Rocket Flight Simulator – Documentation

1. System Requirements

1.1. Software Requirements

To run this software, first the user should be able to run Python on their machine. The specific version does not matter, so long as it is Python 3.0 or later. The user can use any system OS, again so long as they have Python installed and working properly.

The user can choose to run the software in an IDE like VS Code, or even in their terminal.

1.2. Required Python Libraries

This library uses a few Python libraries which the user must have installed on their machine. These include 'math', 'Matplotlib' specifically the 'pyplot' module, and 'NumPy'. These are pretty common Python libraries, which many users may already have installed, however it is important to make sure your machine does for the simulation to run properly. Matplotlib and NumPy can be installed using the links below.

https://matplotlib.org/stable/install/index.html

https://numpy.org/install/

2. Installation and Setup

2.1. Downloading the Project

This project can be installed anywhere on your machine where you can easily access the files. This could include a folder on your desktop, in your downloads folder, or any other directory on your system.

This project only requires the one 'Rocket-Simulation.py' file to run. Everything needed for the simulation is within this file and its code alone. The other design and documentation files can also be found in this folder 'Rocket-Simulation-Project'.

2.2. Setup

The first step for setup includes ensuring your system has a Python version installed on your system. You can check python is properly installed on your system, by simply typing the command 'python' into your terminal, and if it is properly installed it will return the version of python your system is using.

Next, you want to ensure you have all the necessary libraries mentioned earlier installed on your system.

Last, ensure the Rocket-Simulation.py file is installed somewhere you can readily access it, and then you are ready to start using the simulation!

3. <u>How to Run the Simulation</u>

3.1. Opening from a terminal

To run the simulation from the terminal, first the user must navigate to the folder where the project file is stored. On Windows, the best way to do this is to navigate to the folder where the project is stored in through File Explorer, and at the top of the application where the folder path is shown, right-click and select “Copy Address as Text”. Then, open your terminal, and type the command ‘cd {file-path}’, where {file-path} is that address you just copied from the File Explorer. Then hit enter. You should now see your terminal is operating from the chosen file path you just entered and has the necessary project folder open.

Next, to run the program, just type ‘python Rocket-Simulation.py’, and press enter. This will begin the program.

3.2. Using the Simulation

The simulation should now be open. The program will first greet the user and ask them if they would like to use a preset rocket or input their own parameters. For a first-time user, it is recommended to use a preset, so they can familiarize themselves with the program. Following your selection, it’ll either give you a few preset rockets to choose from or ask for you to put in a set of parameters that describe your rocket and the simulation.

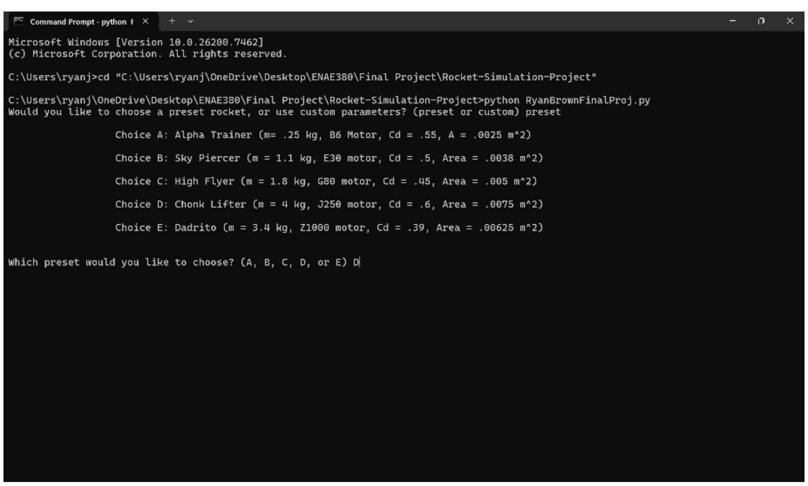

Figure 1: Screen showing the user choosing a preset rocket. In this case, they choose D.

Once the user has selected their preset, or input all of the necessary parameters (if using a custom simulation), the simulation will begin. The user will receive a message showing the rocket has been created, so they are aware their simulation was set up correctly. They will then see a live graph of the rocket's altitude, as a function of time.

When the simulation is done running, this live graph will close, showing the user a few different graphs. One plot shows the altitude and velocity as a function of time, highlighting important points like apogee or peak velocity. There will also be a graph highlighting the rocket's energy breakdown over the period of the flight. These graphs can be saved directly as images and used for the user's own purposes and interpretations.

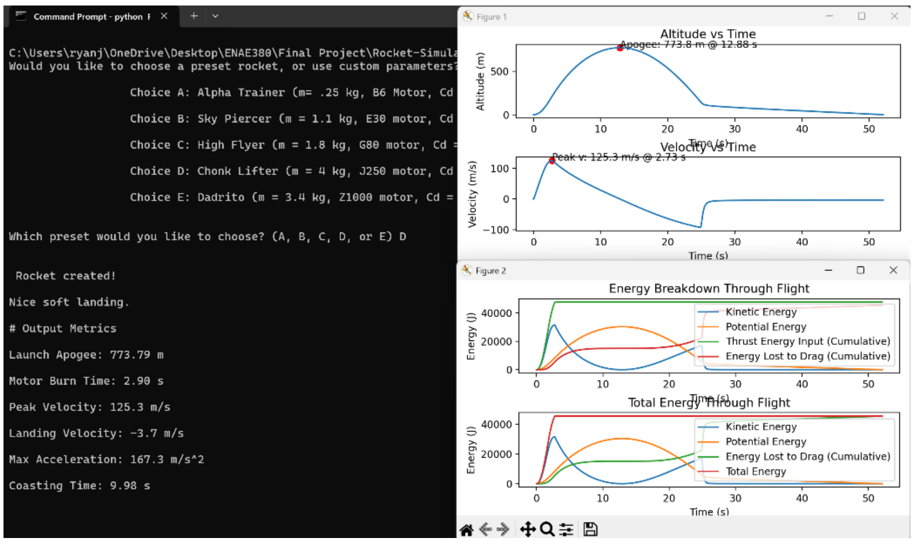

Figure 2: Details of the simulation after it is complete. Output parameters printed to the console on the left, graphs on the right.

