Rocket Calculator

Made by:
Nicholas Luis, Colin Morphy, Dominic Marazita

Introduction

The purpose of this project was to combine our unique Aerospace knowledge of orbital mechanics with the skills we have learned in AERSP424, Advanced Computer Programming.

This program has three distinct capabilities:

1. Creating multistage rockets and simulating the ΔV from them
2. Analyzing potential orbital maneuvers given said ΔV
3. Looking up real satellite data for analysis of its orbit

The latter is especially useful after a rocket launch because you can simulate some new orbits available after you

Limitations

There are some limitations to the program due to the assumptions we made, since orbital mechanics can be very complicated at times.

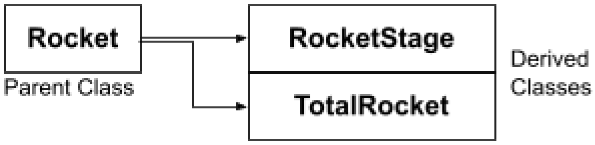
For example, we assume that the inclination changes always occur when the orbit crosses the Earth's equatorial plane– a common assumption because it reduces the amount of fuel required. However, we still need to know the velocity that the spacecraft is traveling at as it crosses this plane:

- If the orbit is circular, the velocity can easily be calculated by knowing the altitude that the spacecraft is orbiting at
- If the orbit is elliptical, the velocity at the inclination change is dependent on the

Classes / Functions

Rocket.h & Rocket.cpp

These files contain 3 classes: A parent '*Rocket*' class and two derived classes, '*RocketStage*' and '*TotalRocket*'.



Rocket Class

This contains variables/functions that are shared between both the individual stages and the whole rocket. Examples include a variable for the mass of the rocket and functions to get/set said mass.

Rocket Functions

- Print function that uses a console-output mutex to prevent multiple threads from printing at the same time
- This mostly contains virtual functions, which are overridden by the derived classes

RocketStage

This essentially creates objects of single-stage rockets, each of which has values for the structure mass, fuel mass, and specific impulse of its engines.

RocketStage functions

- getter/setter functions for the fuel mass, structural (empty) mass, and the specific impulse of the engines

TotalRocket Class

This creates an object that combines the single-stage rockets into a bigger, more capable rocket

TotalRocket Functions

- A get function, *getDeltaV*, that returns the ΔV if the rocket were to burn all of its fuel
- A get function, *getDeltaV*, that returns the ΔV if it only burns a specific amount of fuel
- A function that adds the *RocketStage*s to a queue container
- A function that detaches the stages (removes the objects from the queue)

Orbit.h & Orbit.cpp

Orbit Class

This class simulates an orbit and gives feedback on the possible orbital maneuvers available when given a certain ΔV value to work with.

Orbit Functions

- Getter/Setter functions for the properties of an orbit such as the inclination,
- Functions that return the possible orbit maneuvers given a certain amount of ΔV

Satellite.h & Satellite.cpp

Satellite Class

This class contains variables and functions for information on a pre-built satellite or a satellite which has been input through a file. These include the name, launch number, and eccentricity.

Satellite Functions

- An ifstream function that will open a file on satellite data
- A getline function that will read the input file line by line
- A regex function which extracts data from the file and assigns values to the correct corresponding variables

ConsoleManager.h & ConsoleManager.cpp

ConsoleManager Class

This class manages the console window with a mutex, which prevents multiple threads from outputting to the console at the same time

ConsoleManager Functions

- Print functions that prints a given string
- A set function that takes a user input and stores it in a given variable

GuiManager.h & GuiManager.cpp

GuiManager Class

This class manages the GUI, which includes 2 different windows, as well as several buttons, text boxes, and tabs to interact with.

GuiManager Functions

- Display functions that updates the GUI
- Getter functions that read from files and converts the data into strings

imgui...

These are all part of an online graphics library we used to create a GUI.