3GC3 Fall 2023 - Assignment 1
Solar System Simulation

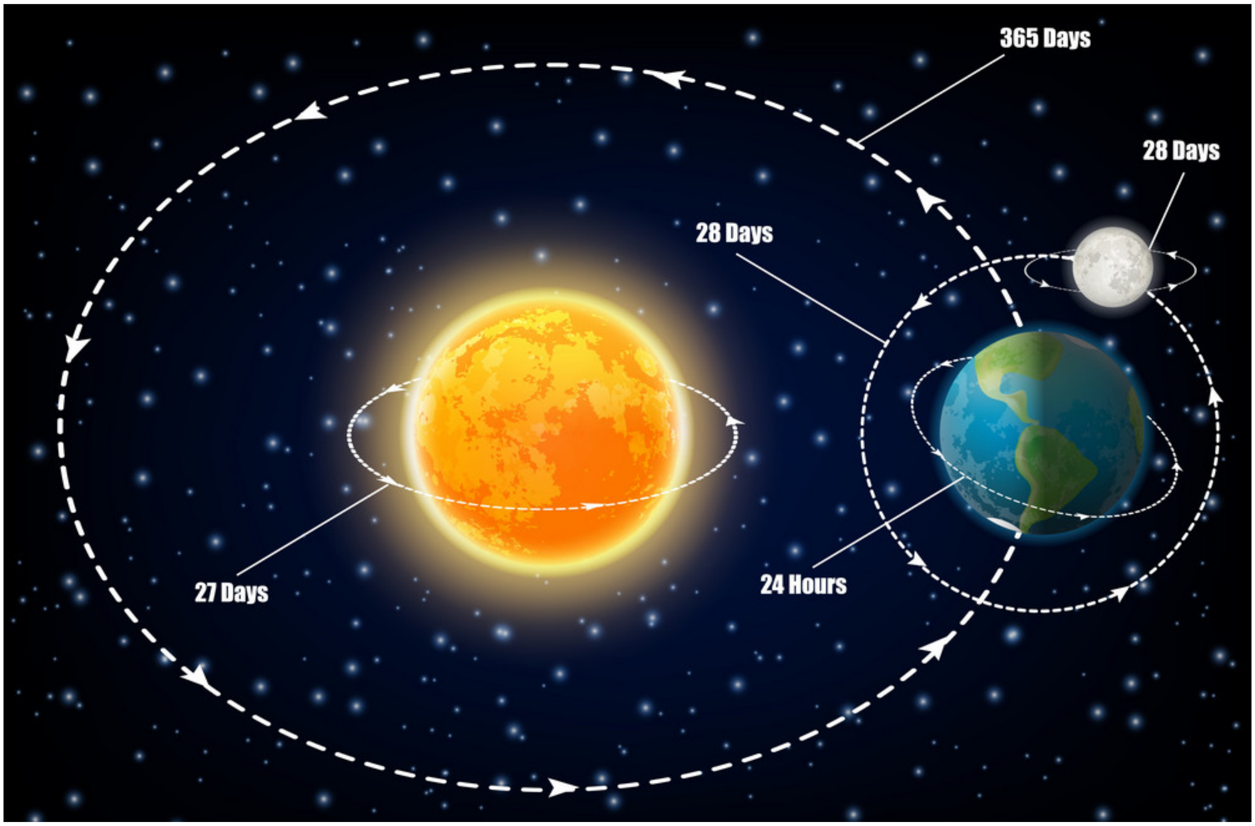


Figure 1. Solar System (Image from ID 145991838 © Bazuzzza | Dreamstime.com)

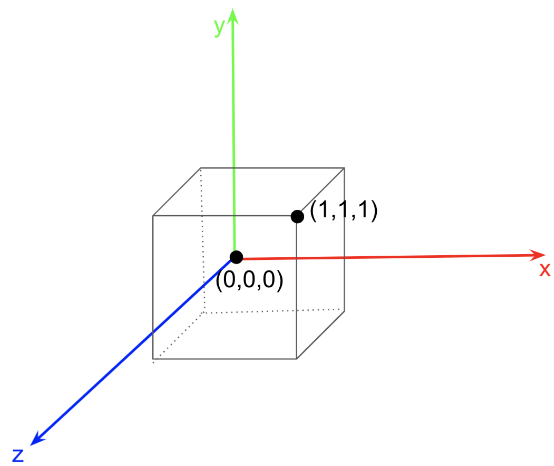
In assignment 1, we will be simulating the relationship between sun, earth and moon. We will proceed through several tasks.

