Gravitational slingshot game

Daria Fedosova
ČVUT–FIT
fedosdar@fit.cvut.cz

January 8, 2023

1 Introduction

This report informs about the semester work created for the course BI-PYT in the winter semester B22. The aim of this work is to create a game working with the principles of gravitational slingshot. This game uses a simplified model of a gravitational slingshot maneuver, also known as gravity assist maneuver, or swing-by maneuver and designed for learning purpose. It is a two dimensional, real-time simulation game set in the playing field of a randomly generated star system. The main goal of the player is to launch the spaceship (Figure 1) that is affected by the gravity of planets and star. Since gravity assist can be used to increase or decrease speed or redirect spacecraft without significant use of limited resources (delta-v budget), which is useful in a real world, main purpose of this game is to visualize how this maneuver works in a simpler, more understandable environment...

2 Used model

Since spacecraft that utilizes gravitational slingshot maneuver in real life is affected by a large numbers of parameters, including galactic movement, Coriolis force, solar wind and weak force, the program uses a simplified model for spaceship and randomized star system. In fact, a simplified model can be used in real practice, for example, both the computer simulation of the Apollo flight and the flight itself was more primitive and included more allowable errors than many modern recreational computer games, which can be seen in the NASA Mission Report.[1] At its core, simplified model can be described by a couple of equations, which were used to create the game. Newton's law of gravitation:

$$g = -\frac{GMr}{r^3}$$

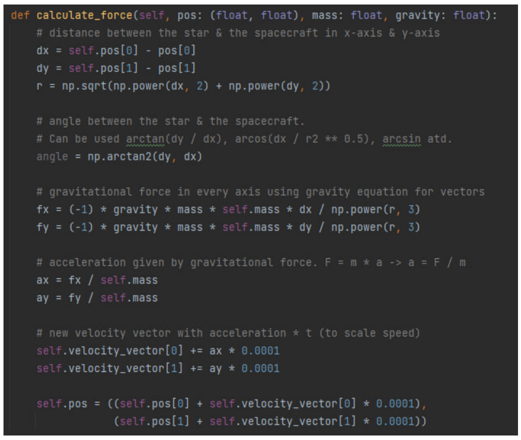
,where r is the position of the test point relative to the mass M. This equation is used in vector form, reflecting the fact that the gravitational field is a vector.[2] The distance formula for the distance between two coordinates on a 2D plane (Euclidean distance):

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

And acceleration:

$$a = \frac{v - v_0}{t} = \frac{\Delta v}{\Delta t}$$

As can be seen from the equations, simplified model needs size of objects, speed (angular), starting position and mass. All this values can be randomized.



```python
def calculate_force(self, pos: (float, float), mass: float, gravity: float):
    # distance between the star & the spacecraft in x-axis & y-axis
    dx = self.pos[0] - pos[0]
    dy = self.pos[1] - pos[1]
    r = np.sqrt(np.power(dx, 2) + np.power(dy, 2))

    # angle between the star & the spacecraft.
    # Can be used arctan(dy / dx), arcos(dx / r2 ** 0.5), arcsin atd.
    angle = np.arctan2(dy, dx)

    # gravitational force in every axis using gravity equation for vectors
    fx = (-1) * gravity * mass * self.mass * dx / np.power(r, 3)
    fy = (-1) * gravity * mass * self.mass * dy / np.power(r, 3)

    # acceleration given by gravitational force. F = m * a -> a = F / m
    ax = fx / self.mass
    ay = fy / self.mass

    # new velocity vector with acceleration * t (to scale speed)
    self.velocity_vector[0] += ax * 0.0001
    self.velocity_vector[1] += ay * 0.0001

    self.pos = ((self.pos[0] + self.velocity_vector[0] * 0.0001),
                (self.pos[1] + self.velocity_vector[1] * 0.0001))
```

Figure 1: Equations in program

In addition, the final version of the game uses a complete random generation of the star system, which can create both realistic and completely unreal system. While generator itself can be used to generate more "real" star systems in terms of speed of rotation or sizes, full randomization was chosen for greater clarity of the basic principle, as well as to increase the number of possible options. Also for fun profit. Additional reason to use this type of generation lies in the principle "show, don't tell". Player cant't see inner workings of a program, only the GUI, and can't know mass or gravitation of an object. That's why direct correlation between visual size of a displayed astronomical body and it's gravitational field was chosen. Other parameters that are unnecessary for clarity, but are used in calculations, do not have such correlation...

