching an internal desire for affordance perfection. In doing so, we saw this idea as an opportunity to learn more about constructing mobile utilities with React Native and connecting such big data to Firebase, and later the broader IoT Google Cloud platform.
(Login/Account page roughly stays the same)

v2

New "Records" Tab

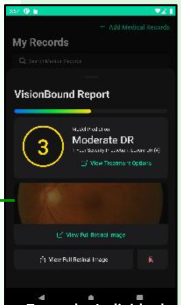


New "Devices" BLE Tab (replaced partners tab)



How? & Goals

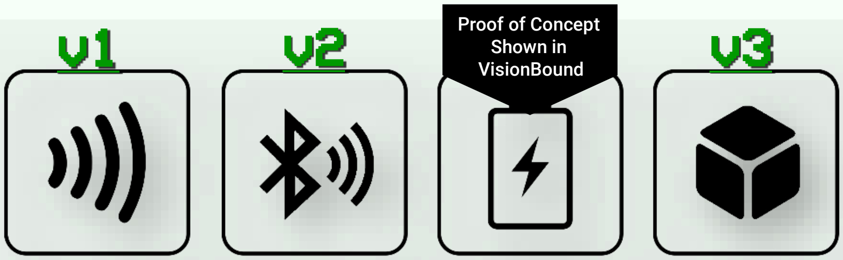
Medibound was developed in React Native and served as the primary introduction, within my education, to NodeJS. I saw NodeJS as a marvel when compared to traditional OOP programming languages. Many of the edits I made to the program would update in real-time without recompiling, an experience that dated Java significantly within my mind's eye. The major components in the development of Medibound v1 came with learning Firebase as a data management platform as well as the standards for medical data communication across the web. However, in Medibound v2 (A result of feedback received from Congressman Bryon Donalds and the Congressional App Challenge), more focused on IoMT management, we focused on connecting devices over a standard of BLE (using a raspberry pi test kit) and incorporating a variety of different methods for extracting data from these devices. These methods included BLE data transfer (for connected devices), Internet/Firebase (dubbed DeviceAPI) transfer, and (my personal favorite) contactless, anonymous NFC transfer[2]. There was something magical about connecting these devices to one central system and watching a system that started as a Photoshop sketch transform into something greater.



Example: Individual Record

Using NFC to add a Record[2] (more on process in VisionBound)

Characteristics of a "Medibound" Device (As it stands now):



NFC Contactless / Anonymous Data Transmition (Still working out legality)	BLE Functionality to Connect Device To Medibound Mobile App for Management	Mobility: Battery Powered, Energy Efficient, & Uses Interpretive Neural Networks	Part of the Medibound Artitecture (Device API / Encorporate FHIR)

My Challenges / My Solutions

The learning curve of moving from a robotics Java environment (or RaspPi) to a NodeJS environment.

When first culminating our initial ideas for our application, our team discovered multiple errors within our learning that would need to be overcome if a final product were ever to flourish. Some of these initial obstacles included workflow management and mobile development. Neither of us had experience in proper career programming nor formal cross-platform mobile development and I had come from a background in web-based development. In addition, I had recently moved away and we initially struggled with maintaining the project.

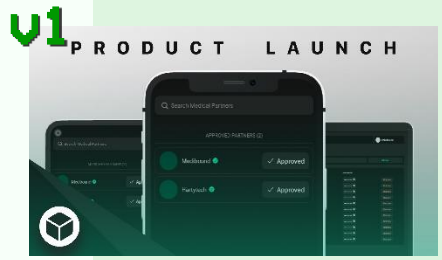
However, we were able to overcome these two challenges through the use of project planning/communication software, such as Slack™, and lengthy programming sessions dedicated to learning React Native (our mobile interface framework). We kept a tight schedule as we entered the school year to make sure we didn't fall behind on this learning process. This initial planning certainly laid the groundwork for much of our future research.

Current GitHub Repo for the React Native App:
https://github.com/medibound/mbapp/

Learning Firebase and Server Communication

Further challenges came in the form of server communication and privatizing our users' medical information. After significant research and comparing similar server solutions, we finally concluded upon using Google's Firebase server network to serve all of our user information and real-time communication requirements. We ended up resolving upon Firebase due to its simplicity in development (due to its expansive API documentation) as well as various security utilities that were available across all of our necessary platforms. On top of this, Firebase's ease of use within our NodeJS/Javascript environment allowed for a lesser learning curve and an easier translation of data in the local-server system.

However, the advantages of Firebase would also, in some instances, serve as hindrances as we would find bottlenecks within such communication that would, in turn, slow the loading of local elements. To resolve such issues, our team resorted to dividing our IoMT data between Firestore (Firebase's long-form data storage) and Firebase Real-Time as well as transferring less/minimized data between our application and Google's back-end. All in all, these challenges aided our overall learning and helped us become better problem-solvers.



Explanation of 1st Version of Application:
https://www.youtube.com/watch?v=8DehAuyERTQ

Explanation of 2nd Version of Application w/ VisionBound:
https://www.youtube.com/watch?v=pHHouSRkQ7I

Takeaways & What's Next

Going forward, I want to develop a proper API platform (as opposed to our makeshift NextJS one) to allow developers creative freedom in curating different layouts of medical data and ultimately adopt the proper security features to commercially comply with government regulation (this product, although complete in its rudimentary functionality, is far from commercial launch). Learning and incorporating FHIR through their set of standards will surely help with this as we continue to shift into a more industrial-ready product.

- I learned a lot more about the limitations and audience of mobile health solutions as well as the current standards and trends within the emerging mobile IoMT device industry (the subject of a later study of such uniform trends)
- I grew ever-ambitious in changing the standard of IoMT development to incorporate the data transfer methods we had found to be great benefits to the consumer. In this sense, we grew eager to learn more about predictive health algorithms and develop my own "Medibound" IoMT device…

Worked with Aum Dhruv **Role:** Programmer, CAD/Photoshop Designer, Speaker

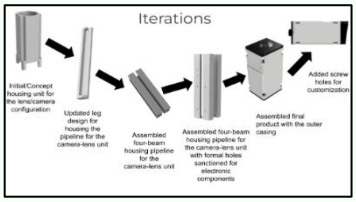
Software & Product Development

VisionBound: A Medibound Device for Classifying DR

Year: 2021-Now | **Status:** Multiple Prototypes; Still creating iterations with further research
Website: https://medibound.com/

Why?