As you can see in Fig. 2, the console will also print out some output parameters, which the user can use again at their own discretion.

3.3. Saving your results

Once you are finished looking at your graphs, close them. This will then prompt the software to ask the user if they would like to save their results to a .txt file. This file can be named whatever the user would like and will include the input parameters of their simulation (body mass, motor choice, chute area, etc.) as well as all the data points within the simulation, formatted as {time, altitude, velocity, acceleration}. These are all in metric units, (s, m, m/s, m/s^2).

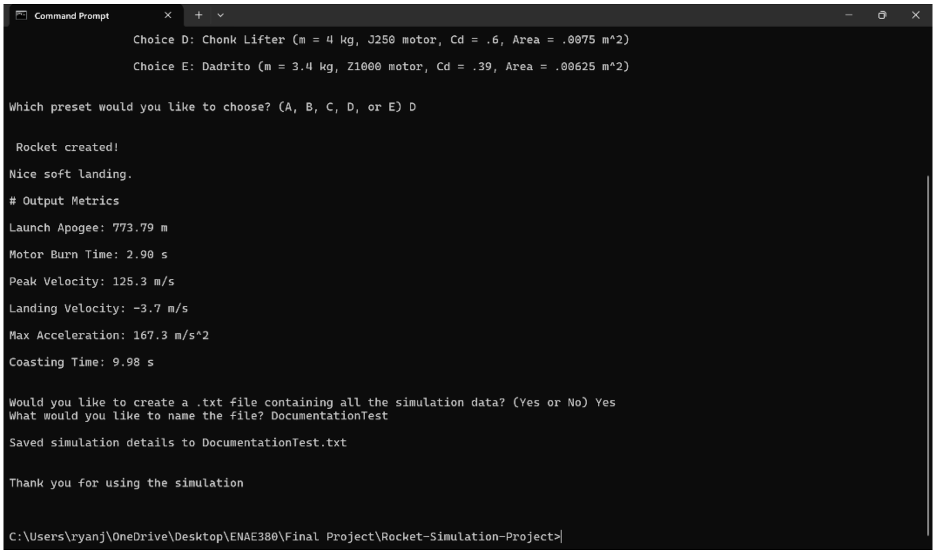

Figure 3: The user is saving their simulation results to DocumentationTest.txt.

This file will then be saved to the same folder as the Rocket-Simulation.py file is stored in. The user can then access the details of their simulation and use that information for whatever further processing they need.

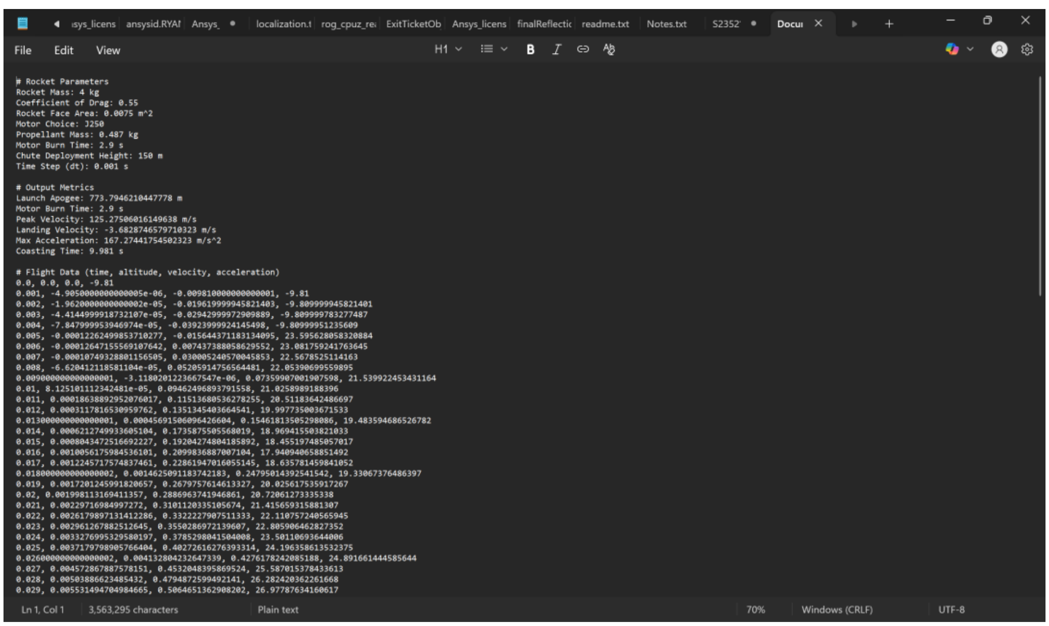

Figure 4: What the saved .txt file will look like.

At this point, the software will close, thanking the user.