**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 or playing volleyball.

**Centrifugal Pump**

**What?**

In this project I developed a centrifugal pump system (impeller and volute casing) that effectively generates suction for a robotic arm. The purpose of this arm is to transport objects via vacuum force.

**How?**

In the beginning, a lot of research was done to understand the underlying mechanisms working behind the pump. It was found that a smaller cross section for the inlet and a wider cross section for the outlet is the most ideal, as it can more effectively generate a low pressure zone near the inlet. This can be understood with Bernoulli’s principle which states that higher velocity = lower pressure. In the flow simulation above, it can be seen that the air near the inlet is flowing quickly (green), whereas the air near the outlet is flowing slowly (blue). This means that the pressure near the inlet is low while the pressure near the outlet is high (Bernoulli’s principle). Since fluids naturally flow from high pressure areas to low pressure areas, the air near the above the inlet will try to even out the pressure difference by entering the inlet (atmospheric pressure is higher than pressure near the inlet). This is the opposite for the air near the outlet (pressure in the outlet will flow to atmosphere since outlet is higher pressure) thus, causing a suction affect.

After understanding the theory, numerous prototypes and models were created using Solidworks. Many of these prototypes were printed and tested to collect data and make further improvements. Flow simulations were also conducted to make improvements and ensure our centrifugal pump would work properly.

**Results**

After fabrication, the impeller can create suction with the use of only one Lego EV3 motor and carefully chosen gear ratios. The suction strength was also tested on a napkin weighing ~4g in which it succeeded.

**Brake Caliper Mounting Arm for Solar Car**

**What?**

In this project, I designed a brake caliper mounting arm for Midnight Sun’s 2023 solar car. This mounting arm would hold the brake calipers stable while using the car’s suspension system as a mounting point. The brake caliper which was being mounted was Wilwood’s GP200 brake caliper.

**How?**

Extensive research was conducted to determine the optimal caliper placement which would influence the design of the mounting arm. Factors such as center of mass, airflow, and simplicity were considered.

After research, numerous prototypes were produced in Solidworks, utilizing 2D caliper drawings. The designed was then further improved through FEA simulations to remove unnecessary materials, lowering the weight (10% reduction in weight achieved). Moreover, DFM principles were considered while designing to ensure low manufacturing costs and time. For instance, standard hole sizes were used, and no abstract geometry was used.

**Results**

My design allows for easy manufacturing, which in turns lower manufacturing cost and time. This means that Midnight Sun can relocate the saved money to aid other areas of the solar car’s development.

**Gravity Car**

**What?**

A gravity car is a vehicle which uses the gravitational energy harvested from a weight to power itself. Specifically, in our project, we use a 1kg mass hung 60 cm in the air, so a total of ~5.88 J of energy.

**How?**

The whole car was build using Tetrix parts, as they are easy to manipulate and sturdy at the same time. There were some key design factors that allowed us to build a fast gravity car. First, we used carefully chosen gear ratios to cross the finish line (2m away) in the shortest amount of time. Second, we used 1 wheel rear wheel drive to reduce the moment of inertia, allowing for faster acceleration. Lastly, We removed the rubber portion off the front tires to reduce the moment of inertia as well. However, the rubber was kept on the back tire to minimize slipping which reduces the efficiency and speed of the car.

**Results**

Our car ended up winning against over 30 other cars in a competition of speed. This competition was to see who could make it to the 2m finish line the fastest.

**Launcho**

**What?**

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

**How?**

To begin, a nerf mechanism was studied to gain an understanding of how a launching mechanism might work. After this research, an initial prototype was built to assess the feasibility of the design. This prototype was fabricated with PEX pip, Nerf gun parts, and duct tape.

After obtaining a solid understanding of the mechanism, and validating design feasibility, a 3D model was created using Solidworks. In total this modeled contained 9 individual intricate components which was then assembled.

Using this model as a reference, some components were 3D printed, and some were machined out of steel. 2 different manufacturing methods were utilized to achieve both the precision and durability required.

The final product was then assembled (via force fits and super glue), and spray painted for aesthetic purposes.

**Results**

The result is a projectile launcher which can consistently launch to a height of 1.5 m (vertical). Although it may not seem like much, the mechanism relies on the power from only one nerf spring. Additionally, the wide base prevents any sort of tipping during use.

**4 Cylinder Engine**

**What?**

**How?**

**Results**

**Codebreaker**

**What?**

To summarize, codebreaker is a software revolving around the game called “mastermind” (see here). These are the features of my program: allows you to play a classic game of mastermind, allows for 2 players to face each other, 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 it is now against Discord’s policy, 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**

To be honest Saitama exceeded my expectations. I thought that this project would come to a halt due to how little I knew about the topic. However, I managed to create a bot which could respond to messages, keep track of statistics, execute commands, and even play music.

**This Website**

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.

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

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.