Quiz 'N' Chew

An Engineering Design Report

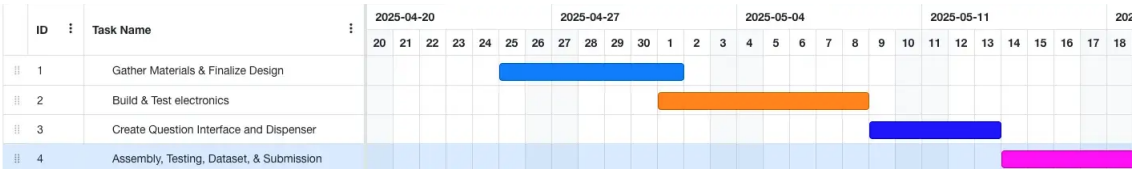
Cody Coleman
Woodberry Forest School

May 27, 2025

Contents

1 Project Description

This project was based off of adrenaline and how it affects the psyche. The basics behind the project were that it should be a mix of Engineering and Computer Science, allowing both sides to come together in one way. This is through the Arduino Uno and the coding softwares C++ and Python. When first establishing this project, a bill of materials and a Gantt chart was made to understand what was needed before randomly buying items as well as a something to keep the project moving on pace. The Bill of Materials can be accessed by clicking here. The project time management can be accessed here.



2 Research and Background

When researching this project two things were kept in mind, how adrenaline helps studying, as well as what materials would be the best to fulfill this project. Adrenaline is a tool that spikes in your memory to separate something from another. Adrenaline can increase your anxiety levels, but it also highlights key moments, which can help with memorization. When a piece of candy drops, it makes the average person feel excited, which can boost someone's adrenaline. When looking into tools for this project these showed up:

- Arduino Uno
  - This device allows everything to connect. This device is the brains behind everything, allowing the Python code to connect the C++ code, which then moves the servos when a question is correct.
- Breadboard Wires
  - Allows the board to power all the resources needed, ground everything, as well as use digital pins so that the Arduino knows which pin is the servo.
- A Breadboard
  - Allows wires within so that everything remains constant and nothing fluctuates.
- 3D printing materials.
  - Resources to actually create the dispenser, so that the servos can actually move something and store candy for the drop.

- Servos
  - One of the most important parts of the whole project, the servos move the container so that when the user gets a question right it turns down and drops the candy.

3 Specify Requirements

- Functional requirements:
  - Dispenses treat only when the answer is correct.
  - Accepts serial input ("correct" / "wrong").
- Constraints:
  - Limited to two servos.
  - Simple user interface (e.g., button input or keyboard).
  - Powered by Arduino.

4 Brainstorm and Select Solution

When brainstorming, originally the project was going to drop one candy after every single question. The idea seemed interesting, but eventually moved towards dropping multiple pieces per drop. This was due to the fact that there was a time constraint and at the time, I had no clue on how to design something that would drop a singular piece of candy. This later would lead to a dropper technique where the servos would both move and a door of sorts would open and drop candy. When thinking about where this would be coded, the first constraint had to be that it needed to be coded in C++, due to it being an Arduino. Although, later it was realized that you can combine multiple languages, so Python was added to the project. Tkinter was the best resource for graphical user interfaces, so that immediately was put on the table when brainstorming for the project.

