Starry Night

An interactive night sky simulator

Contributors: Anthony Lupica

The Goal

- Create a 3d night skyscape with webGl. (Uses external math library gl-matrix.js).
- Add interaction with event handlers
- Define the shape of a sphere with individual point vertexes
- Manipulate the camera view to the inside of the sphere volume
- Include realistic dynamic rotation, and variable star sizes.

The Implementation

- Two vertex array attributes: one to store point information (x, y, z) and another to store point size information for each point (float).
- Use two functions to return these arrays, called with the number of points to generate as an argument.
- Randomize the (x, y, z) coordinates to some float in [-0.5, 0.5) and normalize the created point so all are equidistant from the global origin, resulting in a sphere comprised of points all positioned somewhere on its surface area.
- Use similar logic in second function to find three random floats in [0.01, 1.01), and sum them together. The +.01 is to ensure that no point may have a size of zero. For every point generated on the spere's surface area, all will have a size between 0.03, and 3.00 units.
- Five total matrices are used for the purposes of model transformations, perspective projection, camera adjustment (on z-axis) , and as intermediate matrices for matrix multiplication.
- The camera is "moved," by a sort of equal and opposite translation. All world objects are instead moved accordingly to maintain the camera's position at the origin.

The Interaction

- Three event handlers
- mousemove event listener: Calls function “rotateSpeed()”. Event.movementX property tracks the cursor’s x position between mouse events, effectively tracking the cursors’s speed in the x direction. Use this to scale the rotation speed of the night sky according to how fast the user moves their cursor. Additionally, there is some further increase if the user’s cursor is moving near the edges of the screen.
- onkeyup event “handleReverse()”: checks for ‘r’ key press. If so, will flip the sign of a rotation direction variable (1 or -1) which is a multiplication that is applied in a rotation matrix calculation in the render loop.
- onkeydown event “handleZoom()”: checks for arrow key up/down press. Up zooms into the galax, while down zooms out (both to some defined limit).

The Difficulties/Possible Improvements

- Had some difficulty incorporating animation on the mouse move event that was smooth. Used a collection of if, else-if conditions to determine “speed brackets” for how much the sphere’s rotation should increase according to the cursor’s speed. Smoothness could likely be improved here
- Would have liked to include a look at cursor option, where rotation would match the direction of where the cursor was located on screen, so the sky is always “looking” at the cursor. Had to omit this aspiration.
- Constellations?