Initially, my uncle's vision complications due to diabetes drew me into a Kaggle post on "Diabetic Retinopathy" neural network detection. Although I knew little about neural networks, I was eager to learn and apply such knowledge to my perspective on MIoT devices. Little did I know, that this passion project would transform into an entrepreneurial journey that would sweep my Junior year. After conducting further research on Diabetic Retinopathy, I learned that 79% of the adults with diabetes reside within low-to-middle income nations and serve a major risk of joining the estimated 93 million people worldwide suffering from diabetic retinopathy. This information changed my perspective on the disease and provided a convergence between my interest in neural networks and the MIoT mobility initiatives of the Medibound Project. Dubbing this project, consequently, Visionbound, My partner and I embarked on a journey to compare existing neural network solutions (CNN/KNN) with respect to DR and develop a standard around DR testing for what a "Medibound" device should look like[1].

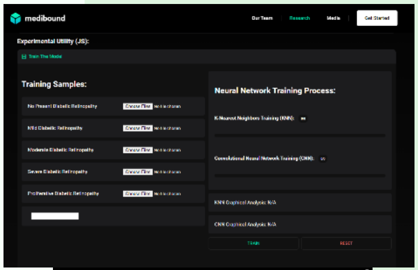


Final VisionBound Render[1]

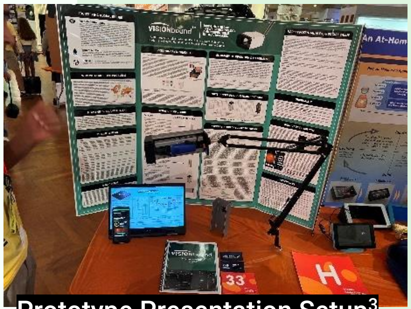
How? & Goals

The Learning Comparison Study (CNN v. KNN):

To study such differences, the investigators developed these two closed algorithms through TensorFlow API (ml5.js port to the webJS environment) to serve as an accessible web utility. After preliminary setup, we began training the algorithm with samples at increments and testing their accuracy via their developed online/public platform. This platform was built in HTML/JS that was pasted into an existing WordPress site (https://medibound.com/project)[2]. In essence, its purpose was to take training and testing images and compare the sample size of each training set to its eventual accuracy (properly guessing the severity of a DR sample 1-5) The experimentation produced accuracies categorized by DR severity and NN algorithm. In both algorithms, there was a general increase in accuracy as the training sample size increased. The CNN algorithm produced greater accuracy across higher training sample sizes ($n \geq 375$) compared to KNN, which produced greater accuracy across lower training sample sizes ($n \leq 250$), supporting our hypothesis. For each training sample size, a two-way ANOVA generated p-values less than the significance level of 0.05. The null hypothesis—no statistical significance between the algorithm type and its average accuracy across training sample sizes—was rejected. We concluded it was a statistically significant relationship, proving that the algorithm type played a role in its accuracy. These results coincide with the global ophthalmic community that seeks widespread testing on behalf of those in low-income areas who are most at risk for DR[3].



Testing/Training Platform[2]



Prototype Presentation Setup[3]
https://medibound.com/ICPoster.pdf

The Development of a Nonfunctional Prototype:

With the information from our previous research, we constructed a prototype that used Diabetic Retinopathy as a focus for testing and Medibound as a management platform. This product was more of a proof-of-concept as it showed how the rather out-there features we considered to be essential to "Medibound" devices could be practically applied to an actual medical utility. These features included NFC functionality[4] to serve a larger sample of testing patients and BLE device management for initiating testing. In addition, we also used Fusion360—executing on much of the knowledge we had attained through robotics—to develop an outer encasing for the product. Although this device never performed an actual test, its development and design are an accurate representation of how researchers can provide mobility while testing DR in the field[5]. In fact, many products produced around the time of our research indicate the very accuracy of our design as well as the need for a unifying standard of data simplification across MIoT devices.

NFC Incorporation w/ Adafruit ST25DV16K[4]

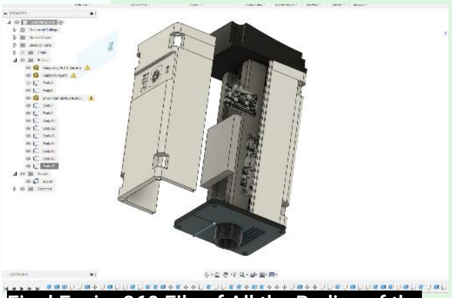
Similar Solution: Working prototype Developed by ITRI (Dec 2021 – 2 months After we finished our nonfunctional prototype)[5]

My Challenges | My Solutions

Distance prototyping and learning with a friend

The major challenges associated with this overarching Visionbound project were many and a lot of them were associated with the distance between my research partner and me (He lives in Florida; I moved to New York). For example, when working with the Kaggle DR library, we often had trouble working together over the computer due to the computational strain of each individual trial. Another instance of this can be found in product development where I didn't have access to a 3D printer within my vicinity.

As a result of this, my research partner, in collaboration with a makerspace at FGCU (his local college) had to print out 3D components and assembly them after I sent him over the CAD files I had designed in New York[6]. This process was repeated multiple times until the outer encasing perfectly fit together, in which case my partner would then send the components up to New York for me to assemble and solder together the necessary electronic components (Raspberry Pi Zero W, 12 MP HQ Camera, a 2465 Adafruit PowerBoost 1000C Rev B, an Adafruit ST25DV16K I2C RFID EEPROM Breakout, a lithium-ion polymer battery (3.7v 2500mAh), a USB-C cable, and an outer shell casing).



