CPSC 314
Assignment 2: Transformation

Due 11:59PM, February 19th, 2021

1 Introduction

In this Assignment you will utilize your knowledge of transformations to make things move. Here we will study how to build and animate with object hierarchies. As for our subjects, we continue on from Assignment 1 with our Arbitrary Armadillo and the Contagious Coronavirus. We removed the coronavirus spikes in this assignment for simplicity.

1.1 Getting the Code

Assignment code is hosted on the UBC Students GitHub. To retrieve it onto your local machine 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://github.students.cs.ubc.ca/cpsc314-2020w-t2/a2-release.git
```

1.2 Template

- The file `A2.html` is the launcher of the assignment. Open it in your preferred browser to run the assignment, to get started.
- The file `A2.js` contains the JavaScript code used to set up the scene and the rendering environment. You will do most of your work here for this assignment.
- The folder `glsl` contains the vertex and fragment shaders for armadillo, eye and sphere.
- 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.

2 Work to be done (100 points)

First, ensure that you can run the template code in your browser. See instructions in Assignment 1. The initial scene should look as in Figure 1. Study the template to get a sense of how it works.

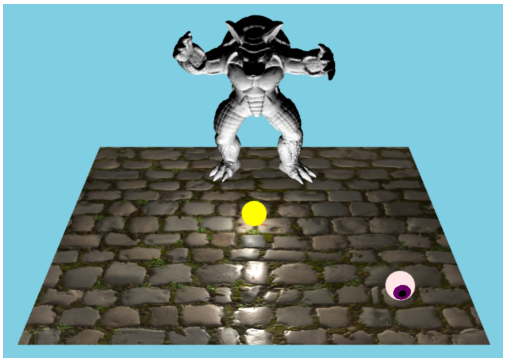

Figure 1: Initial configuration

1. **Part 1: Required Features**

   (a) **(15 points)** Armadillos have Eyes. Your task here is to bless the Armadillo with a pair of googly eyes. One eyeball model is already provided. Your job is to place two copies of this eyeball onto where the eyeballs should be on the Armadillo using an appropriate transformation.

   Hint 1: You can do this entire question in `A2.js`.
   Hint 2: Don't worry if the eyes are initially pointing backwards into the Armadillos head, in the next question you will have the opportunity to orient them correctly.

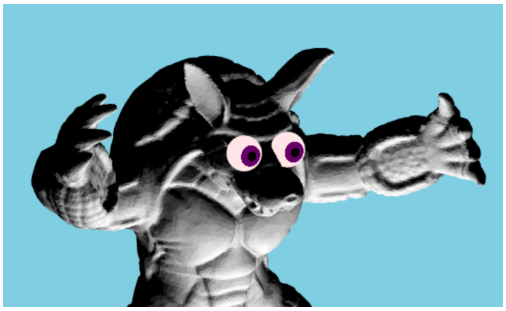

Figure 2: Question 1a). Give the Armadillo eyes.

   (b) **(15 points)** Observe the virus. What good are eyes if they don't `lookAt` anything? Direct the Aramdillo's gaze towards the nefarious floating coronavirus. Ensure that the eyes look at the coronavirus as it moves around the scene as shown in Figure 3

Hint 1: `THREE.Matrix4 and THREE.Object3D` have a method called `lookAt` that may be of use.

Figure 3: Question 1b). Rotate the eyes to look at the coronavirus.

(c) **(20 points)** Laser eyes! Arm the Armadillo with eye lasers. Shoot the lasers from the Armadillo's eyes at the coronavirus when it gets too close. A distance threshold (units in world space) named `LaserDistance` is provided in `A2.js`. To be more specific, create two lasers from the eyes to the virus. You need to create the geometries and materials by yourself. Figure 4 shows a possible result.

Hint 1: You may use any kind of material for the lasers (even a `ShaderMaterial`).
Hint 2: Note that the distance threshold between the virus and the eye is given in the world frame, but the scale of the laser will likely be in a different frame.

Figure 4: Question 1c). Give the Armadillo laser eyes!

(d) **(30 points)** Dodgy Armadillo. Make the Armadillo dodge the virus like in "The Matrix". To do this, first segment the Armadillo into two halves. Identify the vertices above the waist and below the waist in the vertex shader. The vertices below the waist will be positioned in the original coordinate frame of the Armadillo. The vertices above the waist will be rotated away from the coronavirus as it approaches the Armadillo. Use `MinDistance` in `A2.js` for the horizontal dis-

tance threshold and `MaxHeight` for the vertial distance threshold. Deform the top half of the Armadillo only if the virus is located within these distance thresholds.

The end result should look similar to Figure 5. Don't panic if the triangles at the Armadillo's waist are significantly distorted for larger rotations, as in the figure. For this question you will need to modify `armadillo.vs.glsl` and `A2.js` Note that as the Armadillo rotates, the eyes and lasers should rotate with it.

This is the hardest question so make sure to budget your time.

Hint 1: Be careful with coordinate frames. Review the lectures on "Transforming about an auxiliary frame" and Section 5.2.1 of the Textbook. You will need to compute inverses of your coordinate frame transforms in the shader and `A2.js`.

Figure 5: Question 1d). Dodge the virus!

(e) **(20 points)** He's a smooth operator. Here you will blend the deformation of vertices around the waist smoothly, to produce a result as in Figure 6. To do this, weight the deformation to produce the dodge by the height of the vertex relative to the waist. Review the smooth skinning lecture in class for the details.

Figure 6: Question 1e). Smooth the skin at the waist

2. **Part 2: Creative License (Optional)**

   You have many opportunities to unleash your creativity in computer graphics! In this **optional** section, and you are invited to extend the assignment in fun and creative ways. We'll highlight some of the best work in class. A number of exceptional contributions may be awarded bonus points. Some possible suggestions might be:

   - Rig the Aramadillo's head to turn to follow the coronavirus in addition to the eyes.
   - Make the laser beams glow like real laser beams obviously do.
   - Push the vertices of the Armadillo away radially from the coronavirus as it moves into the Armadillo. This should achieve a reasonably realistic indentation in the Armadillo.

2.1 Hand-in Instructions

You must write a clear README.txt file that includes your name, student number, and CWL username, instructions on how to use the program (keyboard actions, etc.) and any information you would like to pass on to the marker. Create a folder called "a2" inside your "cs-314" directory. Within this directory have two subdirectories named "part1," and "part2", and put all the source files and your README.txt file for each part in their respective folder. Do not use further sub-directories. The assignment should be handed in with the exact command:

`handin cs-314 a2`
on a department computer, which you can SSH into.
You may also use `Web-Handin` by following this link `https://my.cs.ubc.ca/docs/hand-in`, logging in with your CWL credentials, and writing "cs-314" for the course, "a2" for the assignment name, and zipping your assignment folder for submission.
It is always in your best interest to make sure your assignment was successfully handed in. To do this, you may either use the `Check submissions` button in Web-Handin, or using the -c flag on the command line `handin -c cs-314 a2`.