5 Develop and Prototype

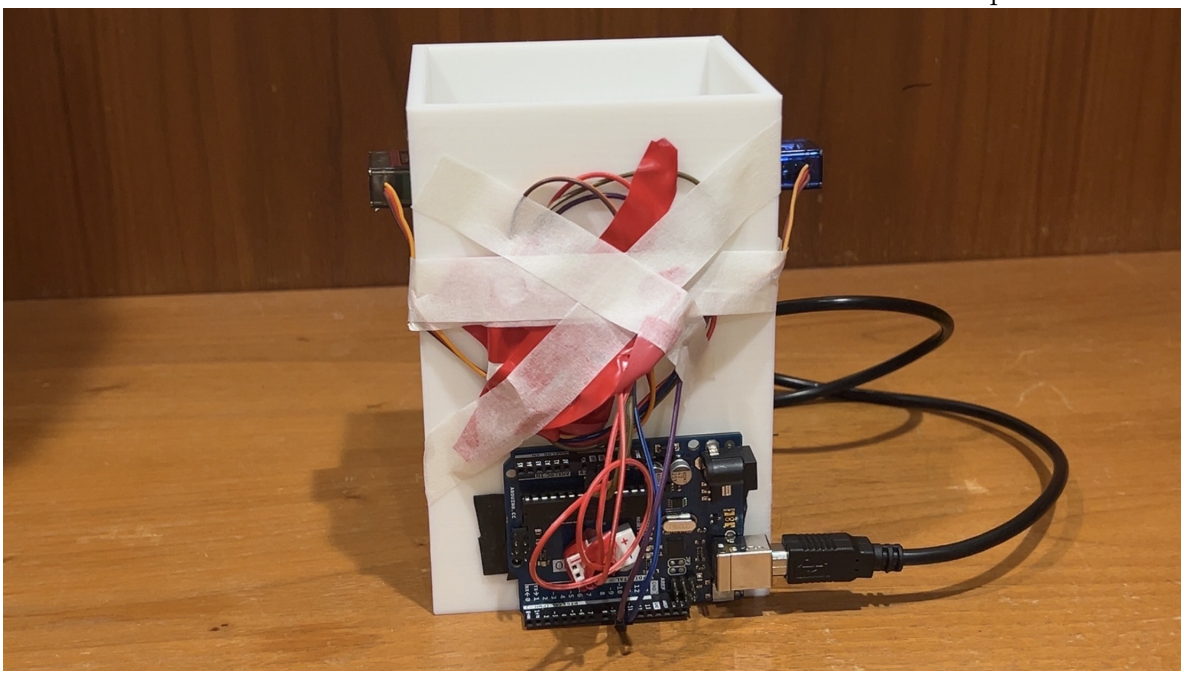
- Brief overview of the hardware setup:
  - Arduino Uno
  - MG90S & MS18
  - Power Cable
- Overview of the software logic `(reference to main.cpp & python.ipynb).`

Some of the hardest challenges when prototyping were the actual code and designing multiple prototypes for the dispenser. Multiple designs were used eventually landing on a design with one section for a servo as well as a small ramp that when the servo moves, it dispenses the candy.

That particular design however had a lot of inefficiencies, some of which included

- The physical tray being too saggy
- The ramp being too short
- The tray falling out of the servo

Because of this, the design was changed to have a much steeper ramp as well as having two servos this time. One of the servos being much stronger, that being a MG90S rather than a MS18. The new model also uses a bread board for even power distribution.



From there, the questions were made. These questions were all physics-based questions made through ChatGPT.

6 Test and Evaluate

The goal behind testing the model was to find the write angle to drop the perfect amount of candy without jamming it, as well as seeing how hard the questions were. It was found that moving the servos 25 degrees in each direction was the optimal movement for both. Since each servo were in different directions, one would have to be wrote to 50 degrees and the other to 0 degrees. From

7 Final Product

The final product worked like a charm, it dropped the candy using the dispenser without really any issues. There were several problems that were brought up or found that could be fixed in later versions though.

Those being:

- Tkinter buttons
  - The Tkinter buttons were very buggy, resulting in having to click the answer multiple times or finding the right spot on the graphical user interface. This could be fixed through sensitivity issues, however the right amount still has not been found, it may also just be a VSCode issue, which is what both sets of code were coded with.
- Candy pieces
  - The next being the candy pieces, many people asked if it could be singular instead. From what was heard, it is definitely better to do it one piece of candy per question solved, which can be fixed in a different version through a extended dropper system.
- The ramp
  - Lastly, the ramp could have been even more steeper, it would still keep candy on it due to the friction of the plastic.
- The Wires
  - The wires were everywhere and eventually had to be taped down. A fix to this could be a section where it can hold the wires rather than taping them down. This would make the model not just more structurally stable, but also look a lot better.

Some future additions could be:

- Web scraping
  - When first beginning the project, the possibility of web-scraping came up, I tried using Quizlet or Khan Academy, but both had strict terms of service against the fact. With request from these sites, web-scraping may be a possibility.
- Physical buttons
  - The possibility of physical buttons was expressed at the beginning of the project as well. It was believed that it wasn't possible due to not being able to connect to the breadboard or the Arduino Uno, but when talking with someone it was exclaimed that it is possible through soldering. This could make it more physical as well as much more fun.

Overall the project was a huge success, hopefully in the future it can be expanded if time is found.

8 Code Appendix

Arduino Code

main.cpp

Python Code

python.ipynb