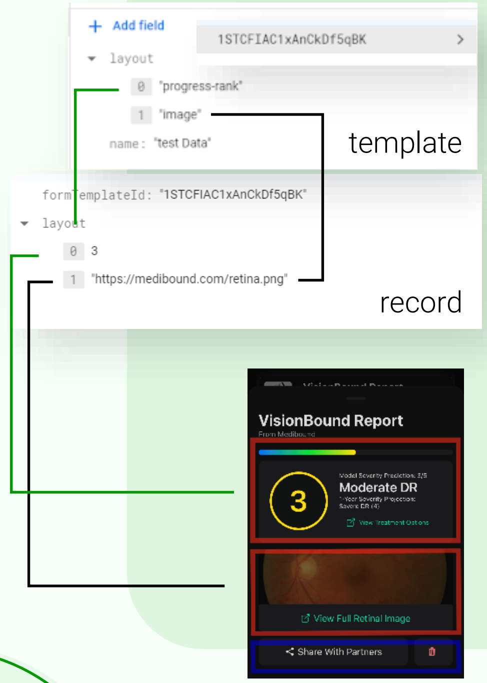
Final Fusion360 File of All the Bodies of the Prototype (w/ Scaled Placeholders for Components)[6]

https://medibound.com/VisionboundDevice.f3d

VisionBound – Medibound communication of data.

Much of the data transmission between Medibound and VisionBound is split into two forms of commands. Action commands (such as "Run a test") and Data commands (such as retrieve an array from NFC).These transmissions were very simple to prototype as we originally provided zero data protection. However, one difficult point about this data transfer was figuring out how to display it on a phone. To do this, I created a very rough form template type that would serve as the basis for how to display data on the mobile phone.

However, the main problem with this prototype lies in that there is currently still no way to create these templates, from a provider standpoint, without directly editing Firebase (which is not feasible) and there is no form of standardization/security of which I could communicate this data with provides. After a recent discussion with Eric Just, a health technology advisor, we have begun to incorporate a medical data standard called FHIR into our application and create a service, probably through Postman (although I am shopping other options like Retool), to create templates and manage data presentation from a developer/provider standpoint.



Takeaways & What's Next

- Although we learned a lot about the internal structure and adjustable metrics of the K-nearest neighbors' algorithm and Convolutional Neural Networks, I recognize that we only scratched the surface of a large ocean of research dating back to Alan Turing on the intricacies of machine learning systems. I will further explore such wonders in depth as I continuously, in the back of my mind, connect such utilities to their possible use cases in emerging health technology.
- We discovered a larger (newly emerging out of COVID-19) industry of IoMT devices that each seems obsolete in their individual evaluation. Specifically, we took notice of the movement to remove the duties of physicians within many low-income testing initiatives through the development of NN confidence. Such stemming interest, referring all the way back to our originally Pathways project, would inspire a market analysis with FGCU looking at the trends amongst these emerging technologies and the resulting standards that would best serve low-income patients.
- Learning doesn't always present itself in similar shades. In robotics, I grew used to evident problems with seemingly definite solutions. Through moving out of that groove, I have learned that end-all-be-all projects, such as Pathways, Medibound, or Visionbound in hindsight, can reveal themselves to be pieces in a longer puzzle of a more niche interest. In some ways, my research into MIoT correlates much with my more abstract film projects. They both reveal a magic that excites people and changes the perspective of many as to what is possible.

Worked with Matthew Harty (Brother) **Role:** Director, Video Editor, Writer, and (not the best) Actor

A Lens into My Mind: Film

"Again": An Analogy for Anxiety

Year: 2022 | **Status:** Still working on it (Might be complete by the new year, check **film.nickharty.com**)

Previous lightsaber films in After Effects:

2018: 13 years old

2015: 10 years old

Project Inspiration

In the summer after my Junior year, I was academically exhausted. Having gotten through my first year of IB and competing in numerous competitions for my work in MIoT development, I desperately wanted a break and saw a return to filmmaking (after a year of off-and-on editing projects) as a welcomed form of relaxation. Swept in the nostalgia of entering my senior year, I knew I wanted to make something that revived the lightsaber effects that got me into VFX while exploring new tactics/technologies that would surpass anything I had previously attempted. This culminated in a film that seems grim upon its title, but behind the cover lies a plethora of learned skills, new environments, and a world for which I was proud to curate. *In the end, I hope to show, in this film, how to mind can run itself in loops trying to deal with the worries of the world.*

Final Output & Development

Lightsaber Prop & Effects/Tracking:

In the very beginning of the film, I planned to have to main character (myself) fight a mysterious birdman figure in a lightsaber duel. Having done VFX work before in lightsaber movement, I wanted to include a few additional elements in the filming process that would make the actual saber editing much easier. In the end, I wanted to design a lightsaber that, akin to those of recent Disney Star War projects, had a clear 3-dimensional frame that felt as if it was a real plasma beam.

Challenges & Solutions

Tracking Setup:

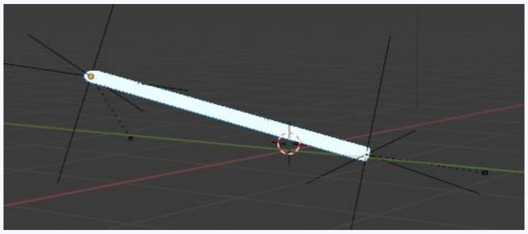
To improve upon previous attempts at lightsaber editing, I went out and bought two 1" dowel rods and two 1 ½" PVC pipes to act as the prop for this effect. Then I cut the following materials down to fit the parameters of my design:

- 13.5 inches for the lightsaber helm (including the end piece)
- 4 feet for the total length of the dowel rod (only of which 34.5 inches are visible)

Then, after spray painting the helm black, I screwed the dowl rods in at the helm and tapped blue and black intervals on each of the prop sabers.

Editing:

First, inspired by saber videos produced by Ian Hubert, I started editing my lightsabers in Blender. It took me a minute to learn how to track my prop blade in Blender, but after some practice with test footage, I was able to attach my 3D blade to the x-z coordinates of my footage (adjusting the y-coordinates / or scale, and the xyz-rotation of the blade for each individual frame).



However, one of the issues I had in Blender was the complexity of individual adjustment. It may have been the particular setup (originally intended for 2D sabers) I had created that was so tedious, but I got frustrated with editing the saber in 3D with no reference and having to go back to the Camera View to check if my saber was in the correct place. Because of these issues, I looked for solutions that would allow me to go back to After Effects (the software I originally edited my first

Sci-Fi/Real Life Inspirations:

The slight sparks and real defined 3D blade from the new Kenobi series.