3 Results

The game starts with the main menu, where player can decide if he wants to generate randomized planetary system or use the real one. Then the program displays the gaming field with chosen option.

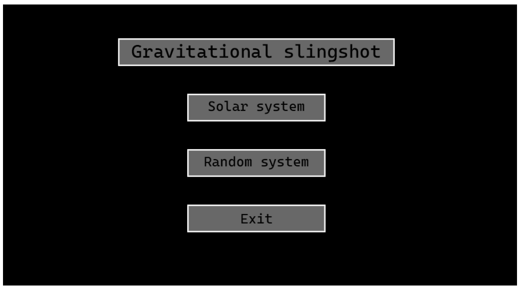


Figure 2: Main menu

Then the player can wait as long as he wants, then he can pause the gaming field and pick direction and velocity of the spacecraft using mouse click. The more distance from the staring point, the more will be velocity. It is possible to set 0 velocity by clicking on the velocity button again. There is also a pause button available. Or the player can return to main screen.

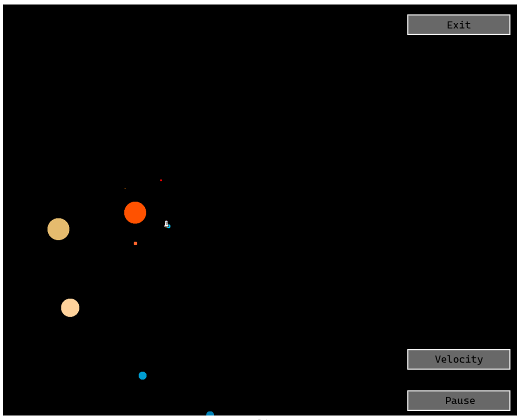


Figure 3: Game Field

Several test cases were conducted. Each test set was designed for each class of the game. Most of the test was directed at testing relatively realistic data, such as generated planetary systems with parameters from real solar system, taken from NASA[3]. Unfortunately, some of the planned tests couldn't be used because of the absence of data for comparison. All tests that were conducted returned positive results. . .

4 Conclusion

The most challenging part was to create even simple model of gravity assist, as it includes physics. Designing and programming of GUI was also a problem, because usually it requires a library, but UI was made without it. To be precise problem was in structure of the pygame library, which did not allowed to delete surfaces and clip surfaces to certain places and that required a work around. The chosen theme of the game has great potential, but requires understanding of quite difficult branch of the physics and a lot of time investment. Possible routes for improvement:

- Better or more "realistic" star system generation.
- Additional information for the player, for example, displayed mass of a body or difference between initial and resulting velocity.
- Easy mode with visible trajectories.
- Set of a randomized goals, for example, to get a ship with a certain speed to a certain point.
- Results recording system.
- New graphics.
- Touch GUI.

References

[1] NASA. Apollo 11 mission report. Manned Spacecraft Center, Houston, Texas, 1969. [p. 4-2, 4-3, 4-4] https://www.nasa.gov/specials/apollo50th/pdf/A11_MissionReport.pdf.

[2] NASA. Solar system exploration. online, 2023. https://solarsystem.nasa.gov/.

[3] David J. Raymond. Radically modern introductory physics. The New Mexico Tech Press. ISBN 9780983039457, 2015. [13.2: Gravitational Field] https://www.nasa.gov/specials/apollo50th/pdf/A11_MissionReport.pdf.