Ben Morehead
Personal Project Portfolio

Up to date as of December 16th, 2021

Table of Contents

1. Automatic Player Tracking Camera (IN PROGRESS)

1.1 Motivation:
As a youth basketball coach, I'm consistently looking for ways to use what I've learned from my engineering degree to improve my ability to help the kids improve their game. My first project of this sort is to create a jig for my tripod to incorporate automatic tracking for the camera. For this project it will be split up mainly into 3 parts: the camera tracking, movement smoothing and electronics. Currently I'm using an implementation of You Only Look Once (YOLO) in pytorch to detect the players on the court.

1.2 Summary:

Summary Point	Information	Notes
Languages Used	Python	TBD
Equipment/Tools Used	Ultralytics YoloV5, Coco Annotator	TBD
Project Length	TBD	TBD
Project Difficulty (1-10)	10	TBD

2. Drone Network in Democratic Republic of Congo (IN PROGRESS)

1.1 Motivation:
For my year long graduation project, me and four others are helping a local startup working on assisting medical equipment transportation in the Democratic Republic of Congo. For this project we are looking to introduce a Low Power Longe Range (LoRA) network amongst the drones making deliveries so that any reliance on cellular service, which is limited in the region, is eliminated. So far we have prototypes of the receivers and transmitters, and we are looking to begin designing our network architecture next.

1.2 Summary:

Summary Point	**Information**	**Notes**
Languages Used	C/Arduino	TBD
Equipment/Tools Used	ESP8266 WiFi Module, LoRA V2 Modules	TBD
Project Length	2 Semesters	TBD
Project Difficulty (1-10)	9	TBD

3. Augmented Reality Climbing Wall (IN PROGRESS)

1.1 Motivation:

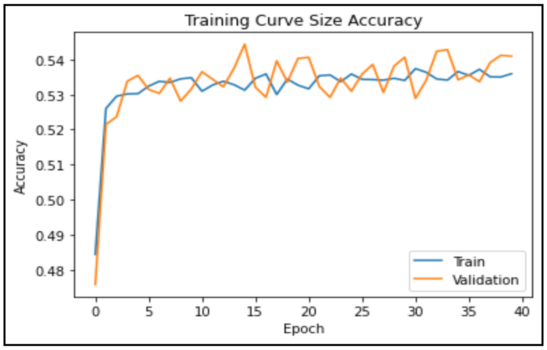
This one is a fun one. A group of my friends are working on building a climbing wall where you can play a game on it. This has been done before, but we are looking to use the resources we have to recreate it for ourselves. As of right now we are looking to use a Kinect to provide a frame feed into a pose detector from the MediaPipe library. That will be our main input that determines the game logic as we can easily track wrist and ankle location with respect to different holds on the wall.

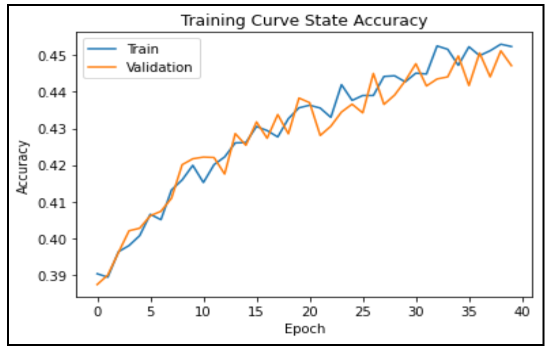
1.2 Summary:

Summary Point	**Information**	**Notes**
Languages Used	Python	TBD
Equipment/Tools Used	Kinect V2, Touch Designer	TBD
Project Length	3 Months	TBD
Project Difficulty (1-10)	10	TBD

4. Wildfire Prediction Model

Finished: December 2021





1.1 Motivation:

In my introduction to artificial intelligence course a group of us decided for our final project to explore how well we could predict american wildfire size and location based on data from the US government. We wanted to use a recurrent neural network (RNN) to do this, and just see what was possible using that architecture. It also helped us with understanding the inner workings of these models, to open up more opportunities for project ideas in the future.

1.2 Design Details:

<u>1.2.1 Architecture</u>

For this project we designed a custom RNN using LSTM units to improve some of the problems with long-sequence RNNs. The RNN encoding that came as an output of the network was then passed into two separate fully-connected networks to handle the classification of geographic area, and size. Using the pytorch library we were able to do so quickly and effortlessly. We trained the data on more than a million data points, and produced reasonable results with this methodology.

<u>1.2.2 Results/Success</u>

We got very mediocre results from this method, reaching ~50% accuracy on the test data that was separated completely from the training and validation data. This is essentially the same as a human guess, as it took historical data and guessed which of 4 most common areas were most likely to have the next fire, and likewise for the size classification. It was a good lesson in which problems fit which types of models and helped with setting up proper infrastructure for a solid and clean machine learning problem.