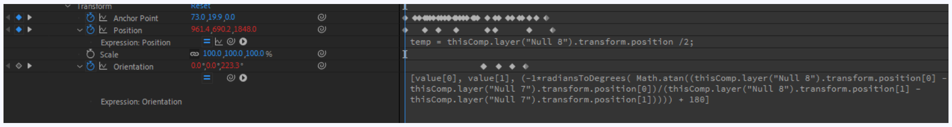
The real life Plasma blade developed by Hacksmith.

two saber films in) while still achieving believable (as far as light diffusion) 3D lightsabers

Resulting Blender Lightsaber Renders (too thick, not focused enough, and hard to adjust/edit)

After watching a video from ProductionCrate, I finally found my final solution. I figured out that, with the Element3D plugin, I could create and polish my saber in After Effects without losing the realism of Blender. At the end of the day, I enjoyed the real-time 2D viewpoint and the easy-to-use handles of AE Element 3D over the guess-and-check system I had been using in Blender.

To create my saber effect, I used a 3D saber model I created in Element3D coupled with tracking points from Mocha. To make the system of adjustment easier, I used the first of the following equations to connect the base position of the saber with the base position of the prop (tracked). The second equation, more importantly, connects the x-z (visual plane) angle of the saber using the inverse tan function between two tracked points (points at the beginning and end of the saber)



Tracking points in Mocha AE

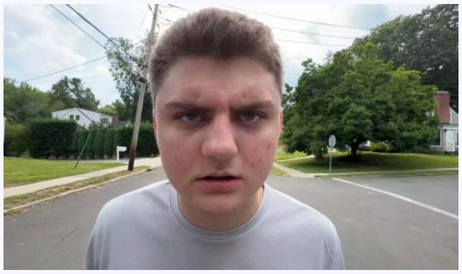
With this function completed, all that was left was to adjust the y (forward-backward) and z-rotation position of the saber every couple frames. This was made even easier by including a helm in my 3D design and adjusting the transparency to line it up with the actual helm.

The final portion of this effect, the glow, was simulated by creating multiple duplicate layers of the saber and, using a Fast Box Blur, adjusting the blur to be exponential across the different layers. This created a layering 3D effect and, once I added a keyframed jitter to this glow, made it feel like an experimental plasma blade.

Camera Effects - Vertigo Shot:

In the beginning and end of the final version of the project, I plan to have the main character in a vertigo shot to indicate their surprise at seeing the birdman (Villian) standing across from him.

This shot took a few attempts but ended up looking fairly solid given its unorthodox capture technique. Instead of using a traditional dolly zoom and long-angle film lens, I used an iPhone 13 Pro camera to zoom in and out of the character's face. Then, to simulate the background loosening, I recentered the character's face using digital scaling in Premiere Pro. Filming in 4K helped reduce any visual detail issues of the digital zoom, especially because I exported the project in 1080p.



Blender 3D Environment:

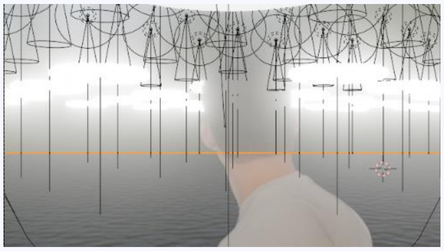
This 3D dimensional environment was created in Blender to serve for a scene where the main character is surrounded by clones of himself… seemingly lost in his own mind as he is being chased by the creature (himself) anxiety.

Learning Blender:
To learn about the features I discuss later in this C&S excerpt, I scavenged first through Ian Hubert videos ("Wild Tricks for Greenscreen in Blender" for example), then onto channels like 3DGreenHorn and askNK who taught me how to create specific effects like moving water or 3D person models.

The Setup: *A greenscreen roughing strung between my bedroom's TV, garbage can, and door.*

Editing:
Having abandoned Blender for saber development, I wanted to incorporate and learn the better features of the utility somewhere in my film. To do this, I developed a sequence where the main character awakes in an infinitely dark room where, after a series of lights turn on, he is surrounded by himself.

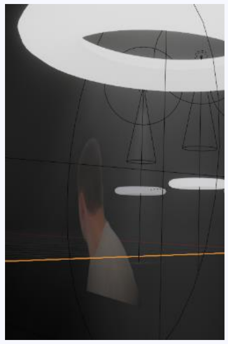
To achieve this, I filmed a short reel of me, in front of the green screen, looking to my left, then to my right, then around me. Before worrying about how to edit the clones, I started off designing the lights. I ended up using illuminated toruses with spotlights shining down through each one. Then I animated them, using keyframes, to turn on in an exponential pattern that matched when my main character was turning around. During this time, I brought the footage in my using the "Images and Planes" tool. In addition, I used a moving bump effect to simulate running water below the main character's feet.



This segment of sequences involving the 3D environment is what is taking me the longest when it comes to finishing this project. However, I want to make sure I get it just right before moving on to the incursion in my VFX journey.

Find all the files used in this project at:

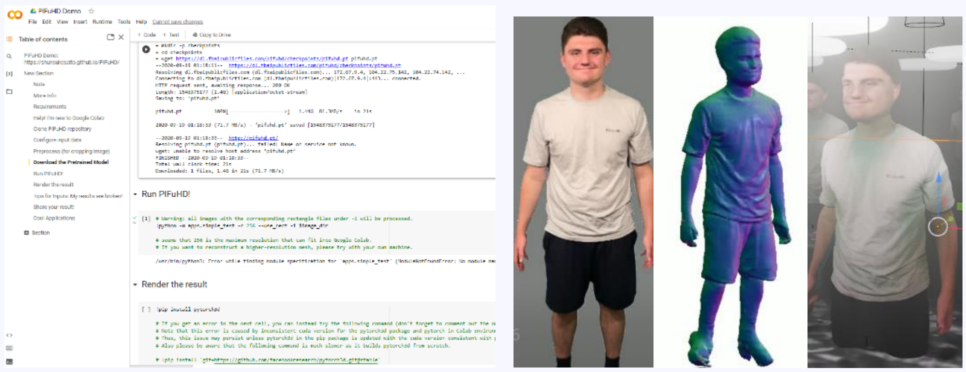
[Click Here for Google Drive Link]



