CSE 470 Homework #2: Pick a Cube

Instructor: D. Hansford

Big Picture: Create a WebGL program that looks like the Figure below. Eight cubes are centered at equally-spaced positions on a circle and one cube is centered at the origin. The outer cubes rotate about their center and around the axis vector defined by their respective center and the origin. Four user interactions are provided.

1. The user can *click on the canvas* to select an outer cube. If the click is close enough to the cube, then the center cube will begin to rotate in the same way that the chosen cube is rotating.
2. The *reset center cube button* resets the center cube to not rotate.
3. The *scale cube slider* will scale the cube from 10% its original size to 100% its original size.
4. The *rotation speed slider* will change the rotation speed from 0 to 2 radians per redraw.

Detailed specifications are provided below.

Concepts:

- 3D geometry with z-buffer hidden surface removal
- 3D clip volume
- Model instance
- Modelview matrix (rotation about a fixed point) in the vertex shader
- User interaction (click in window, button, slider)
- Geometry: circle, distance, representation of 3D object as triangle mesh
- JS graphics programming / program organization

Details:

Cube model (prototype) definition:

- You *must* use the cube definition in the file CreateCubeHW2.js.
- Do *not* modify the vertices in this file.
- Change the code in this file so that each face is assigned a unique color. Your choice! Do not use the background color for a cube face.
- Cube vertices and colors must be sent to the GPU. (However, this is not done in this file.)
- Tip: Look closely at the definition of the cube. Make a sketch! You need to know where the model lives in the coordinate system in order to set up the transformations.

Cube Instances

- The nine cubes displayed are instances of the model or prototype.
- Position the outer cubes on a circle of radius 0.75
- Each instance has the following properties/attributes
    - location (x,y,z) where (x,y) is determined by the circle definition and z=0
    - rotation axis defined as location – origin
- All instances share the same
    - (current) rotation angle; Initialize to 0.0
    - speed of rotation (increment of current rotation angle); Initialize to 0.2

- scale (of the model); Initialize to 0.2 (referred to as 20% on the slider)

HTML page and Canvas definition:

- Create a delineated canvas so the user knows where to click.
  (This can be a border or an off-white color.)
- In HTML text, communicate your name, date, description of the project, and resources used.
  (Please be sure this appears in the HTML document for the user to see.)
- Next to the canvas, instruct the user to click to pick a cube and (briefly) why/how.
- Include a button for resetting the center cube to not rotate.
- Include a scale cube slider and a rotation speed slider.
- Each user interface should include text to guide the user.
- See the project demo figure for a guide.

User Interaction and Functionality:

- Define a variable that defines the radius of a circle that when centered at each cube in the (x,y)-plane, it covers much or all of the cube. This can be as simple as the scale variable for the cubes since the model cube is length 2. Let's call this radius "r".
- When the user clicks in the canvas, find the closest cube and check if the distance to this cube is within "r" distance to the cube. If the click is close to a cube, then the center cube will begin to rotate in the same way that the chosen cube is rotating. If the click is not close to a cube, then no change is made.
- The cube scale slider changes the scale from 10% to 100% of the model cube size.
- Use the MV.js "rotation" function to rotate each cube about a given axis.
  (This function will normalize the rotation axis vector.)
- Set up a modelView matrix for the transformations. The matrix definition must be in render().
  Pass the definition via a uniform to the vertex shader.
  (Do *not* create the matrices in the vertex shader as done in rotatingCube.)

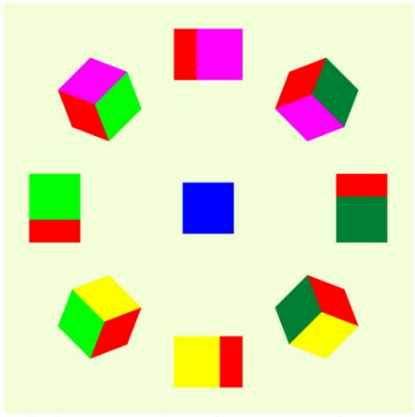
Tips:

- The class demo programs provide good examples for implementing the functionality. The rotatingCube program is the best template for getting started, but remember that the vertex shader for this HW will be different.
- See w3Schools for good JS tutorials.
- Get help early! Don't hesitate to email the instructor or TA for guidance.
- Make a plan. Take baby steps. Print out everything (console.log).
- Comment your code. This will help you think about what you are programming and you will have a better chance for partial credit if some functionality does not work 100%.
- Come to class for more tips.

Turn-in Guidelines

1. Name your html and main js file pickAcube.html and pickAcube.js, respectively. You may create more files.
2. Add your name at the top of each file you created.
3. The file/Common folder structure must be just as the class demo programs are organized.

4. Zip all your files into LastnameFirstInitial_HW2.zip.
   Do *not* include the Common folder.
5. Turn in your assignment on Canvas.



Make the center cube replicate motion of an outer cube.
Click canvas to pick a cube.

reset center cube

scale cube 10% 100%

rotation speed 0 2

CSE 470 HW#2: Pick a Cube

Author: Dianne Hansford **Date:**
Description: This program .... *complete this section*
Functionality:
complete this section
Parameters:
complete this section
Resources: Prof. Angel's codes. *complete this section*