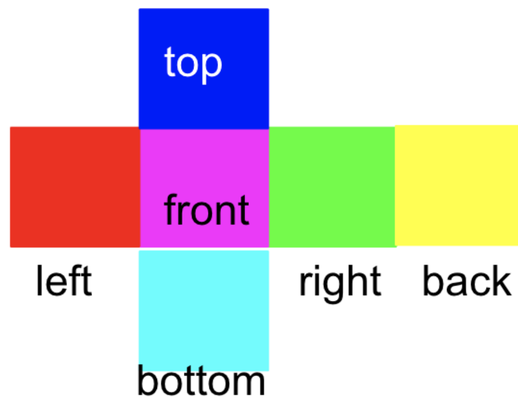
For the entire assignment tasks, make sure to turn on face culling to remove the backface. Make sure depth testing is turned on so that the front object blocks the back one, by using GL_LESS. No lighting is added explicitly.

Task 1: A Single Multi-Colored Cube

Draw a single multi-colored cube in the 3D scene. Please follow the detailed description below.

3D Scene:
A single 3D cube, centered at world space origin (0,0,0), edges range from [-1, 1] on the x, y, z axis. 6 faces of the cube have different colors.





Front face: z=1, color fuchsia rgb (1,0,1)
Back face: z=-1, color yellow rgb (1,1,0)
Left face: x=-1, color red rgb (1,0,0)
Right face: x=1, color green rgb (0,1,0)
Top face: y=1, color blue rgb(0,0,1)
Bottom face: y= -1, color aqua rgb (0,1,1)

Camera Projection:
perspective with 30 degree fov in height, 4/3 for aspect ratio, near at 0.1, far at 1000

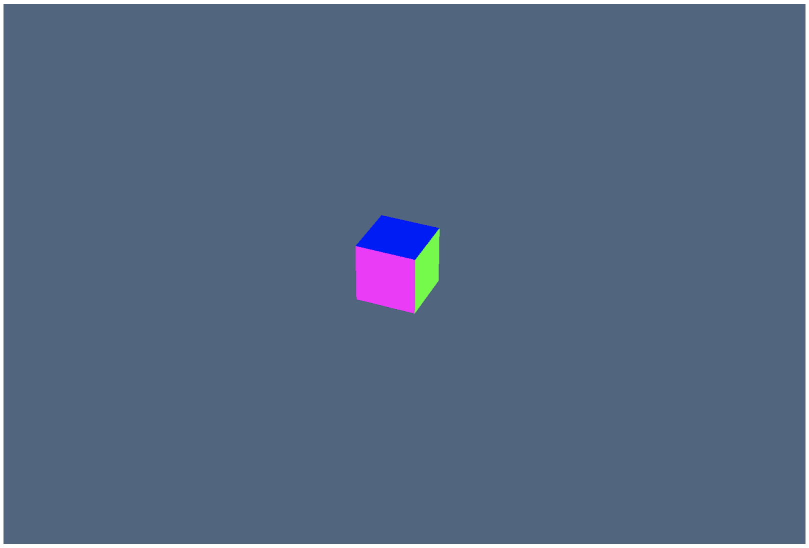
Window Size:
width 1024 x height 768

Background Color:
same as the assignment 0, i.e. rgb (0.3, 0.4, 0.5)

Camera Position and Screenshots:

1. Put the camera in world space location (-4, 5, 6), and look at the center of the cube. Capture a screenshot ppm by pressing key p. Rename the ppm file as “cube1.ppm”.
2. Put the camera in world space location (7, -8, -9), and look at the center of the cube. Capture a screenshot ppm by pressing key p. Rename the ppm file as “cube2.ppm”.

Example Result captured from a different view angle:



Task 2: Static Solar System

Draw the state of sun, earth and moon at the beginning of time i.e. day 0. Please follow the detailed description below.

3D Scene:

Use 3 cubes of different sizes to represent the Sun, earth and moon.

Represent the Sun with a cube of size 12 x 12 x 12 centered at world space (0, 0, 0), x,y,z range [-6, 6]. No self-revolving on day 0.

Represent the earth with a cube of size 6 x 6 x 6. Earth orbit is of radius 24. So the earth is located in world space (24, 0, 0) on day 0. Earth is tilted by 23.4 degrees illustrated below. No orbiting around the Sun, no self-revolving on day 0.