Original Plane-based Plan (distorted at different angles)

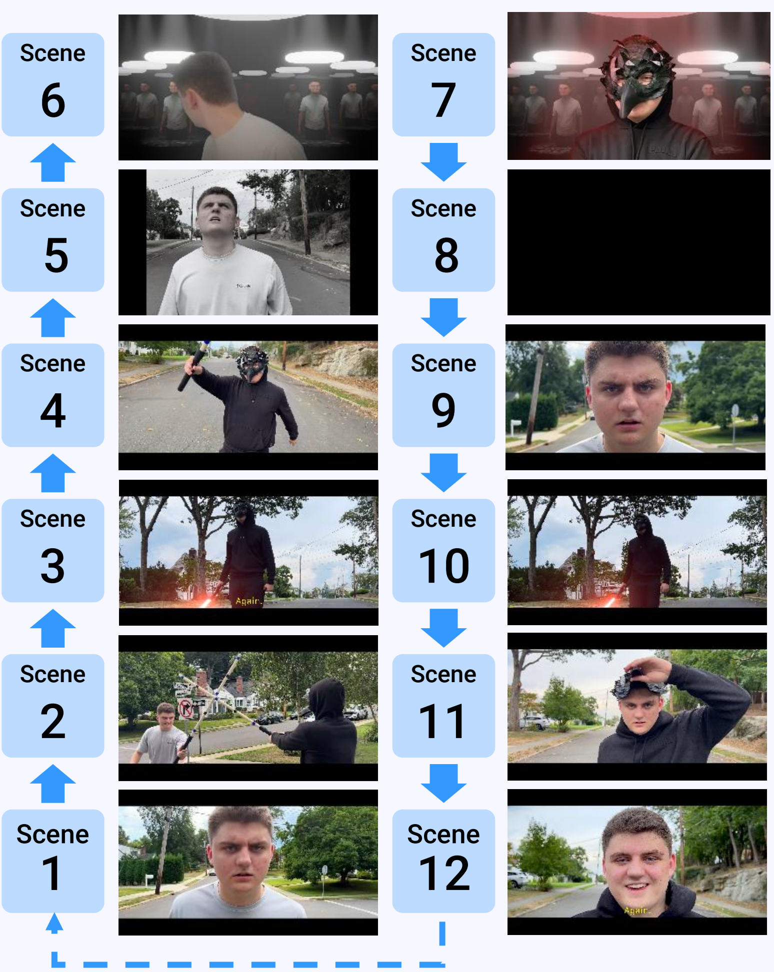
When it came to introducing the clones, it became very difficult to create a realistic effect. Originally using a series of planes that had their own recordings on them, I knew I needed to use some form of 3D model to make it believable from any angle.

To resolve this, I used a neural network from Facebook called [PIFuHD] to create a 3D model of myself from a 2D front-facing image. The result, after running their demo in Google Collab, was exactly what I needed and allowed light to bounce naturally off the clones and create a cool effect.



The setup and resulting models from the PIFuHD models (more seen in the video provided with this portfolio)

The Current "Looping" Plot / Scenes



Scenes:

1. The main character spawns across from a birdman in the middle of a street
2. The Birdman shouts "Again" in an ominous fashion
3. The main character rushes and fights the birdman
4. After a powerful swing, the main character forces the birdman to the ground
5. A tone shift finalizes the scene as the main character is seemingly transported to another world.
6. The main character appears in a black room, surrounded by water, as many clones of himself appear around him.
7. Then, following his bewilderment, one of the overhead lamps turns red and the Birdman appears under that lamp.
8. The Birdman stares at the main character as all the lights go out.
9. When the Birdman's light goes out, the main character is transported back to the beginning of the film, in the street.
10. The Birdman across from him walks forward, removing his mask.
11. It is revealed that the Birdman was the main character all around.
12. The film ends in a similar fashion to the beginning of the film with the Birdman shouting "Again" as the sequence cuts to black.

Worked with Aum Dhruv and Matthew Harty (Brother) **Role:** Director, Video Editor, Writer, and (not a great) Actor

A Lens into My Mind: Film

Climate Change Misinformation Satire News Segment

Year: 2021 | **Status:** Complete

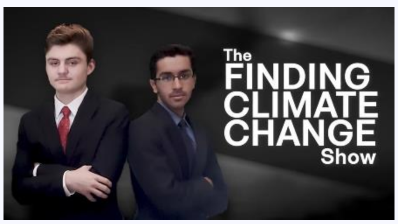
Project Inspiration

Given an assignment in my 10th grade AP English Language course to write a political satire, close friend Aum Dhruv and I saw this as an opportunity to go above and beyond and explore elements of comedic storytelling through film. With only a weekend to finish this project (as not much more than a short audio recording was initially expected from this project), we invested ourselves in a satire piece that attempted to make fun of reactionary media and their tendency to bend the facts, through post-production editing, on the issue of global warming/climate change. To do so, we used a combination of freeform interviews as well as media-structured sitdown discussions with the bias of the controlling media outlet clearly to the advantage. Its use of outrageous claims and absurd elements makes this satire an entertaining hallmark of our sophomore year that surely had our English teacher laughing.

Final Output

Development

Challenges & Solutions



The logo and intro to the parody news segment was inspired by other alike introductions (Cuomo Tonight, Tucker Carlson Tonight, etc.) that featured a series of short enticing clips followed by a premiere shot of the news anchors

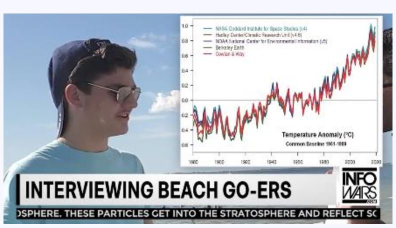
Many challenges arose when trying to animate this introduction in Premiere Pro (especially when it came to composition lag). This ultimately forced me to export the project to After Effects, animate the segment there, and import the resulting mp4 file back into Premiere Pro.

