Project B: Watching the Little Robot Travel Around the Moon

CompSci 351-1: Intro to Computer Graphics ***Fall 2021***

Student Name: Jiaqi Guo NetID: JGR9647

● Project Description

1) ***Project Goal:*** In this project we are going to create a large, animated 3D 'world' that users view and explore with an interactive movable 3D camera. Two sets of GUI controls will be implemented to give the camera 3-dimensional perspectives and user-determined camera position. The HTML canvas object will be resized to fill the full width of the browser window and at exactly (70%) of its height. The 'canvas' object will show two camera images, one made with an orthographic projection matrix while the other one made with a perspective projection matrix. The last thing to mention is the creation of a mesh floor at the bottom of the canvas, and this "floor" plane will stretch out to the horizon in the x, y directions.

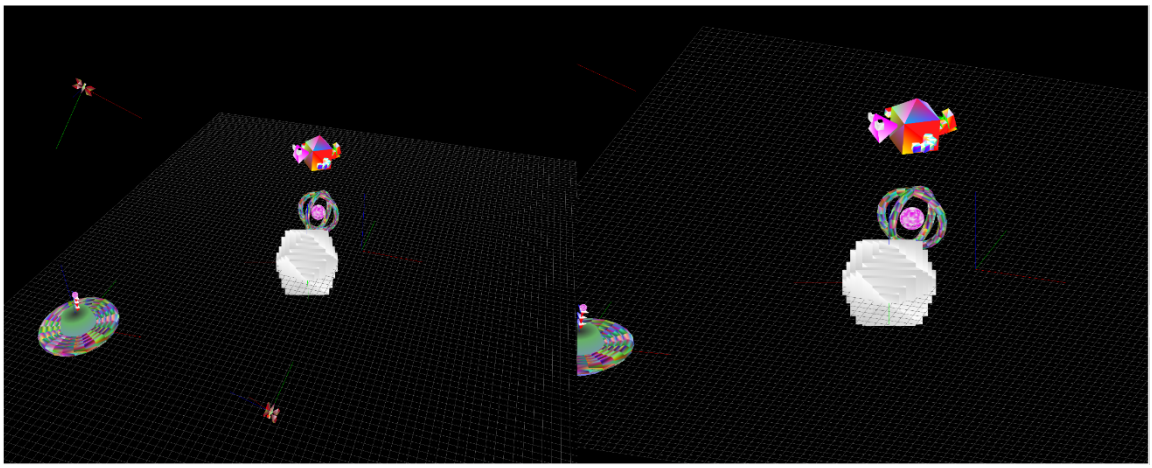


Figure 1 Chrome Screenshot for This Project (Two camera views side by side)

2) ***Project Detail:*** Our 3D world consists of 5 kinds of 3D rigid assemblies, which are the *planet in orbit* , the *moon*, the *spaceship, 2 flying butterflies* and *small robots*.

All 3D objects rotate in a specific user-determined-orbit around the moon to carry out periodic movements. The moon's position cannot be changed, but users can drag the moon (with your mouse) around at any camera positions and all perspectives.

● A Quick User Guide

- ***FUNCTION 1: Controlling the Speed of the Movement***

  a) ***What is it? :*** This function enables the user to *control the speed* of some of the moving 3D bodies when they are moving on the canvas.

  b) ***How to control? :*** Pressing the *"Spin <<"* to make those 3D rigid bodies *move clockwise*, press *"Spin >>"* to make those 3D rigid bodies *move counterclockwise*. Also, *stop/restart* the small robot/butterflies' movement by pressing the *"Run/Stop"* button.

- ***FUNCTION 2: Rotating the Moon with Your Mouse***

  a) What is it? : This function enables the user to rotate the moon.

  b) How to control? : Drag the moon in the middle of the canvas and you will find that the moon will rotates according to the mouse click-down and click-up positions.

- ***FUNCTION 3: Controlling the Cameras (Aiming angle and Position)***

  a) What is it? : This function enables the user to *control the position* and *the aiming angle* of both two cameras (ortho and perspective) with simply pressing the arrow keys (Figure 2) and "AWSD" keys on the keyboard.

  Camera position —— *using arrow keys*

  Aiming angle —— *using "AWSD" keys*

  b) How to control? : Pressing the ***left/right arrow keys*** will adjust the camera's compass-heading to the left/right while the ***up/down arrow keys*** will enable the camera to rotate up/down.

