Explorer Post 1010

Team 20-6696

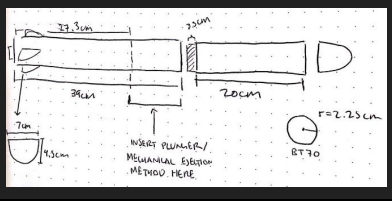
Cole Sherling, Jack Sherling, Sam Troost, Madison Zhao

1. Design Process

Design/Simulation → Construct → Launch → Revise

Brainstorm/Design
- Hand sketch design
- Create OpenRocket file

Simulation
- Check: Flight Simulation
  - Success→ Construct
  - Fail→ Revise + Repeat Step



Build
- Cut body tubes
- Laser-cut fins

Polish
- Sand + fit

Secure
- Glue Type
- Nuts and Bolts

***Code**
- Write initial code
- Flight computer sled

Assemble
- Booster
  - Motor
  - Wadding/ Dog Barf
  - Parachute
- Cargo
  - Egg
  - *Flight Computer
  - *Servo +Air Brakes

Test/Launch
- Parachute Check
- Record Measurements

Analyze
- Download Altimeter data
- Compile data
- Graph

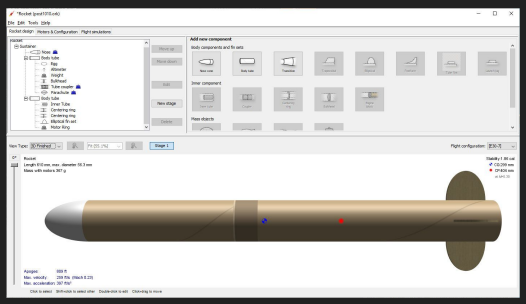
Tweak Parameters
- Weight
- Motor Type
- *Refine code

*Only for Design B (aka Flight Computer Design)

Sam

In order to make successful rockets, we went through a planned design process. First, we brainstormed rocket designs and sketched them out. After that, we created an OpenRocket file and tuned the parameters according to OpenRocket's flight simulations. Once we had a design that would theoretically reach the goal, we started the construction process. We laser-cut the fins out of basswood and cut BT-70 body tubes according to our design in OpenRocket. We also wrote the first iteration of our code for the flight computer design. Once we have a finished rocket, we assemble it and head to the launch site. We record all variables like the exact weight and wind speed. After flying multiple times in a day we record the flight time and altitude in a spreadsheet. We use that data to revise the design and tweak parameters like weight, the motor type, and the code accordingly. (We'll speak more about the analysis in a later slide).

2. Final Design A (Ballistic)



Dimensions:

- Tube diameter: 56mm
- Height: 65 cm
  - Cargo: 20cm
  - Booster: 39cm
  - Fins: 7cm (Diameter)

Construction Materials: BT-70 Cardboard Tubes (Body Tubes), Basswood (Fins), PVC nose cone

Weight: 280 grams

Motor: E30-7T

How were these decisions decided?

-Questions considered→ Aerodynamics? Reusability? Egg Placement?

-Adjustments made→ Round fins. Egg placed below Altimeter

Sam

For the body, we used BT-70 cardboard tubes because they are relatively lightweight and a good diameter. We designed and cut the cargo to be 20 cm long which allows us to fit the altimeter, the egg, and any extra weight we may add to hit the target altitude as well as any electronics. The booster section is designed to be 39 cm long, long enough for the rocket to be stable. The fins are cut out of basswood which is easily cut with a laser cutter and is lightweight, making it ideal.

Cole

When designing a rocket, we needed to make sure it was stable. OpenRocket maps out our center of mass and center of pressure and runs stability calculations for us. OpenRocket also allowed us to run simulations using our rocket, different motors, and parachutes to achieve optimal altitude and time. We always want more control over the altitude of the rocket. In the past, despite using motors of the same model and lot number, there was still a lot of variance in our altitudes. This is when we began looking into microcontrollers and air brakes which we'll discuss in the next slide.