1.3 Summary:

Summary Point	**Information**	**Notes**

Languages Used	Python, SQL	SQL was used to load the data points for the national forestry agency into a pandas dataframe.
Equipment/Tools Used	Pytorch	Used the pytorch library to make the RNN
Project Length	2 Months	Very bare-bones and low-level project so didn't take very long.
Project Difficulty (1-10)	5	Generally easy to complete, but needed a lot of debugging when it came to loading the data via SQL.

5. Cascading Stair Lights

Finished: May 2021

1.1 Motivation:

This project came from a desire to use some of my skills for making an aesthetic adjustment to my living space. Underneath each of our stairs is an LED strip that lights up in a cascading fashion when you take the stairs. It ultimately saves us from having to turn on any additional lights. I did this project with my roommate Stephen Gidge. He wrote an awesome write-up about the full process (https://gidge.dev/stair%20lights/stair-lighting/)

1.2 Design Details:

<u>1.2.1 Analog Light Circuit</u>

There are 16 strips of white LED lights that are attached to the bottom of each step. They all are connected in parallel to the output of a simple transistor circuit. In order to ensure we had enough power for all of the lights we calculated the length we would need initially, and from the specifications could determine the current draw needed for each strip. We ended up only needing a 12V 15A power supply to properly power all of the strips.

<u>1.2.2 Digital Logic</u>

With each of the lights connected to a transistor, a digital signal coming from an arduino mega was used to turn the transistor on and off. We initially were going to use 2 arduinos with serial communication, but it was easier to have all the outputs on one system. The cascading effect is done using basic code using the arduino IDE.

<u>1.2.3 Sensors and Detection</u>

To invoke the lights we installed 2 ultrasonic sensors (one at the top of the stairs and one at the bottom). We tried our best to have a flat surface in front of the sensor, so that the waves didn't deflect too much, and although our top of the stairs sensor works very accurately, there's a curved wall at the bottom of the staircase so sometimes there are false positives.

1.3 Debugging:

<u>1.3.1 Analog Light Circuit</u>

This project took close to a year and a half almost solely due to the transistor circuit. Our first iteration used mosfet transistors and a soldered prototyping board circuit. We tested it with the

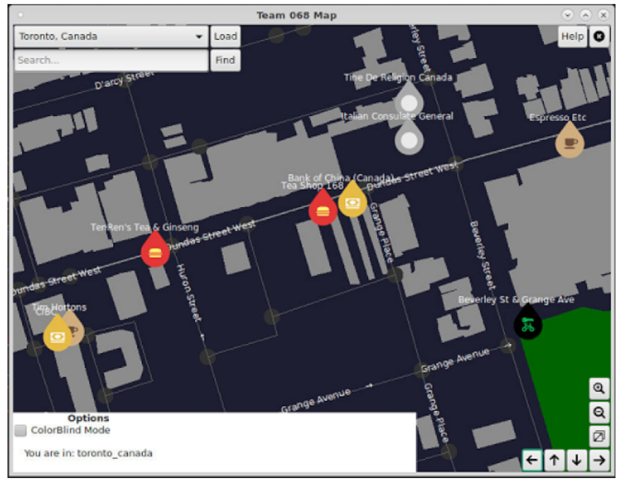
LEDs throughout, to make sure none of the connections were shorted, but after we connected the lights to the stairs we didn't do any additional testing. Once it failed after being fully hooked up, we redesigned our transistor circuit to clean it up and improve our testability. We changed our transistors to digitally rated BJTs and used a breadboard instead of a protoboard. This design actually needed 0 testing as it worked on the first attempt.

1.4 Summary:

Summary Point	**Information**	**Notes**
Languages Used	C/Arduino	Nothing impressive here, it's just a basic arduino/c++ loop
Equipment/Tools Used	Analog Circuits, Arduino	Rating the transistors properly can be very tricky, so testing on this should be prioritized.
Project Length	1.5 Years	It took a long time but we did get it done!
Project Difficulty (1-10)	6	Nothing in the project was out of the ordinary, but it took a while to ensure it was done at a high quality.

6. Custom GIS Software

Finished: April 2019



1.1 Motivation:

In my second year of Electrical and Computer Engineering I took a course dedicated to designing a custom GIS software in C++. The engine for every project in the course was the same, and we were supposed to differentiate ourselves by the UI features we could add, and our approach to classic mapping algorithms such as the Travelling Salesman Problem. Overall this project was a great introduction to a larger scale software project.

1.2 Design Details:

<u>1.2.1 Back End Engine</u>

The problem for this project was separated into two parts: how can we acquire geographical information, and how can we visualize it. The acquisition of information was done through a GIS C++ library provided to us by the course, with which we could make api calls to get a large variety of data points on a provided mapping area. These data points contained information such as intersections, points of interest and environmental features.

<u>1.2.2 Front End UI/Graphics</u>

The front end was done using the EasyGL library. This UI library allowed us to visualize all of the data points mentioned earlier, as well as add some features to our software that made us stand out. Most significantly, I designed a search bar with automatic completion and directions provided on the selection of a location. Although fairly basic in concept, it took a lot of experimentation to get fully accustomed to EasyGL, but the skill of figuring out how a software library works is one I’ve carried through in my career since.