  Pressing the ***"A/D" keys*** will enable the camera to move leftwards/rightwards while the ***"W/S" keys*** will enable the camera to move forwards/backwards.

  c) Attention: Pressing "R" to reset the camera's aiming angel and position.

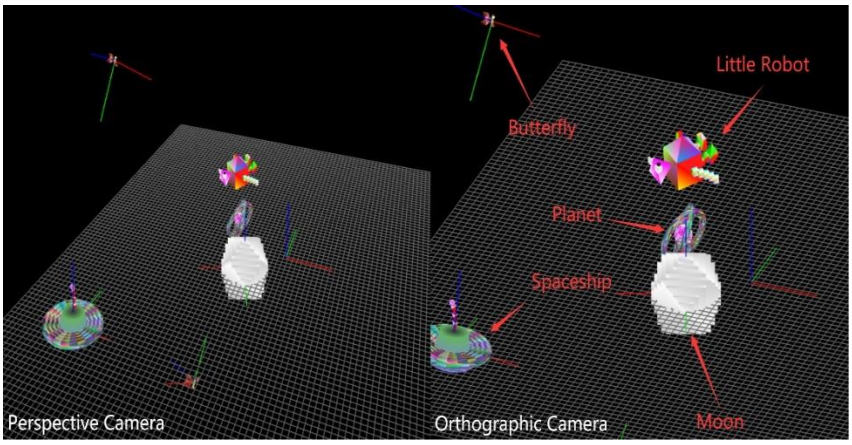
- ***FUNCTION 4: Entering the swing angle of the little robot using webpage control***

  c) What is it? : This function enables the user to control the angle range of rotation of the little robot arm

  d) How to control? : Entering a number between 0-45 in the box and press the submit button.

  e) Attention: The value entered by the user must be between 0 and 45, otherwise nothing will happen.

Result Explanation



(Left half: perspective lens Right half: Orthographic lens)

1. Example: 3D Assemblies and Ground-Plane Grid

The *little robot* is an animated, adjustable assembly made of several 3D parts. It has 4 flexing animated sequential joints.

The *butterfly, planet, spaceship*, and the *moon* are four additional, separate, joint assemblies, also made of rigid 3D parts.

At the bottom of the canvas, we can clearly see *a gray ground-plane grid*, which stretches out to the horizon in the x, y directions.

2. *Example: Change the Aiming Direction and Position of Camera (View Control)*

In this example, we will use the keyboard to control the position of our camera so that the user can observe the 3D world from *different aiming direction* and from *different positions.*

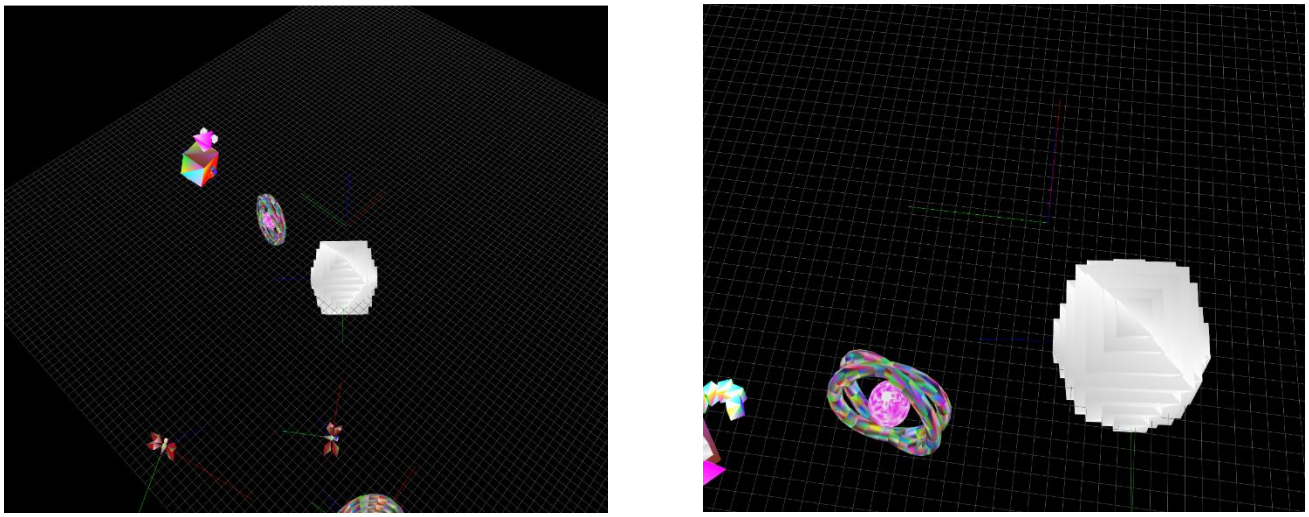


Figure 3,4 Move the Camera Closer and Observe the Moon from Different Aiming Angel (Perspective Camera)
Left: Far View Right: Close View

