**Project A: A Little Robots Swimming in the Starry Night**

***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 construct a Moving, Jointed 3D Assemblies using WebGL. To be more specific, we have to build a collection of several different, visually distinctive rigid 3D parts. These 3D parts are then linked into several kinds of animated, jointed 3D assemblies through a series of 3D transformations (e.g. modelMatrix) and drawing calls. Besides, those moved assemblies should be able to move smoothly and continuously without users’ command, but also respond to users’ mouse & keyboard and GUI controls.
2. 背景图案

   描述已自动生成***Project Detail:*** The project consists of three 3D rigid assemblies, which are the *twinkling stars*, the *moon*, and *small robots* swimming in the night sky.

Among them, the small robot will follow the user-determined-orbit to carry out periodic movements. The moon's position cannot be changed, but users can drag the moon (with your mouse) around to find their favorite angle.

Finally, one of the highlights of the project is the constant twinkling of stars in the night sky. Each star is an independent 3D rigid part, and each star has its unique position and size as well as rotation sensitivity. Users can control the number of stars appearing on each frame of the canvas simply by entering a specific positive number **“N”** into the browser interface (take the above Figure 1 as example, the user’s input is **N = 4** ). For the *transient effect of the human eye*, users might see many stars flashing on the canvas at the same time, which is, in fact, larger than **“N”** that user defined.

Figure 1 Chrome Screenshot for This Project

* **A Quick User Guide**
* ***FUNCTION 1: Creating the Sparkling Stars***
  1. ***What is it? :*** This function enables the user to determine the number N of stars that appears on *each frame* of the canvas, and these stars of different sizes will be everywhere on your canvas. The larger the number you enter, the more stars will appear at the same time.
  2. ***How to control? :*** Just enter the number you want in the input box, and then press the *“Submit the number”* button.
  3. ***Attention:*** The number you entered should be a *non-negative integer*, and *do not enter 1 for your first input* or nothing will happen, since each reload of this program will reset this number to 1(initial states).
* ***FUNCTION 2: Rotating the Moon with Your Mouse***

1. ***What is it? :*** This function enables the user to rotate the moon and the star simultaneously.
2. ***How to control? :*** Drag the moon in the upper left corner of the canvas and you will find that both the moon and the stars rotate at different angles (see Figure 1 as example). Press the *“Reset the change to the moon”* button to reset the rotation angle.
3. ***Attention:*** The stars also rotate when you drag the mouse, but it may not be easy to observe because they are twinkling all the time.

* ***FUNCTION 3: Controlling the Movement of the Little Robot***

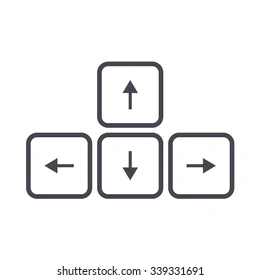
1. ***What is it? :*** This function enables the user to *control the position* of the small robots with simply pressing the arrow keys (Figure 2) on the keyboard.

Figure 2 The Arrow Keys

1. ***How to control? :*** Pressing the left arrow will make the small robot move to the left, and the right arrow will make the small robot move to the right. Similarly, the up and down arrows will control the small robot to move up and down respectively. And Pressing R on the keyboard will *reset* your little robot to its original orbit
2. ***Attention:*** If you accidentally move the robot out of the canvas, press R to reset the robot's position.

* ***FUNCTION 4: Controlling the Speed of the Movement***

1. ***What is it? :*** This function enables the user to *control the speed* of the little robot when it is swimming on the canvas.
2. ***How to control? :*** Pressing the *“Travel in directing <<”* to make the small robot *move clockwise*, press *“Travel in direction >>”* to make the small robot *move counterclockwise*. Also, *stop/restart* the small robot’s movement by pressing the *“Run/Stop”* button.

* **Result Explanation**

1. 图片包含 图形用户界面

   描述已自动生成***Example: Flashing Stars***

In this example, we will show the several features of my project, where we set the number of stars to 7 and click the “Submit the number” button.

Here we use a screenshot to capture a frame of the animated canvas, as shown in the Figure 3 where 7 stars appear on one screen, each with different sizes and different positions. That is because, here, we use a set of random numbers to determine the position and size of the star.

At the same time, we can also observe from the figure that each star has a different rotation angle. This is because we multiplied its change rate by a special change coefficient when calculating the angle, which changed the sensitivity of each star to mouse drag, making them present different angles on the canvas.

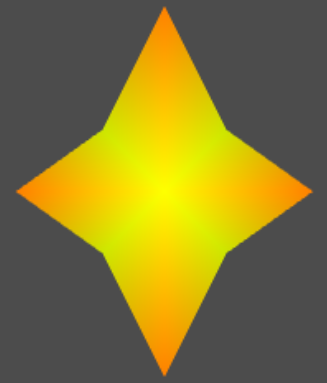


Figure 3 Screen Capture for Function 1 (input N = 7)

Figure 4 Basic 3D Star Model with its Various Transformation

Above, from left to right, are the basic 3D rigid shape of the star and its various transformations.

1. Example: Two 3D Assemblies