CSE 60166 – Mid Term Project Report (Fall 2014)

Sandipan Banerjee

Main idea: This project extends the 2D orrery of assignment 2 into 3D but with all the nine planets in the solar system included. The basic goal of the project was to draw the 3D orrery with actual textures being mapped to the 3D shapes for each planetary body. Some other interactive features have been included as well into this like changing the angular orientation of the orrery by dragging it with the mouse, zooming in and out, drawing and undrawing of orbits, saving frames as screenshots, increasing or decreasing the velocity of motion of the planets and playing explosion sounds as well. The effort to include an asteroid/rocket as an extra feature (as mentioned in the proposal) did not work out however.

Main functions implemented:-

1. Drawing of 3D orrery with the Sun, the nine planets and the moon.
2. Mapping actual textures taken from NASA images onto the planet bodies to get the real look to this orrery.
3. Realistic characteristics have been imparted on to the planetary bodies like radii, their velocity and their distance from the Sun is proportionate to their actual values. The Earth even has a tilt of 23.4 degrees like it actually does in reality.
4. Drawing of circular orbits around the Sun marking the trajectory of the planets.
5. Zooming in and out of the orrery.
6. Speeding up/down (with a bound) the velocity of the planetary motion frame wise.
7. Implemented a dragging function which lets the user drag the orrery up or down using his mouse. The left click button down marks the starting point and where it is let go marks

the ending point. The difference between the final and initial position decides the new angular orientation of the orrery.

8. Playing of an explosion sound clip when the user presses a particular key (although there's no sound in space!).
9. Taking screenshots of the current frame when the user wishes.
10. Resizing the actual orrery when the window is resized by the user.

Steps to run Orrery.py:-

1. Install 'Pygame' which can be found in the Sandipan_Banerjee_MidTermProject folder
2. Open cmd (command prompt) script
3. Go to the directory location (cd ../Sandipan_Banerjee_MidTermProject)
4. Type 'python Orrery3D.py' and press Enter
5. Press '+' to zoom in and '-' to zoom out
6. Press 'o' or 'O' to draw/undrawn the orbits
7. Click on a point on the screen and drag the mouse holding on and let it go at a new position. Based on the difference of these positions (up or down) the angular orientation of the 3D orrery changes
8. Press 's' or 'S' to save the current frame into a ppm file as a screenshot
9. Press 'v' or 'V' to play the sound of an explosion
10. Press 'w' or 'W' to increase the velocity of motion of the planets
11. Press 'x' or 'X' to decrease the velocity of the planets (but always > 0)
12. Press 'q' or 'Q' to close the window

Technical challenges I faced:-

1. The biggest challenge was to convert everything to 3D and calibrate the frustum because I hadn't worked on it before, so I spent many hours looking up the internet on how to do it. I did finally manage to do it.
2. The other big factor was to find the correct textures of the planetary bodies and map them properly to the system. Luckily I found a nice set of textures (although the Saturn ring is missing) and was able to setup the textures.
3. The changing of angular orientation of the orrery was quite easy to achieve after I learnt about the mouse pressing functions in Python. The playing of sound was easy as well.
4. The problem which I couldn't solve however was to start a linear rocket from the Earth. I was able to start a linear body from it but the trajectory calculation which I was doing failed and rendered the lines at wrong positions. There were too many transformations and stacks of matrices for me to get it right. I had to abandon it as a result.

Things I learned from the project:-

1. Rendering 3D shapes with GLSL shader using PyOpenGL
2. Mapping actual textures on the 3D shapes
3. Calibrating the frustum to get the optimum orientation
4. Mouse functions in python
5. Playing of sound clips in python
6. But most importantly, I learnt a whole lot on matrix transformations and how to use them in the right order when a lot of interactive objects are involved

Thanks for going through this report. Have a nice break!