3. Final Design B (Flight Computer)

- Microcontroller & pnut altimeter
- Light signals display status of flight computer
  - Blue=ready for launch, Yellow=problem, Red=success
- Added window allows view of light

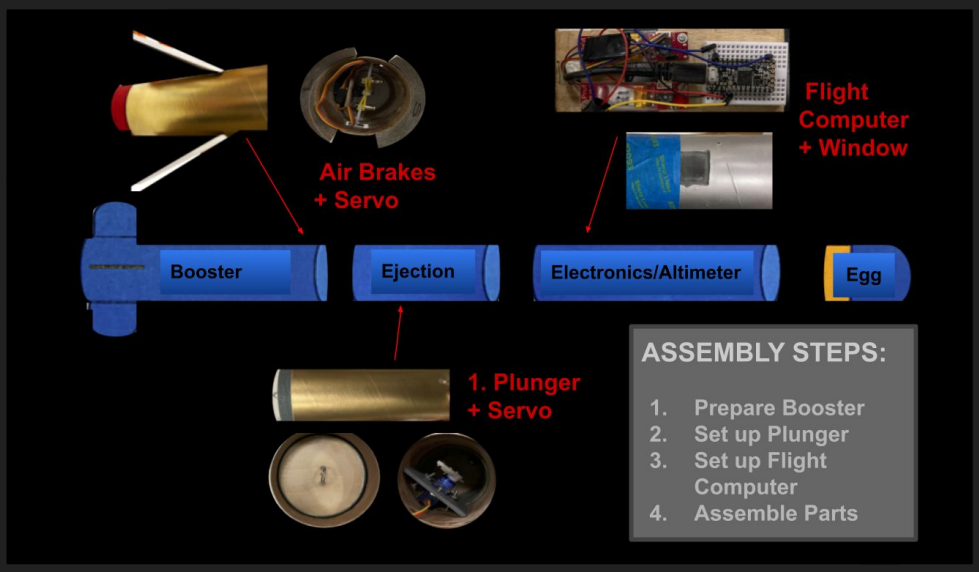
```
if (altitude < 0.8 * maxAltitude and altitude > 30) or (altitudeCount > 2):
    deploy()
```

```
def deploy():
    global deployed
    serv.angle = 180
    deployed = True
```

Jack

This year, we've started developing a rocket which uses a flight computer to increase consistency. Previously, we would have to put all of our faith in the exact quantity of propellant in the motor, the speed of the wind, and so on, but now we have the ability to control for altitude using air brakes or a plunger ejection (discussed in the next slide). The rocket has two altimeters: one for the official judging and one which is connected to our ItsyBitsy microcontroller. Our microcontroller is running CircuitPython, which we chose for its easy capabilities of storing data in files (making testing easier). We have a lighting system using the dotstar light on our microcontroller. If the light is blue it means that we are ready to launch, if it is yellow there is a problem, and if it is red the microcontroller was successful (we would see this after the launch).

