**About Me**

Hi! I’m Anthony. Currently I am busy studying mechanical engineering at the University of Waterloo. While studying is my focus, I also work on fun side projects, and hone other relevant engineering skills. In the future I hope that I can work in the aerospace industry, as I am fascinated by our ability to make huge objects travel at high speeds in the air. When I am not working, you can find me spending my time hitting the gym, playing volleyball, or gaming.

**Launcho**

**What?**

Launcho is an innovative toy, created by a team of 5 people including me. It is a spring powered rocket launcher that can fire on command while being able to adjust its angle of fire in all 3 dimensions.

A visual representation of the product was created. The objective of this was to get a more concrete idea of the end goal. This model was created in Solidworks with each individual component being modeled then added to the assembly.

A scaled model was also created using Solidworks. Each part had to be dimensioned in a way that does not interfere with other parts of the assembly. Eventually, this model was used to 3D print several parts

Some parts had to be stronger than 3D printing allowed for. Thus, other materials such as steel and PEX were used. However, these had to be machined to the correct dimensions. Above is a picture of the machined PEX pipe.

**Results**

Throughout this project, I picked up many useful skills. First, I improved tremendously with Solidworks as I spent a lot of time trying to model everything to a T. Specifically, I improved at assembly, mating, and modeling more complex geometries. Moreover, I also learned 3D printing. Contrary to my initial belief, 3D printing is actually extremely tedious. This was learned through the countless failed printing attempts we had during the project.

**Codebreaker**

**What?**

To summarize, codebreaker is a software revolving around the game called “mastermind” (see here). My software allows you to play a classic game of mastermind, it allows for 2 players to face each, it allows the user to create a code for 3 different levels of AI, and it also features a leaderboard.

**How?**

Player versus player was accomplished by having 2 different players play the game on one machine. They would take turns inputting their guesses.

Easy mode AI simply guesses at random, hence “easy mode”. As you can guess, this AI will almost NEVER guess your code as it will not adjust its guess base on a previous guess. To put it in retrospect, this AI has 4 guesses each with a 1/256 chance of being correct.

Medium difficulty AI works a little differently. This AI will base its guess off its last guess. The first guess will be random but every guess after that will have logic behind it. The logic is that, if there is a correct position and color, it will guess the exact same color for that slot again, and if there is a correct color only, it will keep the color and change the position. This way, it is a bit smarter than the easy AI

Hard difficulty AI is really hard. Actually, it is impossible to beat. This AI utilises an algorithm called Knuth’s algorithm. Essentially, it will compute the guess which eliminates the maximum number of possibilities from the number of possibilities that are left. This AI will ALWAYS break the code in under 5 tries.

The leaderboard is just a simple text file updated with Java. It keeps track of best score (based on time and guesses) for classic mastermind.

**Results?**

This project mainly helped me improve my proficiency working with classes. However, I also learned a lot about the Knuth’s algorithm. It was a pretty hard algorithm to grasp, and even harder to apply. Overall, I am happy with the outcome.

**Saitama**

**What?**

In short Saitama is a general-purpose discord bot. But what is a discord bot? It is server hosted application which is able to interact with a chat service called discord. Specifically, Saitama is able respond to certain messages keep track of interesting user statistics, execute commands such as ban, and once upon a time, Saitama was even able to play music. However, since discord kept updating their API, this function no longer works.

**How?**

Since discord API was only available for python and JavaScript, I could only choose between those 2 languages. In the end, I decided to use python because I already had some prior knowledge of the language, and it was easier to work with.

A virtual environment was created to store all necessary libraries for the project since I did not want to store them on my main machine. Some of these libraries include YouTube-dl for YouTube to mp3 capabilities, and obviously discord.py to connect to the discord servers.

.json files were used to store important information that might need to be changed in the future. These files are useful because they are easily accessible and alterable with code or by hand.

The overall structure of the project utilized several classes, each in charge of a different function. This way, the code did not get cluttered, and it was easy to navigate and read.

\*To see the source code, check out my GitHub repository

**Results**

I learned a lot about virtual environment, discord API, and general python. Starting out this project, I had no clue what a virtual environment was, no clue how to build a discord bot, and barely had any python experience. Now I can say that I am confident I would be able utilise these tools again.

This Website

What?

This website started out with the intention with me trying to create a personal portfolio. For the most part, it is just a simple website with some bugs.

How?

I created this website using html for a base, CSS to style, and JavaScript for more advanced functions.

This website is hosted straight from GitHub with Netlify

Results

As it was my first time doing web development, I learned a lot about the process. Specifically, I learned about html, css, domains, and hosting. This is an ongoing project so I may learn more in the future.