1.3 Summary:

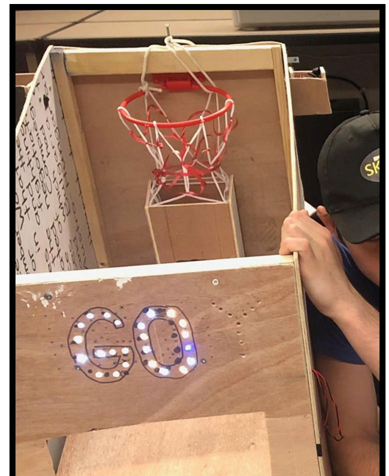
Summary Point	Information	Notes
Languages Used	C++	This was my first real experience utilizing external software libraries, so it was a good chance for exploration and experimentation
Equipment/Tools Used	N/A	Pure software project

Project Length	1 Semester	Project for my ECE297 Communications course
Project Difficulty (1-10)	7	A little challenging, but overall most of the work was in fixing the fine details rather than major design decisions

7. Table Top Arcade Basketball

Finished: November 2018

7.1 Motivation:



In the fall semester of 2018, I was taking a digital systems course and my partner Lisa and I were tasked with designing a custom project that uses the DE1-SOC FPGA Board (the main tool used within the class). Being a fan of basketball, I saw a way in which a small scale basketball arcade game could be implemented using the FPGA Board, leading to a functioning game that incorporates some basic electronics, digital game logic, and display output.

7.2 Design Details:

<u>7.2.1 Physical Structure</u>

The physical game was made out of some scrap 2x2s and thin plywood. The frame exists as a slanted box so that after every shot the ball can roll back to the player unprompted. The hoop was purchased from the local dollar store and attached to the wood base, and some masonite and wood glue were used to create a compartment to limit the ball from bouncing too much outside of the box.

Attached from the back in the compartment is a photoresistor. This allowed us to actually track when a ball went effectively through the hoop, as the photoresistor would experience a darkened environment. The photoresistor was then attached to a simple mosfet circuit that allowed a strong signal to be sent to the GPIO port of the FPGA board, which in turn lead to a logic signal to increase the score.

<u>7.2.2 FPGA-Game Logic</u>

The game state-system was implemented with Verilog. With basic on board accessories such as the GPIO and on-system clock, we were able to create a time based game that would reset the score at the end of the timer, and update the high score respectively. The game was started with a button located on the side of the physical structure, that then fed a signal to the GPIO of the FPGA.

<u>7.2.3 VGA Output</u>

The VGA Output was implemented through a process we learned in class. The work involved creating bitmap representations of the different elements we'd want to show on screen. We then used those files in our pre-existing VGA output module to properly display information acquired from the sensor and timer.

7.3 Debugging:

7.3.1 Physical Structure

The major issues that arose with the physical structure surrounded tracking the ball. A score was initially tallied using a trip-wire style laser and photoresistor combo, but there were inconsistencies with where the ball would fall through the hoop, leading to no interruption of the photoresistor setup. To fix the consistency issue, I changed the environment of the photoresistor to be in a shaded box where the ball needed to fall, and as the ball passed by the resistor the circuit was able to detect that change in light.

7.3.2 FPGA-Game Logic

Game logic was fairly standard. An on-board timer determined how long the analog circuit would track the ball for, highscore was saved in local random access memory, and there would be a full reset on the completion of the game. Given the simplicity of the system there was not a lot of debugging to be done.

7.3.3 VGA Output

The VGA had limited bugs as it used a module acquired from the course.

7.4 Summary:

Summary Point	**Information**	**Notes**
Languages Used	Verilog	Was my first large project with Verilog, gained more experience later on in follow-up courses.
Equipment/Tools Used	DE1-SOC FPGA Board, Simple BJT circuit with Photoresistor, Basic woodworking	A lot of experimenting with the physical product, the software and logic was more straightforward.
Project Length	1 Month	Was the project for our digital systems course so a month of course time was allocated for it
Project Difficulty (1-10)	7	It was my first experience using very low-level digital hardware.

8. (Bonus) Two Floor Connect Four

Finished: November 2019

8.1 Motivation:

We just wanted to big a really big connect four, so we did.

8.2 Design Details:

8.2.1 Pieces

The pieces were made out of basic plywood with a 2 foot diameter. Around the side of it we wrapped so malleable plastic so that it would brush up against the frame when being dropped, taking away some of the force from the final impact.

8.2.2 Main Structure

The pieces get placed into a very basic constructed compartment, one for each column of the game. To catch the pieces there are bungee cords going across all of the bottom of the structure. We used the smooth side of some masonite boards and a basic triangular prism to create a ramp in which the pieces get slid into from the second floor.

8.2.3 Safety Considerations

The biggest problem, which is very important when doing a massive project like this, is user safety. We designed the structure to use leverages of the local environment to ensure that when freestanding it couldn't be moved at all, and even once we did that we tied it up to 6 different points.