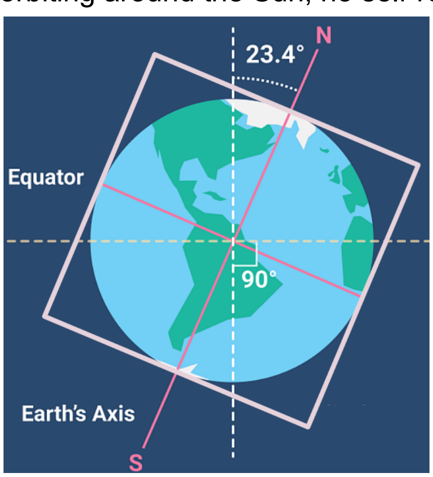


Figure 2. Earth tilting.

Representing the moon with a cube of size 3 x 3 x 3. Moon orbit is of radius 12. Note that the moon orbits around the earth. So the moon should be located in world space (36, 0, 0) on day 0. No orbit around earth, no self-revolving yet on day 0.

Day 0 state can be illustrated from Figure 3.

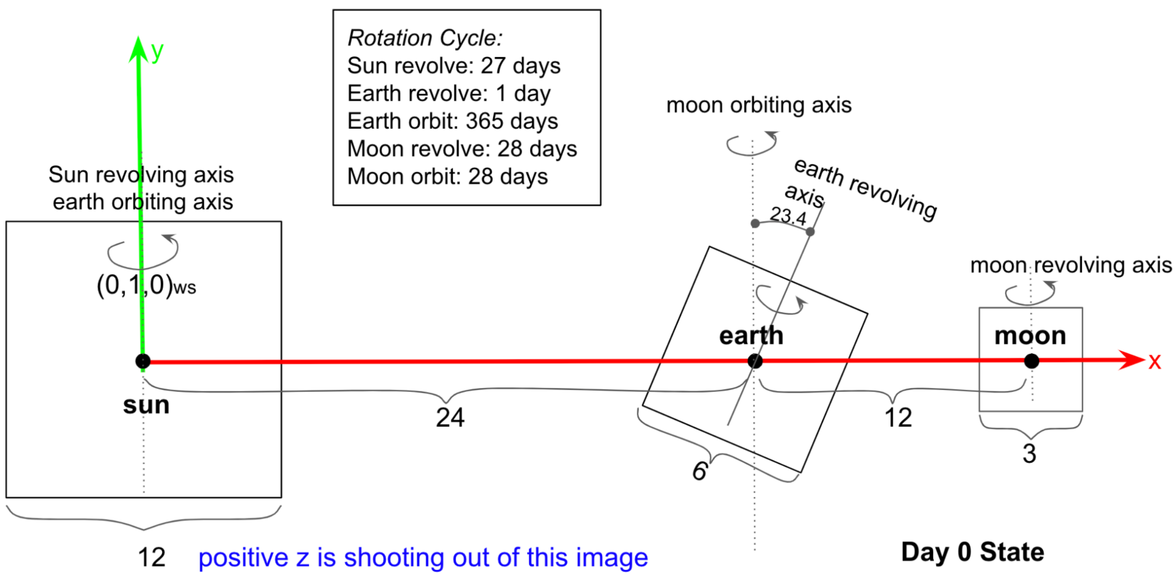


Figure 3. Simulation Configuration on Day 0.

Camera Projection:
perspective with 30 degree fov in height, 4/3 for aspect ratio, near at 0.1, far at 1000

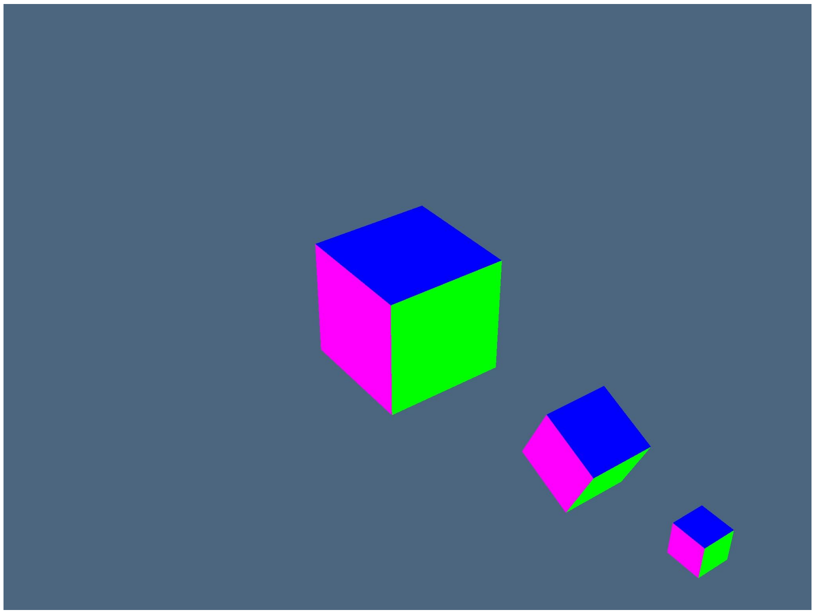
Window Size:
width 1024 x height 768

Background Color:
same as the assignment 0, i.e. rgb (0.3, 0.4, 0.5)