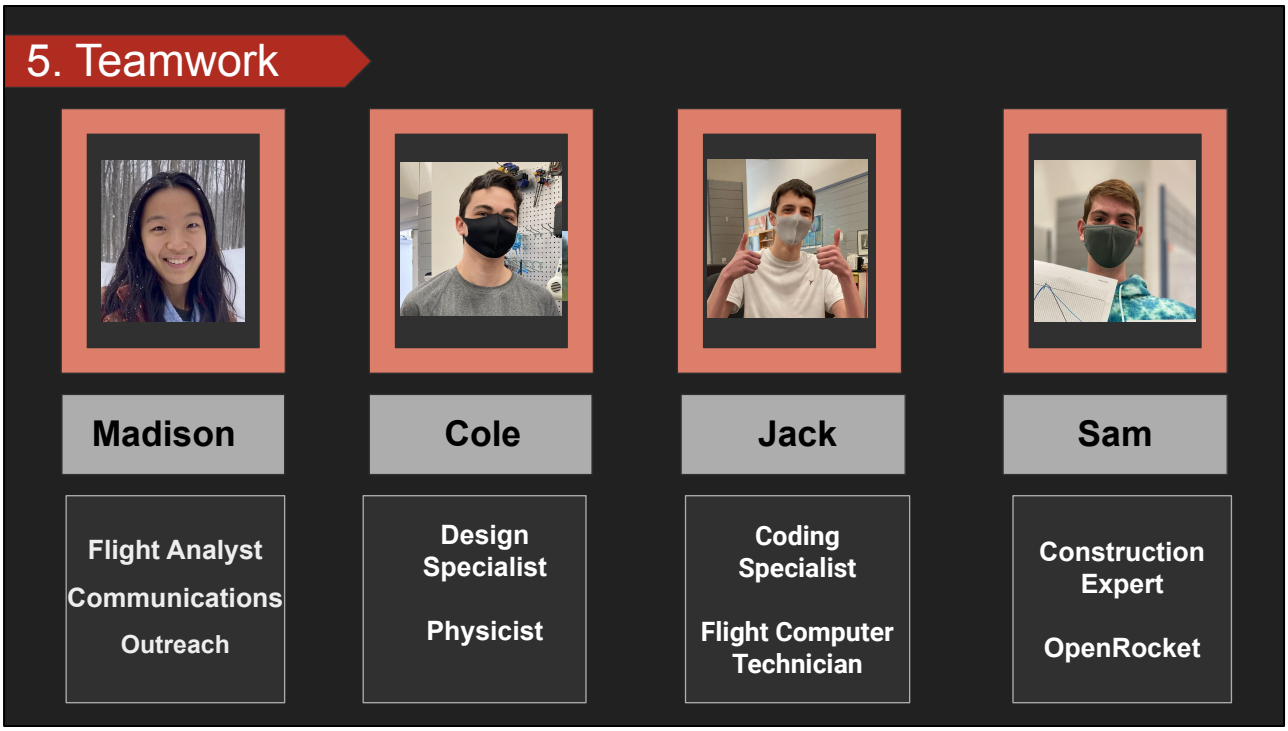
4. Final Design B (Manual Ejection)



Jack
For the air brakes, we calculate the velocity of the rocket by finding the change in altitude over the time. From this, and incorporating air resistance, we can determine our expected apogee. If the expected apogee is below our 800 feet target, we will do nothing, because our air brakes would only reduce altitude. However, if our expected altitude is above 800 feet, we will open the air brakes until our expected altitude reaches 800 feet. We will continue to check our expected altitude to be as accurate as possible.

For the plunger ejection, we check if we are above a certain altitude, such as 775 feet. If we reach this altitude, we call our deploy function (seen on our previous slide) to spin a servo which activates the plunger, forcing the parachute out before the normal ejection charge.

Sam
The air brakes are cut out of BT-70 body tubing so they match the diameter of the rocket and won't produce any unwanted drag before they're deployed. Both the plunger and air brakes are driven by a single servo motor. For the air brakes, the servo pulls on rubber bands that extend the air brakes in and out. The servo in the plunger releases a spring and separates the rocket at the right time. If the plunger doesn't separate the rocket properly, the motor ejection charge will still safely separate the rocket as a fail-safe.



Madison
We split up the roles based on our individual strengths. That's actually how we split up writing the presentation as well! I've probably been on the team the longest (4 years now) so that's why a lot of my roles are more managerial. I coordinate launches andlook for outreach opportunities (Ex. USA Science and Engineering Festival and Cub Scout events pre-covid). I also help analyze any data we record.

