CS 371 – Fall 2020
Project 2

150 Points **Due: Friday, 11/06/20 at 11:59PM in Canvas**

To complete this assignment, you will need to create a 3D digital model of the image below, write code to render it, and write code to animate it.

Modeling and rendering: Consider the different views of the image above. You will need to model the image with a 3D mesh of triangles. One approach to creating the mesh would be to get some graph paper, draw the image and figure out a set of coordinates for the vertices and a set of edges for the triangles that would generate that image. You should color the image red with the back vertices being a slightly darker shade of red to make the 3D nature more visible. You may use any WebGL drawing primitive to render the image

Animation: You will need to write code to change the location of vertices over time to animate your model. Your code should use the following methods for changing the vertices:

1. Use affine transformations (such as scaling, rotation, or translation). These transformations should be applied to the vertices in the Vertex Shader. Your code should contain:
   - one or more translations
   - one or more scalings
   - multiple rotations about various axes so we can view the 3D nature of the shape
   - one or more twists

   The image should spin, move slowly over the viewport, change size and be deformed. This animation should be in a continuous loop so the image returns to its original position and state at some point, and then restarts the animation sequence. Use as much of the viewport real estate as possible but don't let the image disappear from the viewport.

2. Use the `lookAt` function to change the default position of the camera on startup. Your code should change the modelview matrix with this camera view and send it to the vertex shader. Recall that the model matrix is provided by the previous step and the view matrix is provided by this step. You should combine these matrices together to form the modelview matrix. These transformations should be applied to the vertices in the Vertex Shader.

3. Use either the orthographic or perspective projection matrices in your code too. You don't necessarily have to change the projection from the default that WebGL provides, but your code must create the projection matrix and send it to the vertex shader.

80% of your grade will be determined by the above requirements.

4. Your animation should also include another motion/effect - for example, maybe a few color changes in the animation sequence; or instead of lines, it draws with points; or the image explodes into little dots and then comes back together. It doesn't have to be complex, but it should be your own. Do something creative; 20% of your grade will be assigned here, with basic effects getting half of this 20%. To get the full 20%, your animation must have some creativity beyond the basics. If you are modifying the vertex positions by changing the coordinates in the buffer, make sure you use `gl.DYNAMIC_DRAW`.

Note:

- As usual, you should modularize your code using other methods.
- Your program must have a substantial introductory documentation block. By reading that I should learn the particular transformations, camera movements, etc. that you made to create this animation.
- Each method in your program should have an introductory documentation block that clearly and accurately describes its role in the program
- Make sure that your program's internal logic is self-documenting through the judicious use of meaningful variable names and indentation. Use underscores or a mix of upper and lower case to achieve this self-documenting style, e.g., `sweep_angle` or `sweepAngle`
- Please indent your code correctly

Submission

To complete this assignment, simply submit your HTML and Javascript file, zipped up in a single folder, to the Canvas submission folder **Project 2** by the deadline specified. The `Common` files should not be submitted; instead, expect them to be in a sibling directory to your unzipped folder.