CPSC 314
Assignment 1: Introduction to Three.js, WebGL, and Vertex Shaders

Due 4PM, Jan 23, 2015

1 Introduction

The main goals of this assignment are to setup your graphics development environment, including checking your browser compatibility, enabling the use of local files, and an initial exploration of the uses of vertex shaders. For this exploration you will be using a template provided by the instructor, including shader code (`.glsl` files).
Your main work will be to develop a high level understanding of how the code works, to modify or write vertex shaders, and to use rudimentary communication between the JavaScript program and the shaders. Some of the details of what is going on in the rest of the code will only become clear a bit later in the course. You are of course welcome to take a peek now, especially for the last part of the assignment. Some of the concepts are explained in Appendix A of your textbook, and in the web resources listed on the course web page.
To program a shader, you will use a programming language called GLSL (OpenGL ES Shading Language version 1.0). Note that there are several versions of GLSL, with more advanced features, available in regular OpenGL. Make sure that any code you find while trying to learn GLSL is the correct version.
This assignment uses a simple scene with an "Armadillo" character, interacting with a "gem". You can move the camera around the scene by dragging with a mouse.

1.1 Template

- The file `A1.html` is the launcher of the assignment. Open it in your preferred browser to run the assigment, to get started.
- The file `A1.js` contains the JavaScript code used to set up the scene and the rendering environment. You will need to make minor changes in it to answer the questions.
- The folder `glsl` contains the vertex and fragment shaders for the armadillo and gem geometry. This is where you will do most of your coding.

- The folder `js` contains the required JavaScript libraries. You do not need to change anything here.
- The folder `obj` contains the geometric models loaded in the scene.
- The folder `images` contains the texture images used.

1.2 Execution

As mentioned above, the assignment can be run by opening the file `A1.html` in any modern browser. However, most browsers will prevent pages from accessing local files on your computer. If you simply open `A1.html`, you may get a black screen and an error message on the console similar to this:

```
XMLHttpRequest cannot load armadillo.vs.glsl. Cross origin
   requests are only supported for protocol schemes: http,
   data, https.
```

Please see this web page for options on how to run things locally:

`https://github.com/mrdoob/three.js/wiki/How-to-run-things-locally`

2 Work to be done (100 pts)

First, ensure that you can run the template code in your browser. See the instructions above. Study the template to get a sense of how it works.

1. (70 points)

   (a) **15 pts** Gem modification.

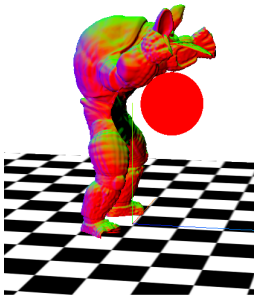
   The variables `gemPosition` (the position of the gem center in world coordinates) and `gemRadius` declared in `A1.js` are changed using the keyboard, and passed to the gem shader (in `gem.vs.glsl`) using `uniform` variables. Modify the gem shader to move the gem and change its size in response to keyboard input. Important: **do not** use Three.js functions; you must modify the shader for credit.

   (b) **20 pts** Color by Location.

   In the template, `armadillo.vs.glsl` and `armadillo.fs.glsl` use the normal of the current vertex to set the surface color. Modify them to use the position of the current vertex. That is to say, color code (x,y,z) coordinates using (r,g,b) colors in some way that is obvious to see.

   (c) **35 pts** Proximity Deformation.

   For this part you will need to change `armadillo.vs.glsl` and `A1.js`. The armadillo mesh should deform to the surface of the gem, as illustrated in the figure below. One simple way is to check if a vertex is within the gem, and if it is, move



the vertex to the surface. You should pass the necessary information about the gem to the armadillo shader.

Hint 1: See how uniforms are passed on the gem shader. *Hint 2:* It may be easier to see the deformation if you reduce the displayed size of the gem sphere. This was done in the illustration. *Hint 3:* You may use one of the predefined transformation matrices, listed below.

```
uniform mat4 modelMatrix;
uniform mat4 modelViewMatrix;
uniform mat4 projectionMatrix;
uniform mat4 viewMatrix;
uniform mat3 normalMatrix;
```

IMPORTANT ADDENDUM as mentioned in Lecture 6 (along with demo). Instead of using these matrices, you can make the following changes in A1.js, for part 1(c).

Change

```
loadOBJ('obj/armadillo.obj', armadilloMaterial, 3, 0,3,0, 0,Math.PI,0);
```

To

```
loadOBJ('obj/armadillo.obj', armadilloMaterial, 1, 0,0,0, 0,0,0);
```

Change

```
var gemPosition = {type: 'v3', value: new THREE.Vector3(0,5,3)};
```

To

```
var gemPosition = {type: 'v3', value: new THREE.Vector3(0,1,1)};
```

After these changes, you do not have to construct a matrix or use the above matrices for this assignment.

2. **Part 2:** (30 pts) Creative License

   For this part we want to see what you can do. Your ideas should use at least one new shader, and should be of a similar complexity to the previous tasks. If you have any doubts, make sure to OK it with a prof or TA. Some possible suggestions might be:

- deform the vertices in your object in a wave over time.
- explode the model along face normals to view all the triangles that make it up.

Bonus marks may be given at the discretion of the marker for particularly noteworthy explorations.

2.1 Hand-in Instructions

You do not have to hand in any printed code. Create a README.txt file that includes your name, student number, and login ID, and any information you would like to pass on to the marker. Create a folder called "a1" under your "cs314" directory. Within this directory have two subdirectories named "part1," and "part2", and put all the source files, your makefile, and your README.txt file for each part in the respective folder. Do not use further sub-directories. The assignment should be handed in with the exact command:

```
handin cs314 a1
```