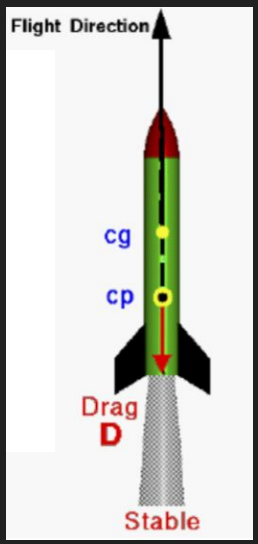
Cole
I handle the design aspects of the rocket. What fin shapes to look into and what nose cones work best. I also contribute information that I learn in my high level physics and aerospace engineering classes to our meeting discussions.

Jack
I've been programming for a few years now and have designed websites in the past. My main contributions to the team include doing the coding for our flight computer.

Sam
I'm also a member of Narhams and have a lot of knowledge on the physical construction process. We use a laser cutter to cut our fins and OpenRocket software to create our design so I do a lot of work turning our design into a reality.

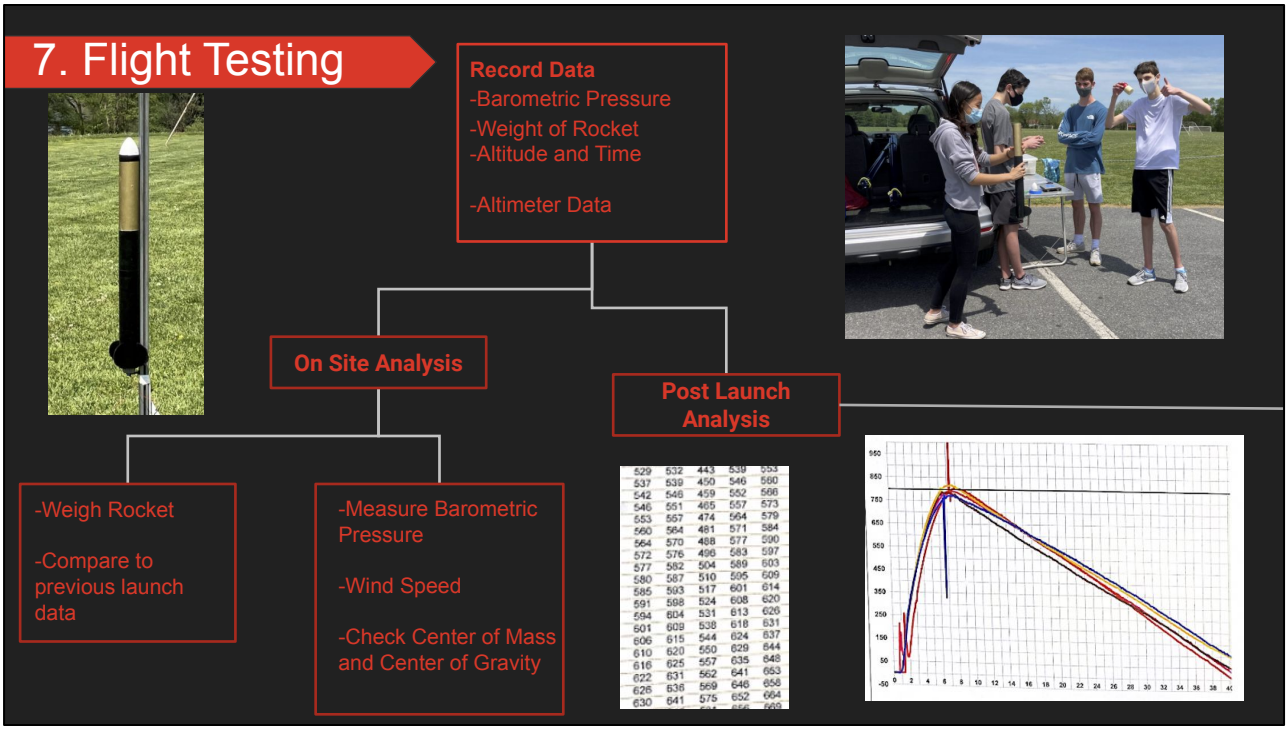
6. Rocket Science

00	Factors	Adjustments
01	Drag	• Semicircle fins are most aerodynamic • X-Chute shortens descent time • Refer to OpenRocket for CP, CG, Drag
02	Weight	• Motor Selection→ E30-7T • Insert variable mass to account for flight computer weight
03	Weather Conditions	• Track barometric pressure + temp. • Coordinate early launch times ○ Avoid heat thermals and wind • Record Data!



