CPSC 314
Assignment 2: Fun with Transformations

Due 4PM, Feb 20, 2015

1 Introduction

The main goal of this assignment is to create an understanding of rotation, translation, and scaling transformations. This will be achieved by first creating a hierarchy of shapes representing a very simple character comprised of boxes and ellipsoids. Then you will be animating the hand and eyes of this simple character.

The template code provides a starting base for the character creation, shown in Figure 1. A has been provided therein. This template code should be extended to perform the following tasks in succession :

- Add objects in a hierarchy representing an arm of the character.
- Animate the arm in a waving pattern.
- Add objects in a hierarchy representing the eyes to the character.
- Create two objects representing two laser beams pointing from the eyes to the gem.

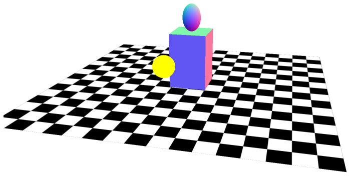

Figure 1: The provided template.

The template is very similar to the previous one, with the main differences in the JavaScript file:

- The code which loaded the armadillo has been replaced with one which now constructs the torso and the head of the character. The torso is constructed in the world coordinate frame and the head is constructed in the torso coordinate frame.

- A new placeholder function, `updateBody`, is provided. This is called every frame and can be used to animate the geometry.

1.1 Constraints

Three.js provides several built in methods for easy parenting, rotation, translation and scaling of objects. However, since the purpose of this assignment is to understand the principles behind these methods, **the use of these tools is not allowed**. More specifically, you must:

1. Explicity declare new matrices using the `Matrix4().set` method, and pass the required matrices to the shaders. Creating matrices through operations from other existing matrices is also permited (eg: multiplication).

2. Objects must be moved by changing their coordinate frame using the `object.setMatrix` method. Operations on their `position`, `rotation` and `scale` attributes are not allowed.

You can find examples of matrix initialization and object positioning as described above in the template code.

2 Work to be done (100 pts)

First, ensure that you can run the template code in your browser. Next, study the template to get a sense of how it works. Then, proceed to the following tasks :

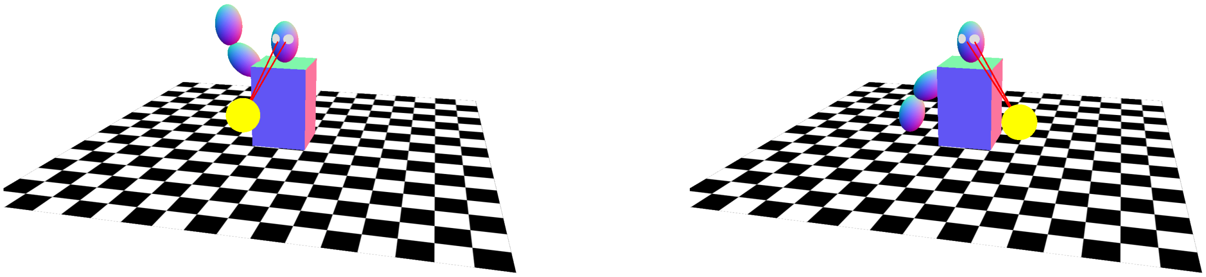

Figure 2: The intended result

Part 1 : Exercises (70 points)

1. **Build an arm (10 pts)**

   Add an arm to the character body created in the template code. The arm can be constructed using two ellipses representing an upper arm, placed with respect to the torso coordinate frame, and a forearm, placed with respect to the upper arm coordinate frame.

2. **Animate arm (25 pts)**

   Rotate the upper arm and forearm coordinate frames so that the arm waves up and down. The forearm should not only move with the upper arm, but also have an independent motion with respect to the upper arm. Both the arm and the forearm should move in a cyclical manner. Please make sure that the animation is neither too fast nor too slow to facilitate easy visualization.

3. **Build eyes (10 pts)**

   Add two spherical eyes to the character body. They should be placed with respect to the head coordinate frame, one slightly to the left, while the other to the right with respect to the center of the head. Make the eyes have white color.

4. **Track gem with laser pointers (25 pts)**

   Place the `leftBeam` and `rightBeam` objects representing laser beams (defined in the template) at the center of each of the eyes. In other words, place each beam at the origin of the eye coordinate frame. Then, rotate and scale it it so that it points towards the center of the gem, given by `gemPosition` in world coordinates. *Hint :* Read about polar coordinates.

Part 2 : Creative License (30 pts)

For this part we want to see what you can do. If you have any doubts, make sure to OK it with a prof or TA. Some possible suggestions might be:

- Complete the character to have both arms, legs etc and animate them.
- Create non-humanoid or non-bipedal characters with animations.
- Animate the armadillo's arms and legs. You can investigate "linear blend skinning" for better effect (advanced).
- Load animations from text files to animate the transforms for various body parts. These animations can be output from various DCC such as Maya, 3D Studio Max etc.

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

3 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 "a2" 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 a2
```