CSS450: Assignment 4 Question Sheet [This assignment carries twice the weight]

Instructions:

- In questions where you are asked to explain, please be concise.
- Show your work when necessary, be neat, precise, and brief!
- To help us grade your assignments and return this to you in a timely fashion please:
    - Put your name and answers in the answer sheet only (separate link provided). Anything you write outside of the answer sheet **will not** be graded.
    - Provide your answers in the order of the problems.
    - Please use only one side of 8.5x11 paper.
    - Please make sure you bring a hardcopy print out **of the answer sheet (!!only!!)** to submit at the beginning of class. Please do not print out the problems.

Your assignment may not be graded if any of the above is violated, you have been warned.

1. In Week3 class example 3.9 (http://courses.washington.edu/css450/2016.Fall/WeeklyExamples/Week3/3.9.Timer+SimpleSimulation/public_html/index.html), the XForm GUI above the canvas reports the current values in the Transform of the rectangle shape that we drag out. However, notice that when the shape falls, the Translation X/Y slider bars do not reflect the position of the shape. Remember that you need to click on the "T" radio button to observe the translation state of the rectangle shape.

   In the context of MVC architecture, this problem can be described as the __(a) *which of the Model/View/Controller*__ component has changed state and yet the __(b) *which of the Model/View/Controller*__ component is not aware of the changes resulting in the state of the two components being out of sync. In order to fix this problem, we can insert this code: __(c) *the code you would insert*__ into this function: __(d) *name the function*__ in this file: (e) *name the file*.

2. Recall that we saw the rendering of mesh objects during Thursday's class. The following *ClassExample* defines the code that is capable of creating a Mesh object, and drawing it. The *createCustomMesh()* function creates the *mMesh* object that can be drawn by WebGL, and the *draw()* function draws it. This code will not run because we do not have the library support for ***OBJ***, and we do not know what is the *meshInString* parameter passed to *createCustomMesh()*. However, we can see the integration of this code into the code base that we are familiar with, in particular, we see the code working with classes that we are familiar with: *gEngine*, *SimpleShader*, *Transform*, and *Camera*.

```
ClassExample.prototype.createCustomMesh = function (meshInString) {
    this.mMesh = new OBJ.Mesh(meshInString);      // 1. Creates a Mesh Object
    OBJ.initMeshBuffers(gEngine.Core.getGL(),
          this.mMesh);                           // 2. Initialize WebGL Buffers
    this.mConstColorShader = new SimpleShader(    //
        "src/GLSLShaders/SimpleVS.glsl",         // 3. Shader for the Mesh
        "src/GLSLShaders/SimpleFS.glsl");        //
    this.mXf = new Transform();                  // 4. Transform Object for the Mesh
    this.mXf.setXPos(200);                       // 5. Sets the X-size of the object
    this.mXf.setYPos(200);                       // 6. Sets the y-size of the object
    this.mXf.setSize(30, 30);                    // 7. Sets the size of the mesh
};

ClassExample.prototype.draw = function (camera) {
    gEngine.Core.clearCanvas([0.9, 0.9, 0.9, 1]); // 8. Clear canvas
    camera.setupViewProjection();                // 9. Camera Magic
    if (this.mMesh !== null) {
        var gl = gEngine.Core.getGL();           //10. Gets a reference to WebGL
        this.mConstColorShader.activateShader(    //11. Activates the shader
            this.mMesh.vertexBuffer,
            [1, 0, 0, 1],
            camera.getVPMatrix());
        this.mConstColorShader.loadObjectTransform(this.mXf.getXform());
                                                 //12. sets the mesh transform

        gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER,    //13. Next two lines draws
                this.mMesh.indexBuffer);
        gl.drawElements(gl.TRIANGLES,
                this.mMesh.indexBuffer.numItems, gl.UNSIGNED_SHORT, 0);
    }
};
```

Staring at the above code and comparing to the *Renderable* and *SquareRenderable* classes from Example 4.3 (http://courses.washington.edu/css450/2016.Fall/WeeklyExamples/Week4/4.3.ApproximateCollision/public_html/index.html), you begin to realize that although you don't know all the details, you do actually have an amazingly in-depth knowledge. For example:

a. You know there must be a WebGL buffer that contains all the vertices for the mesh. What is the full name of the variable that refers to this WebGL buffer?
   Please list the full-scope name of the variable, beginning with ***this***, your answer will look like: *this.something.something….*.

b. What would be the color of the drawn mesh object?

Staring at the code a little more, you realize that you are literally seeing a *MeshRenderable* class. The *MeshRenderable* can simply subclass from *Renderable* with *createCustomMesh()* function being the constructor and the *draw()* function overriding the definition in the *Renderable* class. Of course, you recognize that some modifications must be made. For example, line-3 of the given code, the line that says: `// 3. Shader for the Mesh`, you know that this line is unnecessary in a subclass of *Renderable* class because the shader is already defined in the super class. You also recognize that lines 5, 6, and 7 are setting the transform for the object and do not belong in a constructor. Now, examine the code closely and identify all other lines of code that do not belong in the corresponding functions:

c. Please list the rest of the code in *createCustomMesh()* that do not belong in the constructor of a *MeshRenderable* class. *Your answer will be a comma-separated list of integers.*

d. Refer to the given *draw()* function. Please list all of the lines that do not belong in a *MeshRenderable::draw()* function. *Your answer will be a comma-separated list of integers.*

See? With proper conceptual framework, you can perform abstraction based on simple pattern matching! ☺