Stained Glass

In this assignment you will be drawing an interactive stained glass window with HTML and WebGL/Three.js. The window will have a minimum number of rectangles, triangles, circles, and other shapes (see below). The user should be able to adjust the alpha value where the transparency of each shape inside the border will be increased or decreased. Follow the directions to implement your window.

PHASE I

This phase consists of designing your window. You do not need to turn anything in for this phase. The window should be 800 x 600 and include a wooden (brown) frame, a solid black interior, at least five rectangles, at least five triangles, at least two circles, and at least four shapes drawn with a curve (such as **quadraticCurveTo**). The shapes should vary in color, size, and location. Use a brown background and .holes to create the inside of the window. You must use **Shape/ShapeGeometry** (along with **.moveTo**, **.lineTo**) to create all of the shapes except for the circles which may be drawn with **CircleGeometry**. Your drawing must be symmetrical along either the vertical or horizontal axis. The most time consuming part of this assignment will be deciding on the coordinates for each shape vertex. I suggest that you manually draw your design on plain paper or graph paper and label each vertex with its coordinates.

PHASE 2

Your program will consist of two files – **stainedGlass.html** and **stainedGlass.js**. Use a slider to give the user the ability to decrease (or increase) the source alpha for more (or less) transparency. When the user increases (or decreases) the transparency, subtract (or add) a fixed amount from the alpha for all shapes inside of the frame by looping through all of the opacity values (except for the border) and then redrawing the scene. Both the slider and the opacity of each object should have an initial value of **0.5**. The far left value of the slider should give you a blank window with only the border showing. The far right value of the slider should result in objects with no transparency. (You are free to experiment with other blend functions, alpha values, and background colors.) The requirements for each file are listed below.

stainedGlass.html

Creates a 800 x 600 window. The window should have the following components:

```
body{
        background-color: #000000;
        overflow: hidden;
        margin: 0px 0px 0px 0px;
}
```

```
#slider1{

        position:  relative;

        left: 0px;

        top:  0px;

}
```

The slider should have a range from 0.0 to 1.0, increment/decrement in steps of 0.1, and be initialized to 0.5.

stainedGlass.js

Contains the at least the following functions and any more that you think are necessary or helpful:

init
Sets the window color, the canvas width and height, create a new scene and orghographic camera, and add an event handler that listens for the alpha change. Make the canvas width and the canvas height 50 units less than the .innerWidth and .innerHeight to make room for the slider. Create the camera with the following:

```
viewLength = 1000;
aspRat = canvasWidth/canvasHeight;
camera = new THREE.OrthographicCamera(-aspRat*viewLength/2,
                                       aspRat*viewLength/2,
                                      viewLength/2,
                                     -viewLength/2,
                                     -1000, 1000);
```

draw
Calls the following functions and adds all created shapes to the scene. Do not add shapes to the scene in the following functions.

drawBorder
Draws the border with .holes.

drawAllRects
Draws and stores all rectangles in the scene. Calls any helper functions that you think may be helpful.

drawAllTris
Draws and stores all triangles in the scene. Calls any helper functions that you think may be helpful.

drawAllCircles
Draws and stores all circles in the scene. Calls any helper functions that you think may be helpful.

drawAllCurvedShapes
Draws and stores all shapes created with a curve. Calls any helper functions that you think may be helpful.

drawOtherShapes (optional)
Draws and stores all shapes that can be categorized by one of the above functions. Calls any helper functions that you think may be helpful.

renderScene
Renders the scene.

DELIVERABLES

Compiling and Executing: Your program should execute with Google Chrome or Mozilla Firefox. Place both of your files in the same directory level. To grade your project, I will simply open **stainedGlass.html** with Chrome or Firefox.

Format: Your code must adhere to accepted program conventions regarding line length, documentation, spacing, etc. Style and comment your code as if it were a .java file. In the header comment for each file state whether you want the project to be executed in Chrome or Firefox.

Teams: You may work in teams of one or two students. If you work in teams of two, make sure that the team members are listed in **stainedGlass.js**'s comment section. People in different groups may talk about the concepts in the assignment but may not share code.

Submission: Compress your files together and turn in the single file to the assignment link in Blackboard by midnight on **February 20**. You may turn in your assignment up to a week late. Five points will be deducted for everyday late including holidays and weekends.

GRADING

Programs that do not compile will receive a score of 0. You will be graded according to the following:

Program execution (Minimum program functionality as described above)	**75%**
Quality (Quality of the rendered image including aesthetic appearance and creativity)	**15%**
Coding Style/Documentation	**10%**