Most of the static (non-video cutouts) elements of this shot were curated in Photoshop before being imported and animated in AE.

This consistent overlay didn't follow the "InfoWars" formula for which it brands. Instead, we pulled inspiration for this overlay from primetime news segments (Fox, CNN, etc.) that were more recognizable. The rolling card at the bottom reports factual climate change information which feeds into satirical elements of what is being said on-screen.

Some of the challenges with this design included the implementation of a rolling bottom segment and an animated change in occasional titles.

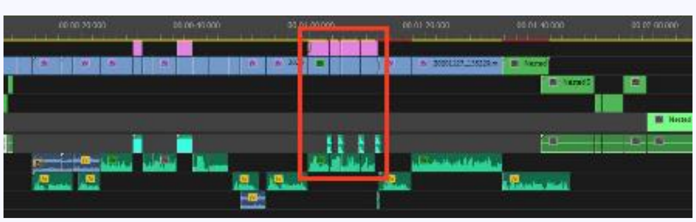
The shift in titles (Slide up and new title slides up to take its place), was managed by a Premiere "push" transition with an overlaying mask to hide the text motion that would side across the screen. This would make it look like the title was gliding up and disappearing into the top of the news bar while the new title appeared as it slide up from the bottom of the news bar.



This overlay contained a graph, chart, or photo that corresponded with what the interviewer was asking. In addition, this clip was heavily cut to provide for the exact comedic effect.

The challenge with this series of clips was mostly the timing. When my partner and I went out to film these outdoor scenes, we very much considered these "Borat" takes… in that we weren't sure how they were gonna fit in at the end.

By cutting out the interviewer questions, I was able to structure the question better in post with an additional audio overlay. Then, I trimmed the answers down to their mere sound bite and put them in series with a slight pause between each for each comedic answer to sink in.

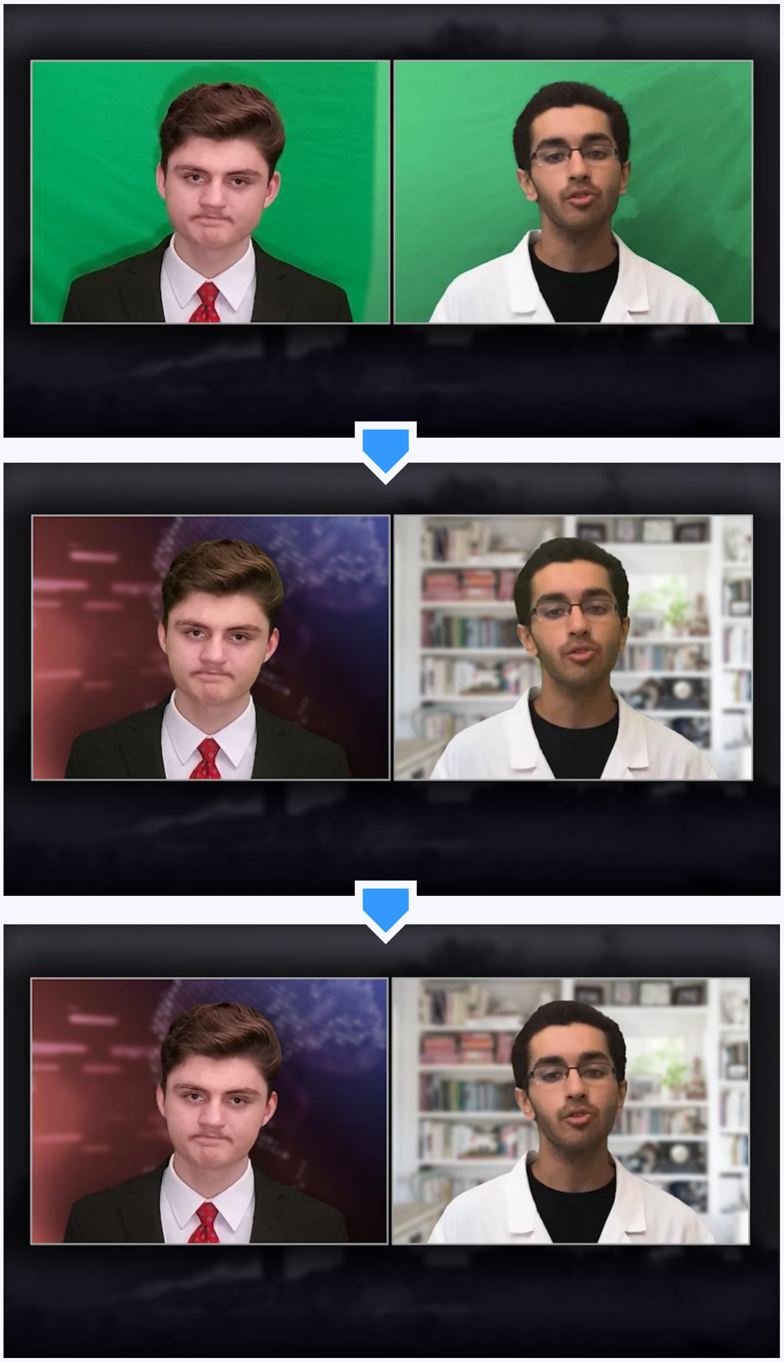


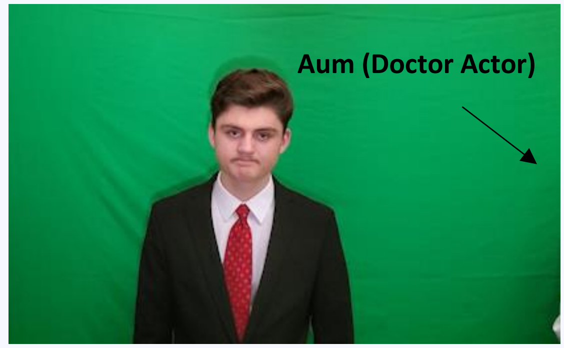
Premiere Pro Timeline

These scenes include all of the seemingly in-studio scenes that were, in fact, shot right next to each other in the back room of my house. This was my first experimentation with greenscreens and so it took a lot of tweaking, tutorial reading, and edge cleaning in order to get the effect just right.