3. *Example: Re-sizable Webpage*

Our program depicts two side by side viewport that together fill all the width of the browser window

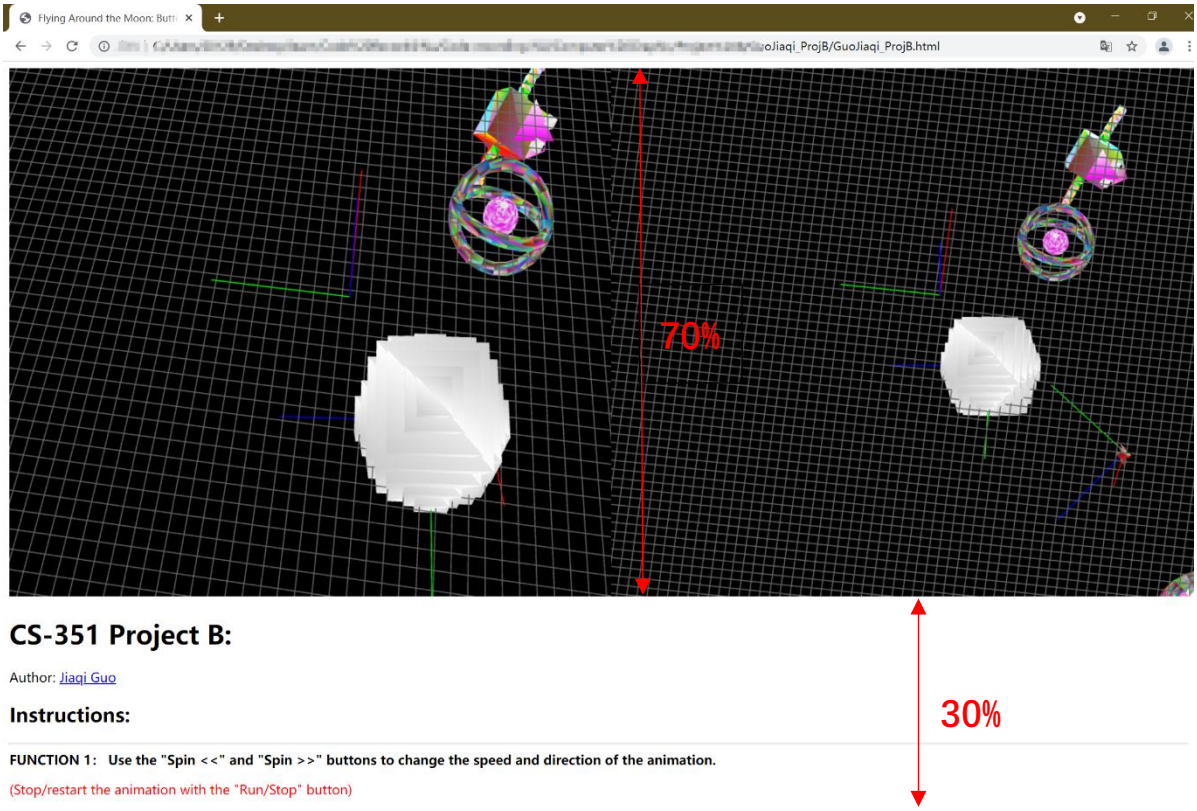


Figure 5 A screenshot of our application running in chrome

and 70% of its height. The left viewport shows image from a 3D perspective camera (its vertical FOV is fixed at 35°), the right shows the view from an orthographic camera.

