CPSC 314
Assignment 2: Fun with Transformations

Due 23:59:59, Oct 19th, 2015

1 Introduction

The main goal of this assignment is to practice rotation, translation and scaling transformations. You will be responsible for creating, applying, and updating the transformation matrices in javascript without three.js matrix creation helper functions. However, the overall transformation matrices should remain relatively simple. There are three main tasks to be done: finish the body geometry, animate the legs to run, and add some further creative components to the scene.

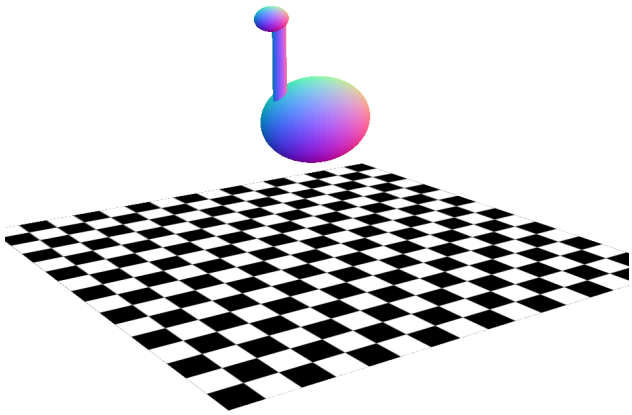

Figure 1: The provided template.

The template code is very similar to the previous assignment template. The main differences are in the javascript file:

- The armadillo is no longer present. Instead, a part of the new body geometry has been constructed for you. There is a torso, (placed with respect to the world coordinate frame) a neck, (placed with respect to the torso coordinate frame) and a head (placed with respect to the neck coordinate frame).

- A new incomplete function called `updateBody` is provided. It is called every frame and you should complete it to animate the geometry.

2 Important Rules

Three.js provides several tools for easy parenting, rotation, translation and scaling of objects. Yet, as the purpose of this assignment is to understand the principles behind all of this methods, **you are not allowed to use them**. More specifically, you must:

1. Explicity declare new matrices using the `Matrix4().set` method. Creating matrices through operations from other existing matrices is also permited (eg: multiplication). Generating matrices through utility methods such as `Matrix4().makeRotationX` is not allowed.

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.

3. Explicitly parenting objects using the `parent` attribute is not allowed.

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

3 Work to be done (100 pts)

First, ensure that you can run the template code in your browser. See the instructions from the previous assignment. **Remember to follow the requirements described in the previous section.** Study the template to get a sense of how it works.

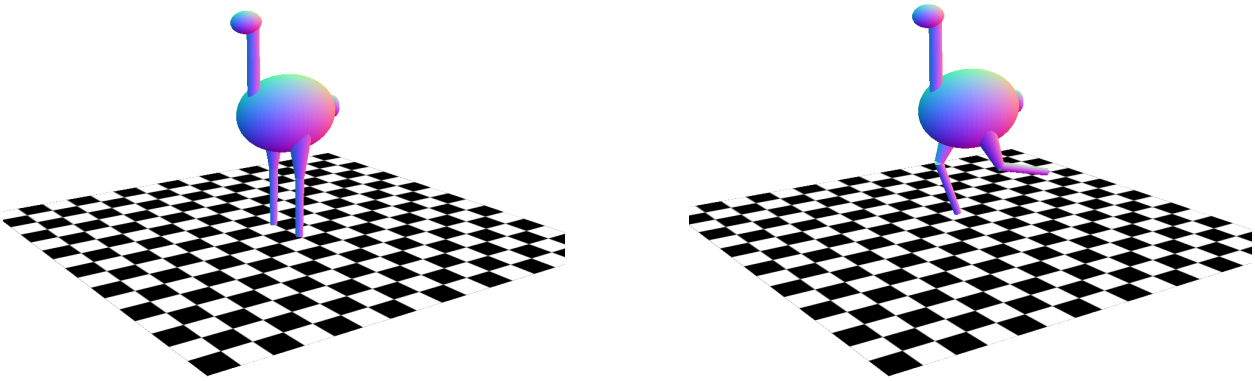

Figure 2: An example result for Q1

1. Exercises (70 points)

   (a) **15 pts** Build legs.

   Add two legs to the body. Each leg should be made of two parts, both of which are cylinders: a thigh, (placed with respect to the torso coordinate frame) and a lower leg (placed with respect to the thigh coordinate frame).

   The leg geometries have already been built for you. You can use those geometries or create your own. You do not need to write your own shaders because you can re-use the default material that has already been applied to the torso, neck and head.

   (b) **35 pts** Animate legs.

   Rotate the thigh and lower leg coordinate frames so that they move like the ostrich is running. The lower leg should not only move with respect to the thigh, but also have an independent motion. Make sure the running animation is cyclical and continues to move back and forth. Also make sure they are moving at a reasonable speed to be easily observed. You should use a timer to keep animation speed consistent instead of animating the legs every frame.

   The variable t has been declared for you already and contains total ellapsed program time. Obviously, this value will always increase and we need a cyclical animation. You will therefore need a function to turn a constant value into a cyclical one and use it to change the orientation and position of the objects you want to animate.

   (c) **20 pts** Add more poses.

   Give the ostrich 5 different poses for the static body coordinate frames. Pressing the number keys between 0 and 5 should each show a different pose. For example, you could press 1 to have the ostrich be jumping and 2 to have it reach its neck towards the ground. The number 0 should show your running animation.

2. Creative License (30 pts)

   For this part we want to see what you can do. We encourage you to implement some new shaders. Your idea 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:

   - Make vertex shaders for some of the moving objects and make use of their dynamic coordinate frames (defined in glsl code as `modelMatrix`).
   - Add more limbs to the ostrich and make your static frames support them.
   - Add another ostrich that is animated/posed separately.

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

3.1 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 "a2" under your "cs314" directory. Within this directory there are 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
```