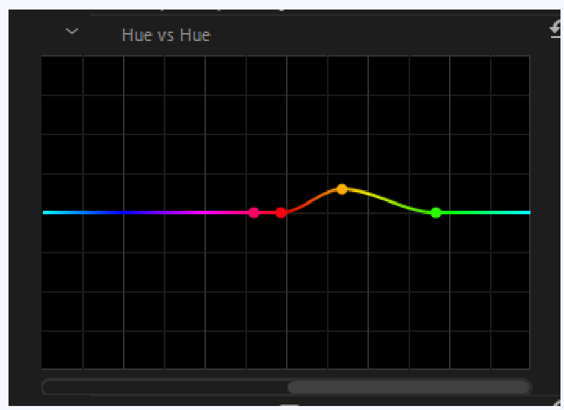
The major challenges with these shots, especially the interview shots, were the actual shooting of the scene and the post-production greenscreen editing.

After a few different setups, I rationed our single greenscreen and microphone by having us film side-by-side which, in turn, made the conversation segments easier to psychologically manage (given that the response was real rather than imaginary)





Then, in post, I began editing the greenscreen segments into their own interview boxes of which I designed in Photoshop. Following placement, I used a Luma Key to remove the green from the background. Still dissatisfied, I ended up adding in a hue shift that would ease out the green glow around the edges of my cutout. The final element of realism was a drop shadow I added to the cutout figure to simulate an actual studio backdrop.



Where to Watch?
http://film.nickharty.com/
Where to Get the Premiere Pro Files:
https://drive.google.com/drive/folders/1eCLU8ydUafjwFRS5l5CTZlxnEJaBBYwg?usp=sharing

Worked with BINGO Rich and Beef O' Brady's for testing **Role:** Programmer, Primary Researcher, User

Fun Hands-On Projects

The Most Niche BINGO Software

QR Code Printouts from BINGO sessions

Year: 2021 | **Status:** One-Off Completed Build

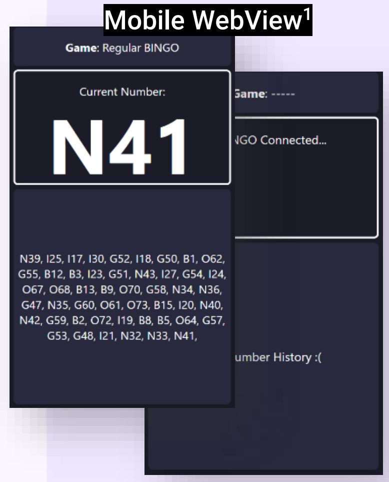
Project Inspiration

For most of my years in Florida, the local Beef'O'Brady's had become an iconic meeting place for family and friends alike. Every Wednesday night, our Estero community would crowd the small restaurant to play BINGO and catch up with others. This tradition went on for years and I was sad to see it pass when I eventually left Florida in my junior year. However, before leaving, I thought I'd pay it forward to my friend, Rich, who ran BINGO night by building him his own BINGO software. This idea came to me after a conversation we had about the limitations of his current setup and how he wished that the game wouldn't constantly be halted by new players (in need of a numerical history) or by those on the other side of the room who simply couldn't hear new numbers being called. As a fun challenge, I sought to solve these problems.

Consumer Web App

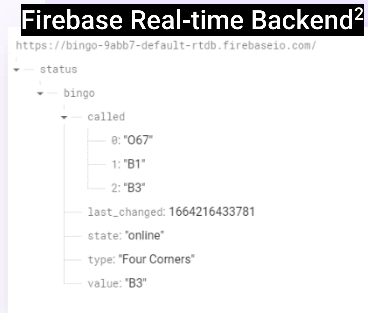
The web application was intended to conform to two factors. One was that it was meant to be optimized for mobile devices and the other being that it had to consist of simple, large text. To achieve these goals, I utilized a simple Next.JS build that read off a Firebase Real-time status[1] (which was controlled by the admin Desktop application). The challenge in building this portion of the system wasn't exactly the complexity of the application. I knew how to work in React and communicate effectively with a Firebase node. The difficulty lay in user testing and making sure that I was addressing the problem at large. The core factors I needed in this application were the name of the game being played, the current number in large letters, and a list of previous calls. These additions would provide for quicker BINGO games and less resistance overall. After a few iterations and feedback from Rich regarding the design (also recommending the use of QR codes as a gateway to the viewport), I landed on a final build that was simple, but also immensely useful for anyone playing the game (including myself).

Mobile WebView[1]



Test Site: https://bingo-gamma.vercel.app/
GitHub: https://github.com/hartytech/bingo

Firebase Real-time Backend[2]

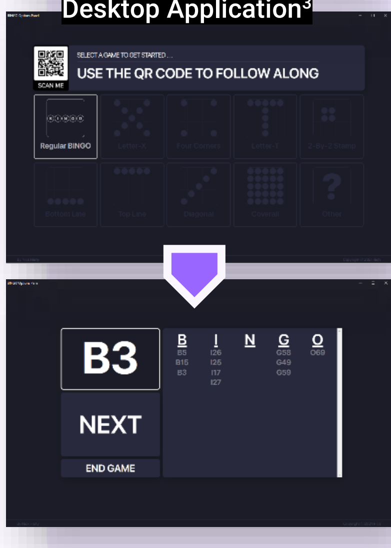


Management Desktop Application

Having used Electron.JS in the past, I found it to be a very suitable option for building a desktop application to run on Rich's Windows machine. To fit the bill of replacing his old BINGO software, I had to make it more accessible and more customized to the BINGO iterations that he played on a regular basis. In developing this application, I did just that. First, I created a main menu that would graphically and textually present the admin with an assortment of games to play. In addition, once this menu was loaded, the Firebase Real-time backend[2] would get a notification that the control system was online. Once a game was selected, the screen would change to one that presented control buttons and a list of previously called numbers. This screen would be reflected in the Firebase Real-time backend and, in turn, on the consumer application, making the whole system in sync. This was the ultimate beauty of this system: The user was able to view the information being called as it was being called and, therefore, eliminating confusion regarding previous numbers or the type of game being played.[3]

Many of the challenges I had in building this desktop application sprung from the ways in which I wanted to present data differently on both the consumer and admin side. When the admin would click the "End Game" action button, how would I halt the game on the user side? How would I deal with a database timeout that would inevitably kick in once the admin was idle for a period? I dealt with these issues by refining, over an unexpected amount of hours, the idle screen on the consumer side that would activate after the database went idle and would later re-activate once a new action was chosen by the admin (ultimately this was the function of notifying firebase if the system was online or not). The limitations of this system, which I would refine if I ever took it to market, are the specific use-case of the application and the one-lobby (meaning that only one admin can be controlling BINGO; No other rooms) nature of the system.

