

CPSC 314

Assignment 1: A Jumping Bunny. Introduction to Three.js, WebGL 2.0, and Shaders

Due 11:59PM, Jan 17, 2019

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 and fragment shaders. For this exploration you will be using a template provided by the instructor, including shader code (`.glsl` files in the `A1.html` file). Your main work will be to develop a high level understanding of how the code works, to modify or write 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 3.0). 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 consisting of an “bunny” character and a single white egg. Your task will be to make this bunny run, jump, and lay colour changing eggs. You can move the camera around the scene by dragging with a mouse, pan by holding down the right mouse button while dragging, and zoom by scrolling the mouse wheel.

1.1 Getting the Code

Assignment code is hosted on UBC STASH BitBucket, to retrieve it onto your local machine first ensure that you have logged in at least once onto BitBucket:

```
https://stash.ugrad.cs.ubc.ca:8443.
```

Then navigate to the folder on your machine where you intend to keep your assignment code, and run the following command from the terminal or command line:

```
git clone https://stash.ugrad.cs.ubc.ca:8443/scm/cpsc314-2018wt2_students/a1.git
```

1.2 Template

- The file `A1.html` is the launcher of the assignment. Open it in your preferred browser to run the assignment, 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 bunny and egg 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.

1.3 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:

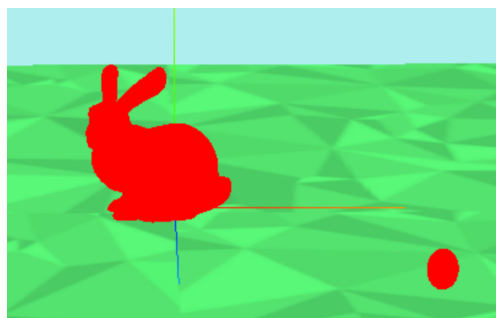
```
XMLHttpRequest cannot load... Cross origin requests are  
only supported for protocol schemes: http, data, https.
```

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

<https://threejs.org/docs/manual/en/introduction/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.

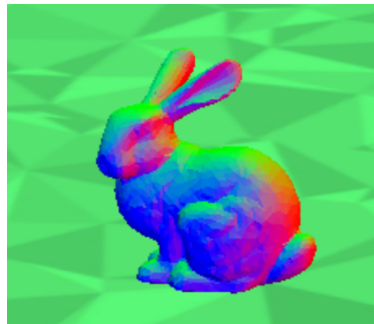


1. (70 points)

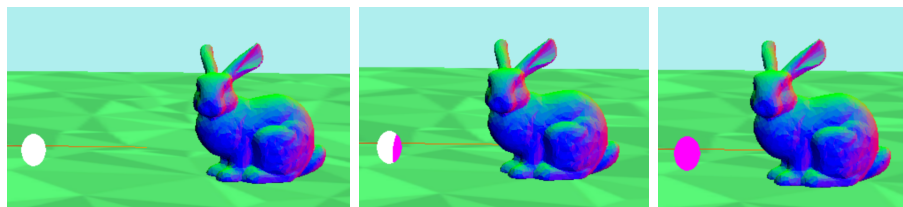
(a) **20 pts** Moving & Coloring the Bunny.

The variable `bunnyPosition` (the position of the bunny center in world coordinates) declared in `A1.js` is changed using the keyboard, and passed to the bunny vertex shader (in `bunny.vs.glsl`) using uniform variables. First, modify the bunny vertex shader to move the bunny in response to keyboard input.

Then, change the color of the bunny using the interpolated normals from its vertex shader `bunny.vs.glsl`, in the bunny fragment shader (in `bunny.fs.glsl`). Important: **do not** use Three.js functions; you must modify the shader for credit.

(b) **30 pts** Laying Colourful Eggs.

For this part you need to modify `A1.js` and the egg shader `egg.fs.glsl`, to make the bunny lay white eggs with the 'X' key that change to purple with distance when the bunny is in close proximity to them, and change back to white when the bunny moves far enough away. To lay the eggs you will need to modify `A1.js`, there is already an egg in the scene you can use code from to make the geometry; add an egg at the position of the bunny in response to keyboards input, pay close attention to how the egg already in the scene is added. To change the colour of the eggs you will need to pass relevant information about the position of the bunny to the egg shader to do this.



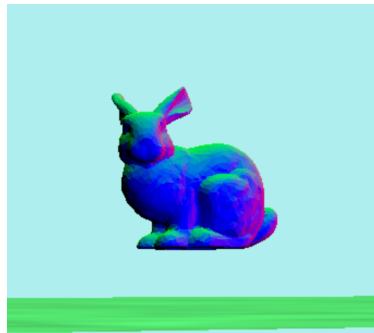
Hint 1: See how uniforms are passed to the bunny shaders. *Hint 2:* You should pass the necessary information about the bunny to the egg shaders. *Hint 3:* See how varying variables are passed to the bunny fragment shader. *Hint 4:* You may use one of the predefined transformation matrices, listed below. `uniform mat4`

```
modelMatrix;
uniform mat4 modelViewMatrix;
```

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

(c) **20 pts** Hippity Hop, Making the Bunny Jump!

Modify `A1.js` and the bunny shader `bunny.vs.glsl`, to make the bunny jump up and back down when you press and hold 'Z'. One simple way is to move the vertices of the bunny up and down in a cyclic manner.



Hint 1: You may find it useful to increment a variable on each update in `A1.js`, resetting it when appropriate. *Hint 2:* You should pass an appropriate uniform variable to the bunny shader.

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 more interesting ways.
- explode the model along face normals to view all the triangles that make it up.
- change the eggs in some dynamic way.
- force the bunny to complete a full hop cycle (no snapping back to its default position).

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
```