Camera Position and Screenshots:
1. Put the camera in world space location (0, 0, 150), and look at the center of the **Sun**. Capture a screenshot ppm by pressing key p. Rename the ppm file as “static_front.ppm”.
2. Put the camera in world space location (0, 120, 1), and look at the center of the **Sun**. Capture a screenshot ppm by pressing key p. Rename the ppm file as “static_top.ppm”.
3. Put the camera in world space location (100, 5, 5), and look at the center of the **Sun**. Capture a screenshot ppm by pressing key p. Rename the ppm file as “static_right.ppm”.
4. Put the camera in world space location (-70, 80, 90), and look at the center of the **Sun**. Capture a screenshot ppm by pressing key p. Rename the ppm file as “static_sun.ppm”.

5. Put the camera in world space location (70, 60, 50), and look at the center of the **earth**. Capture a screenshot ppm by pressing key p. Rename the ppm file as "static_earth.ppm".

Example Result captured from a different view angle:



Task 3: Rotate the Solar System

In this task, we will start orbiting and revolving the sun, earth and moon.

3.1 Rotation Angles

Implement the following functions in code, which pass in the day number, and return the rotation angle since day 0. Note: day is a floating number, 1/24.0f day means an hour. Rotation cycle is illustrated in Figure 3. For example, if the sun takes 27 days to revolve around itself, it rotates 360.0f/27 degrees per day, multiplying it with the day number, your function can compute the rotation angle since day 0.

```
//sun
float get_sun_rotate_angle_around_itself(float day);

//earth
float get_earth_rotate_angle_around_sun(float day);
float get_earth_rotate_angle_around_itself(float day);
```

```
//moon
float get_moon_rotate_angle_around_earth(float day);
float get_moon_rotate_angle_around_itself(float day);
```

3.2 Revolve and Orbit

Based on the revolving and orbiting angles computed in the previous step, given day number, rotate the sun, moon and earth accordingly.

The rotation axis is illustrated in Figure 3. Note that 1. revolving is around the object itself, 2. orbiting is relative. The earth orbits around the sun, the moon orbits around the earth, following common sense.

Camera Projection:

perspective with 30 degree fov in height, 4/3 for aspect ratio, near at 0.1, far at 1000

Window Size:

width 1024 x height 768

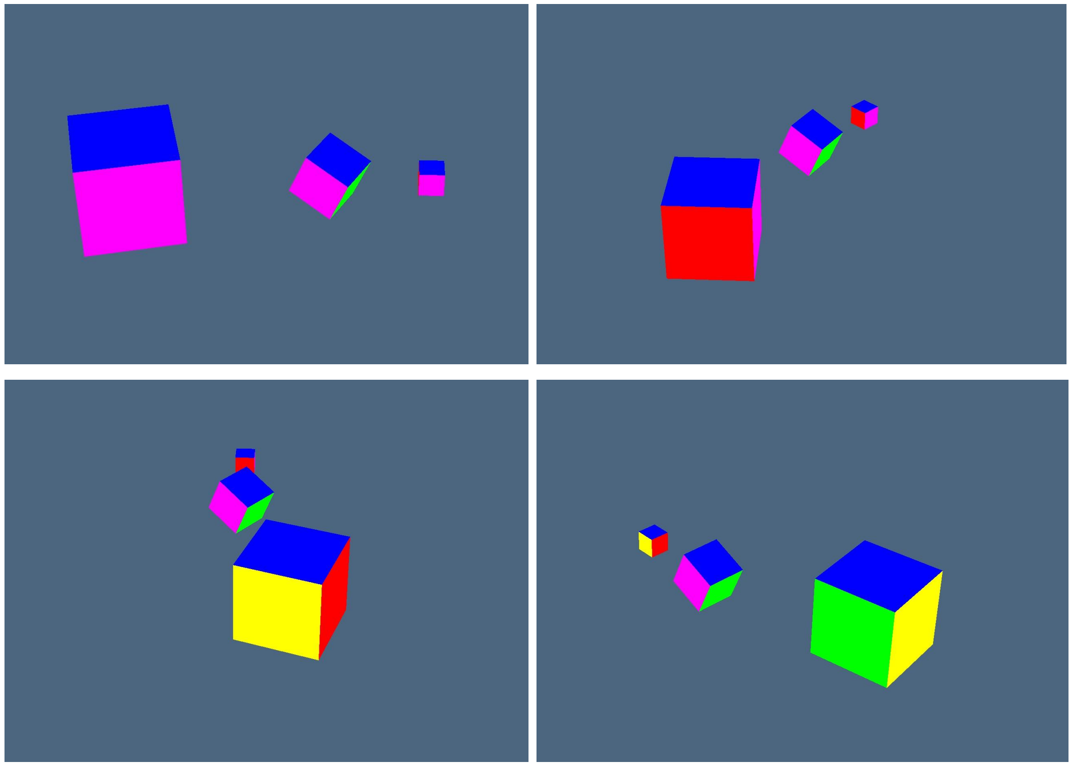
Background Color:

