USER MANUAL
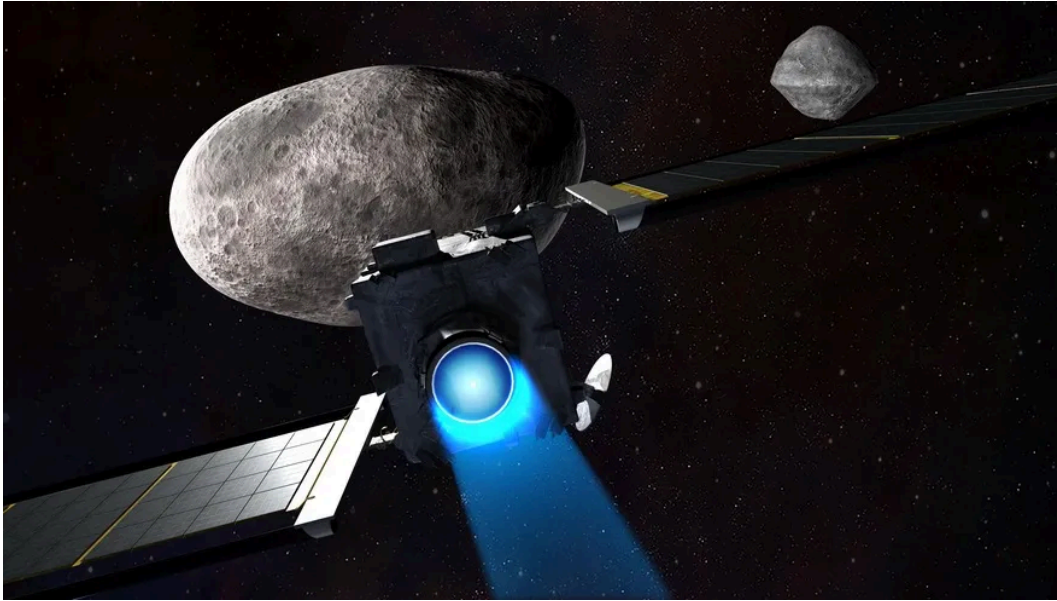
D.A.R.T. Collisions - UWB Space - CSS 458

Credit: https://spacenews.com/nasa-ready-to-launch-dart-planetary-defense-demonstration-mission/

UNIVERSITY *of* WASHINGTON

BOTHELL

Table of Contents

Introduction:

Our D.A.R.T. collision program is based on creating a stable solar system based on real world data, with modified assumptions. We assume there are no gravitational influences aside from the eight primary planets, our sun, and our randomly generated asteroids. We assume no changes in mass for our objects. We assume perfectly spherical objects. We assume no introduction of interstellar objects during simulation. With a generated solar system, the model can be adjusted with dozens of keyword arguments to analyze the effects. The program is capable of performing many different analyses, including the effects of hitting an asteroid with a D.A.R.T. projectile.

Note: D.A.R.T. is the Double Asteroid Redirection Test. NASA's first attempt at redirecting an asteroid's flight trajectory by striking it with a spacecraft.

Authors:

Kyle Williams - *University of Washington Bothell, CSS 458*
Nora Osmanova - *University of Washington Bothell, CSS 458*
Josh Farkas - *University of Washington Bothell, CSS 458*

Requirements:

- ☐ **Python 3.13.5**
- ☐ **NumPy**
- ☐ **Matplotlib**
- ☐ **PyTest**
- ☐ **Copy**

Usage:

Tune the analysis file's run_single_test function to produce a model with desired values, using keywords in the model declaration. Run the analysis.py file to produce analysis graphs and optionally an animation. Analysis file can be modified to produce different analyses and animations.

Files:

model.py - Model contains a class that generates and houses all of the body objects. Model's class has dozens of init keyword arguments to adjust for different model behaviour. Model initializes planets, asteroids, and D.A.R.T.s. It calls the body class to update its velocity and position parameters n amount of times for each existing body.

data.py - File containing body object declarations with real data from the nasa horizons project for mass, radius, position, velocity, and label. Used by model to create body objects.

body.py - Houses data for objects defined in data.py file. Contains logic for updating velocity and position of body objects using runge-kutta timestepping and net-force calculations.

asteroid.py - Inherits body class. Contains logic specific to asteroids.

dart.py - Inherits body class. Contains logic specific to D.A.R.T.s

planet.py - Inherits body class. Contains logic specific to planets.

animation.py - Creates a matplotlib animation of data provided by the model class. Allows you to adjust which body is centered in the animation as well as the observation space with keyword arguments.

analysis.py - Performs analysis on model results by calculating different parameters related to the bodies and tuning the model attributes. Primary entry file to running the model. Pairs analysis with animations.

_test.py - PyTest suite for testing different functions and class declarations throughout the program.

References:

https://ssd.jpl.nasa.gov/horizons/app.html#/