Cole

We have spreadsheets with time and altitude data from every launch, which we analyze to determine the impact of weight and height on the performance of our rocket. When the barometric pressure is high, we adapt by adding less weight to the rocket because the air is denser, thus producing more drag. We adjust our variable mass in the nose cone to try and control altitude. When a test flight goes too high, we add more mass. When one goes too low, we remove mass. We tape the mass into the nose cone for consistent weight distribution. We designed our rocket to have its center of gravity further up than its center of pressure for stability. One thing we had to consider were the pressure fluctuations when opening the parachutes at high speeds through our mechanical plunger ejection mechanism. We chose the E30-7T motor because it, based on past experience, got rockets of this weight to 800 feet. We chose semicircular fins since they are more aerodynamic than polygonal fins.



Madison

Although we get an idea of how our design flies with the OpenRocket software, we try to record as much data as we can every launch as the actual flights always differ from the hypothetical. We document through pictures, altimeter, wind speed, barometric pressure collectors, etc. When it comes to analysis, we've categorized it into analysis we do while on the field (immediately before flights) and after launches, which we do back in our engineering makerspace.

This slide covers our on site analysis. Immediately after assembling our rocket, we weigh the rocket, check wind speed and barometric pressure. Depending on how much time we have, we sometimes also run a test to check the balance of our rocket by tying a string around the rocket at the center of gravity and swinging it slowly in a circle. This helps ensure that our rocket is stable and does not wobble while in flight. However, we get most of our data collection from our altimeter. After returning from launches, we download the data and can create graphs of our flight path (see visuals of 5 flights above).

8. Flight Testing (Cont.)

Record Data
Barometric Pressure
Weight of Rocket
Altitude and Time

Altimeter Data

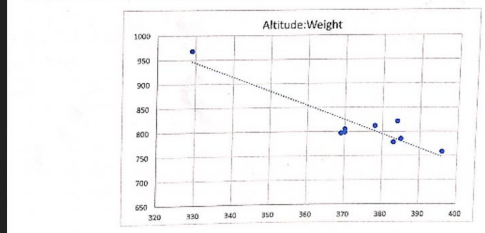
Post Launch Analysis

-Scatterplot and Regression w/ Altitude v. Height

-Download Altimeter data

-Graph flight projectiles

Date	Launch Site	Tem	Baro	Wind	Team	Rocket	Fli	Motor/Lot	Ignitor	Weight	Alt	Chute	Altitude	Time	Comment
20-Jan	Walkersville	30	29.80	9	6696	Black	1	E30-7T		329	13	18"	668	43	Too high
20-Jan	Walkersville	30	29.80	9	6696	Black	2	E30-7T		384	13	18"	821	46	Zipper
20-Jan	Walkersville	30	29.80	9	6696	Black	3	E30-7T		386	13	18"	758	44	Booster came off
17-Feb	Walkersville	47	29.92	3	6696	Black	1	E30-7T 110526	FFJr	378	13	18" X	812	45	Nice flight, score=20
17-Feb	Walkersville	47	29.92	3	6696	Black	2	E30-7T 110526		385	13	18" X	785	41	Nice flight, score=15
8-Mar	Walkersville	45	30.09	6	6696	Black	1	E30-7T 021120	package	383	13	18"X	778	37	Too low
8-Mar	Walkersville	45	30.09	6	6696	Black	2	E30-7T 021120	package	369	13	18"X	797	39	Nice flight - score=7
8-Mar	Walkersville	45	30.09	6	6696	Black	3	E30-7T 011520	package	370	13	18"X	799	38.4	Qual 1 - score=7.4
8-Mar	Walkersville	45	30.09	6	6696	Black	4	E30-7T 011520	package	370	13	18"X	805	45.8	Qual 2 - score=16.24
24-Apr	G. Meadow	60	?	7	6696	Black	1	E30-7T 012220	package	373	16	18X"	837	26	Chute didn't fully deploy
24-Apr	G. Meadow	60	?	7	6696	Black	2	E30-7T 012220	package	381	16	18X"	802	43	Nice flight, score=2
								E30-7T							
								E30-7T							



