

Explanations of the program

Roiron Yohann yohann.roiron@mines-paristech.fr

February 29, 2016

this document is made to present the java program done for the SpaceX interview. I was asked to do:

In order to assess your Java level, we'd like you to develop a quick app. As you're in simulation, we'd like you to graphically simulate the launch of our Falcon 9. Don't bother with details, we just want a working and realistic app. So i decided to code a physico-realistic application which as to simulate the launch of a rocket in a 2D. earth centred space. The aim of the program is to simulate the influence of the fuel capacity in this 3 bodies environment.

1 Physics

We are going to models the system as a three body, one is fixed: the earth, one moving around the earth following the 2nd law of Newton, and the last one Is The rocket that we are trying to launch without crashing. The aim of this program is to simulate the launch (A very simplified one, nobody separation, a simple booster, and no rotations issue) and the recuperation of the module. To do that, trying to be as realistic as possible we had to adapt few things:

1. To be as close as possible from reality, we have a simple booster: you start it at the beginning and you cannot stop it, its weight is decreasing while you are using it, so it don't have any weight when it's empty, it's almost like a body separation.
2. The second engine can be started and stopped as much as you want but have a very small tank, and is not very powerful.
3. You can turn the rocket as much as you want without using any fuel (add another tank will not have a lot of interest) and to be easier to use, the rocket didn't rotate by itself.

$$m \frac{d\vec{V}}{dt} = \vec{u} + \vec{F}_{Earth \rightarrow Sat} + \vec{F}_{Moon \rightarrow Sat}$$

We can discretise the system:

$$\vec{V}(t+1) - \vec{V}(t) = \frac{1}{m} \vec{u}(t) + \frac{1}{m} \vec{F}_{Earth \rightarrow Sat}(t) + \frac{1}{m} \vec{F}_{Moon \rightarrow Sat}(t)$$

And considering this equation with the mass of the system which is almost always decreasing, you can update the system.

But to adjust the physic of the system, I had to adjust a lot of coefficients, like the power-to-weight and the consumption of the booster, or the weight of the different body, to make the system react like a real one. I adjust them one by one, from the weight to the consumption, to be able to launch rockets, without losing it.

The white line is a prediction of the system: If you continue like that, where you are the system going. It's very interesting to see how this simple three body system can evolve in such a complicate way.

The prediction part in the physical motor file can consume a lot of power, so you can, if needed you can decrease the size of the prediction array.

To improve this project, we should see if this discrete model is deriving from the theoretical one, and maybe increase the step. Also watch the coherence of the entire project, the gas engine is too powerful compared to the booster and real life, but here it's easier to have a powerful booster to test the program. I also implement a very simple collision program (5 points in a rectangle with a circle) and better algorithms could be implemented. From a physical point of view, the atmosphere is never taken in accompt, that's why it's almost impossible to land, so it can also be an idea to implement.

This project, available on github (14roiron/spacex) was made in almost 20 hours in 4 days.