same as the assignment 0, i.e. rgb (0.3, 0.4, 0.5)

Camera Position and Screenshots:

1. Set variable day = 0, rotate your scene accordingly. Put the camera in world space location (80, 70, 60), and look at the center of the Sun. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_d0.ppm”
2. Set variable day = 1, rotate your scene accordingly. Put the camera in world space location (80, 70, 60), and look at the center of the Sun. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_d1.ppm”
3. Set variable day = 0.3, rotate your scene accordingly. Put the camera in world space location (80, 70, 60), and look at the center of the Sun. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_dpt3.ppm”
4. Set variable day = 128, rotate your scene accordingly. Put the camera in world space location (80, 70, 60), and look at the center of the Sun. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_d128.ppm”
5. Set variable day = 365, rotate your scene accordingly. Put the camera in world space location (80, 70, 60), and look at the center of the Sun. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_d365.ppm”
6. Set variable day = 365, rotate your scene accordingly. Put the camera in world space location (-50, -60, -70), and look at the center of the **earth**. Capture a screenshot ppm by pressing key p. Rename the ppm file as “rotate_d365_earth.ppm”

Example Results captured from a different view angle:



Note that one way to verify if your rotation is correct: check if the moon is "tidally locked", i.e. the moon rotates, but it rotates at the same speed that it orbits around earth, thus the moon keeps the same face (red) pointing towards the earth.

Task 4: Dynamic Solar System

To make a dynamic solar system simulation, in the main program, define a floating number `day` variable, and inside the while loop, add 1hr to the `day` variable and draw the solar system of that day.

```
float day = 0, inc = 1.0f/24;
while (!glfwWindowShouldClose(window))
{
    processInput(window);
    // draw solar system on that day
    day += inc;
```

```
        glfwSwapBuffers(window);
        glfwPollEvents();
    }
```

Run the code and you will see the simulated animation of the solar system.

Camera Projection:
perspective with 30 degree fov in height, 4/3 for aspect ratio, near at 0.1, far at 1000

Window Size:
width 1024 x height 768

Background Color:
same as the assignment 0, i.e. rgb (0.3, 0.4, 0.5)

Camera Position and Screenshots:
Put the camera in the world space location (100, 50, 100), and look at the center of the Sun. Use any screen capture tool to record a 30s long movie showing the animation. (You don't need to use ppm and key_p). Rename the movie file "dynamic_sun".

Put the camera in the world space location (100, 50, 100), and look at the center of the **earth**. Note that your camera stays at (50, 50, 100) all the time, while its gaze follows the earth center. Use any screen capture tool to record a 30s long movie showing the animation. Rename the movie file "dynamic_earth".

Put the camera in the world space location (100, 50, 100), and look at the center of the **moon**. Use any screen capture tool to record a 30s long movie showing the animation. Rename the movie file "dynamic_moon".

Note: when capturing videos, make sure it includes the whole window frame including the caption bar. If there is no window frame in your videos, you won't get full points.

Submission Checklist

Submit your **code** together with all the captured screenshots and movie files.

3 points - "cube1.ppm"
3 points - "cube2.ppm"

5 points - "static_front.ppm"
5 points - "static_top.ppm"
5 points - "static_right.ppm"
5 points - "static_sun.ppm"

5 points - “static_earth.ppm”

5 points - “rotate_d0.ppm”
5 points - “rotate_d1.ppm”
5 points - “rotate_dpt3.ppm”
5 points - “rotate_d128.ppm”
5 points - “rotate_d365.ppm”
5 points - “rotate_d365_earth.ppm”

5 points - “dynamic_sun”
7 points - “dynamic_earth”
7 points - “dynamic_moon”

20 points - source code

Please start early. Submission deadline is Oct. 28, 2023.