27-Jan	Walkersville	45	29.38	10	6696	Black	1	E20-7W		383	13	18" X	713	36	Turned into wind, not straight
27-Jan	Walkersville	45	29.38	10	6696	Black	2	E20-7W		365	13	18" X	597	19	Sputter ignition, not straight

Madison

Besides creating projectile graphs of the flight, we create a spreadsheet with basic information about each flight (date, location, pressure, wind, weight, etc.). We also type up notes for each flight in the margin. The two most important pieces of information we collect is the altitude and the weight. Last year, after graphing scatterplots of our flights we found out that there seems to be a linear relationship between the altitude and weight (at least in our ballistic model with our specific motor). We discovered that for about every gram of weight we add, the rocket's altitude is 2 feet lower. Here above, we have a graph with our actual data and a regression line that represents the expected height and weight. Our ballistic model is very light this year so we have room to add weight depending on how high we want the rocket to go. For finals, depending on whether the target altitude were 775 feet or 825 feet, we can use the scatter plot and regression to estimate what weight our rocket should be to go to the desired altitude.

9. Challenges

-COVID→ less team interactions, construction time, etc.

-Windy launches→ rockets stuck in trees and decreased altitude

-Bad weather on launch dates

-Not enough wadding→ ripped parachutes

-Parachutes and descent time

Everyone

In these past two years now, we've encountered a lot of challenges. For one, COVID restricted our team to move weekly meetings online, and resulted in us not having as much access to our makerspace to design our rockets. Besides these unanticipated difficulties, though, the most challenges we faced were still on the field. We discovered last year that weather was a big factor to watch out for. When we fly in the winter, the ground is often frozen and the fins often break upon impact. On the flip side, launching in warmer weathers often results in thermals that impact our flight times. Launching in windy weather led to a lot of drifting. We had our rocket stuck in a tree and spent nearly an hour trying to get it down! We also ran into a few issues when assembling the rocket. We switched to a x-chute this year since our times were consistently above the target time range with a normal parachute but it took us a while to figure out the best way to wrap our x-chute.

10. Lessons Learned

- **Save Designs/Record Data**→ Most important: Weight, Design Measurements, Altitudes and Times
- **Designate roles** for efficiency and consistency
- Motor Selection: Same **lot number**=more consistency
- Run **ground tests** for parachutes and flight computer to avoid failure in the air
- **Check wind** speeds
- Bring poles and bow and arrow...

Everyone

What these challenges meant was that we had to come up with a lot of solutions. For future years, the most important advice we have learned is to save designs and record everything! On the chance that a rocket is not recovered from the launch, it's important to have all designs and modifications saved. We realized that a factor that contributed to unnecessary variability during our launches was switching up our roles. We decided that by having the same team member pack the parachute every time, another pack the wadding and so on, we could make our flights more consistent. One interesting fact we learned was that Enerjet creates their motors in different batches. Different batches often differ slightly in how high they go. One way we made our flights more precise was by using motors from the same lot number. To check the parachute, balance, and flight computer of our rocket, we run tests on the ground to avoid mistakes during the actual flight. Finally, on the chance that our rocket lands in a tree, we bring extendable poles and sometimes a bow and arrow to retrieve our model!

11. Thank You

Thanks for Watching!

Everyone
Thank you for watching our presentation! Feel free to ask any questions about our design and we'll be happy to answer.