Desktop Application[3]



GitHub: https://github.com/hartytech/ electron-desktop-bingo

Larger Impact

Although the restaurant would, unfortunately, close down rather recently, my software system was used for 7-8 months and proved that I could build (and train people on) reliable software that I could trust to remain useful long after I surpassed any serviceable distance. This was the first time I had built consumer software that was used on a frequent basis by 30-40 people… and it sure made me nervous. But, beyond the nerves, this software provided me reassurance that lines of code can have a real positive impact on the lives of others and that my interest leads me down a path I am proud to follow.

Worked with Java the Hutts (Robotics) & my brother (Arcade Machine) Role: Programmer, Builder/Designer, Tester

Fun Hands-On Projects

Other Important Projects

Community Pi Soccer Robots

Year: 2020 (Completed)

When the 2018-2019 FTC season ended, the team and I decided to set a goal for the summer. The goal was for each of our members to build a set of fully functional soccer robots to present at community events. As I was playing soccer at the time, this idea was very much a brainchild of my personal experiences, and, because of this, I took on most of the responsibility for programming and designing the basic necessities for each of these robots before our team members began customizing their own versions.

My Final Soccer Robot

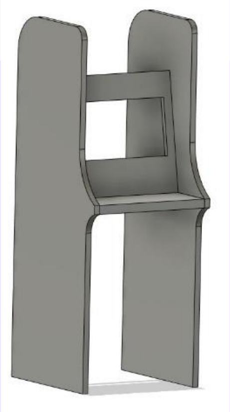
I began to design these robots on the physical side. By building a test robot, which eventually turned into my personal robot design, I was able to determine that each robot needed a solenoid, two hex motors, and a Raspberry Pi 3B (along with a solenoid hat). With this in mind, I turned to the programming side to figure out how to unify control across all of the robots. Each team member needed their robot to move forward and backward, be able to turn, and fire its solenoid.

I ended up programming the Pi's in Python, using a BLE library (along with an Xbox controller driver) to link any available controller immediately upon boot (done by running the file in /etc/rc.local - a file run upon boot).

Once these customized robots were finalized using plexiglass cutting and Rev FTC components, we used them with initial success at STEMtastics and an Imaginarium STEM exhibit to communicate the principles of engineering to elementary/middle school students.

Bonding While Building an Arcade Machine

Year: 2022 (Currently Ongoing)



Over the summer, my younger brother and I began creating Unity games in an effort to learn more about game development. I had originally started this journey with my brother because I saw his passion for pixel art development alongside his love for video games and–through a bit of convincing–was able to get him started watching Unity tutorial videos and developing some of his own projects.

Together, my brother and I developed two games over the course of the summer. One was titled "EggHop", a 2D platformer game, and the other was titled "Package Delivery Simulator" (still technically in development), a 3D object avoidance game. In the beginning, I was with him developing much of the original content and teaching/mentoring him on the basics of programming (given that C# was his first major incursion in OOP). However, as time went on, he became increasingly independent. This made me proud as an older brother and I was excited when he, after joining Engineering Club this year as a freshman, wanted to build an arcade machine to house our projects.

One problem with this dream was that, alongside him, I was very new to the traditional woodworking shop as much of my manufacturing experience in robotics had been around 3D printing and CNC wood/metal cutting. In this sense, both of us were starting from scratch.

Over the past month of working on this project, I've had a lot of fun learning how to use new machinery (drill presses, table saws, jointers, belt sanders, etc.). Currently, we are working on finishing our control board which is decked out with a Raspberry Pi 4 and a set of joysticks/buttons. Knowing there is something new to learn every time we walk into the shop in the morning keeps me and my brother passionate about this project and ready to turn a small idea into an arcade shelf that will act as the engineering centerpiece for Harrison High School.

Arcade Game Demos:

EggHop

Test Site:
https://egghophb.pixlbound.com/

Package Delivery

Test Site:
https://pdshwc.pixlbound.com/

Our Game Development Site (I made in WordPress):
https://pixlbound.com/

How Will I Expand Upon these Categories? What do I Want to Study?

"Robotics" Aspirations:

- I want to further my knowledge of sensor-based (LIDAR, Stereo Vision, etc.) navigation and dive deep into creating systems that focus on accurate localization (often through mathematical interpretation) in foreign environments.
- I also want to continue to work in teams and experience the excitement of joint learning in a space concentrated with sprawling components, various manuals, and advanced tools for creation.

"Software & Product Development" Aspirations:

- I want to learn more about the intricate nature of top-down product design that is used in consumer technology.
- I also want to continue to explore the emerging world of consumer IoMT. Working to expand my Medibound application by implementing recent FHIR (a means of communicating medical data) standards will be the first step in bringing down barriers to medical testing and increasing the accessibility of healthcare in underdeveloped areas of the world.

"A Lens into My Mind: Film" Aspirations:

- I want to explore and deepen my ability to describe fictional environments in greater realism through VFX. Specifically, I want to work with tracking points on a greenscreen or in environments such as Disney's "Volume"... plotting a real person in a fully imagined environment and tweaking every detail as I progress as a designer.

"Fun Hands-On Projects" Aspirations:

- I want to research how technology interacts with people. Specifically, I want to explore the very reason why I seek to make things. I want to study the "instance of magic in people's eyes when something traditionally complex becomes commonplace and easy to access" and how to, with any form of measurable consistency, ensure this effect when developing products for the consumer.

*Thank you for reading. These projects are only **the beginning…***