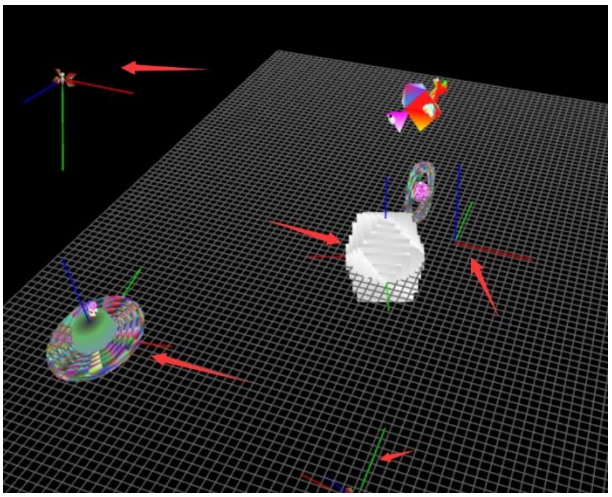


Figure 5 The world coordinate and object coordinates

4. *Example: Show 3D World Axes & Some 3D Model Axes*

One fixed world coordinate is located at the center of the ground-plane grid. In addition, there are four independent coordinate systems inside other four 3D objects.

For each coordinate, we have bright red for x axis, bright, green for y axis, bright blue for z axis.

● Scene Graphs for Two Main 3D Assemblies in This Project

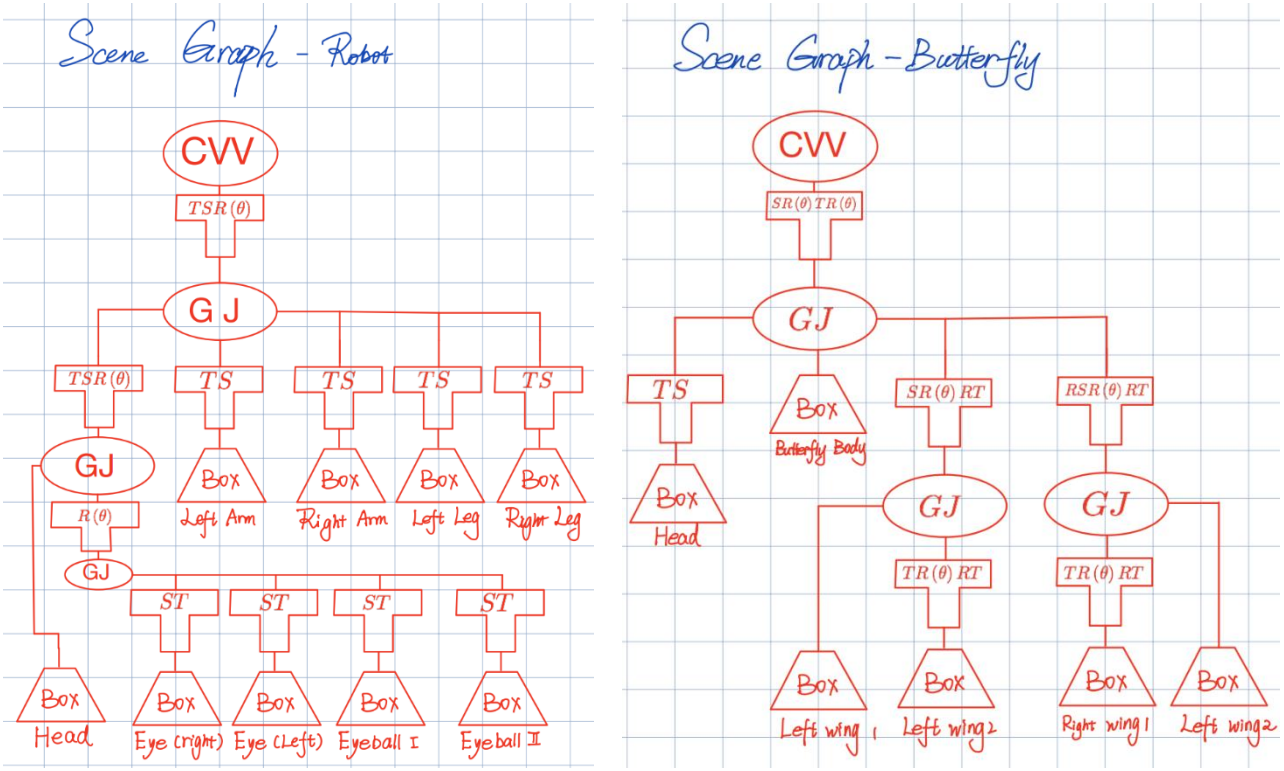


Figure 2 The Scene Graph for the 3D Assemblies (Little robot & Butterflies)