

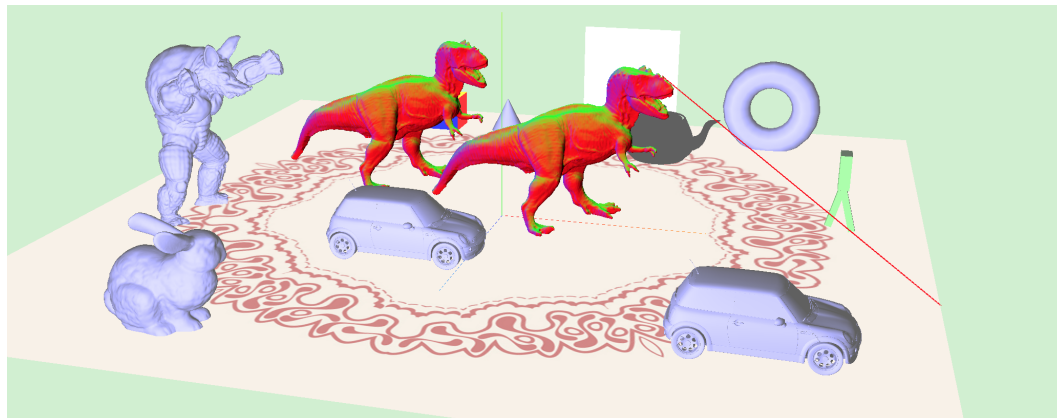
# CPSC 314 Assignment 3

Due 23:59:59 on February 5, 2018

## 1. (27 points) Animating with Transformations

Your task will be to create and animate a scene with dinosaurs. The learning goals are to be able to create and use transformation hierarchies, creating basic character animations using these, and using additional `three.js` functionality, including keystroke-driven interactions.

Be as creative and ambitious as you like. The best work will be posted on the course web site in the Hall of Fame and will be shown in class.



### Important Rules:

As illustrated in the template code, you are allowed to use some of the basic matrix support functions provided by `Three.js`. This includes: `Matrix4.multiply()`, `Matrix4.makeTranslation()`, `Matrix4.makeRotationX()`, `Matrix4.makeRotationY()`, `Matrix4.makeRotationZ()`, `Matrix4.makeScale()`.

`Three.js` provides tools to help create and use hierarchies, mainly via the specification of a `parent` frame, for objects. However, for this assignment, all objects will have the world coordinate frame as their parent, which is the default. Instead, you will be directly modifying the  $4 \times 4$  modeling transformation associated with each object, i.e., `obj.matrix`, where 'obj' is the object in question. This matrix should take points to the world coordinate frame. The template code provides useful examples in this regard.

### Requirements:

- (a) (5 points) First, be able to demonstrate an understanding of the fundamentals of the code to your TA. There are no specific deliverables for this aspect.
  - (i) Ensure that the scene loads and plays the basic animation. Remember to enable local file access.
  - (ii) Understand how objects are loaded from OBJ files e.g., see `initFileObjects()` and are then instanced, i.e., see `onResourcesLoaded()`. Add an extra minicooper somewhere in your scene at a desired location and in a desired orientation. Test your scene.
  - (iii) Now, comment out all the unused OBJ objects, except for the trex. This will

speedup the loading for much of the remaining work below. The many objects exist in the template code simply as examples. Test your scene.

(iv) Understand how the `trex` and `mydino` keyframes work. Make the `trex` jump higher, and make `mydino` do a bigger leg swing.

(v) Understand how `mydinoSetMatrices()` makes use of animation variables, `avars`, to build the relevant transformation matrices. Increase the length, i.e., height, of the legs from 1m to 2m (see the end of `initObjects()`). Modify the transformation matrices in `mydinoSetMatrices()` to accomodate this. Note that because the origin of the box geometry used for the body parts is at the center of the box, modeling a joint requires translating to the edge of body, rotating, and then translating to the center of the child link.

- (b) (5 points) Replace the human-looking `mydino` with a hierarchical articulated figure that has at least 15 links. Look at existing images of dinosaurs and dinosaur skeletons, i.e., Google images, in order to be able to construct skeletons that have a better resemblance to dinosaurs. At this point, you could use fixed rotations for the joint angles. You can build it out of primitives of your own choosing. Custom geometry is fine as well, although document how it was created. Creating a sketch on paper will be helpful for keeping track of the dimensions and joint locations. Be proud of your creation :-)
- (c) (3 points) Create a reasonable in-place walk cycle for your dinosaur. Focus on the leg and arm tion, and do not worry yet about the forward motion of the `mydino`.
- (d) (4 points) Now animate the position and orientation of `mydino` so that it follows a curved path, or a circular path. Animate the motion of a `minicooper` instance so that it approximately follows your dinosaur.
- (e) (2 points) Update the “tracking laser beam” (`laserLine`) geometry so that it acts as a line that connects the front of the `minicooper` to the body of `mydino`; see `laserUpdate()`.
- (f) (2 points) Add keystrokes commands to: turn the laser on-and-off and to be able to play the motion backwards in time. Add global-scope variables as needed.
- (g) (4 points) Enhance your scene, via multiple geometry and lighting improvements.
- (h) (2 points) Implement two of the following features:
- keystroke to make changes to the lighting;
  - develop a keyframed dancing motion for your dino
  - have a key that makes your dino jump or flip
  - have a `trex` follow the `minicooper`, which is following you
- (i) (4 points) Ideas for bonus marks:
- create a plant or tree with animated branches or leaves
  - complete more of the suggested ideas above
  - have your dino leave footprints, add flying birds
  - be able to steer your dino using keystrokes
  - more ideas of your own creation

To hand in:

- Hand in a README.txt file. This should include your name, student number, and user name, as well as any information you wish to pass along to the marker. List any external resources (web pages, books, consultation with others) that you used. This assignment is to be completed individually. Comment on your implemented features and keybindings in your README file.
- To hand in, create a `cs314` subdirectory under your home directory, and create an `a3` subdirectory underneath that. Copy all the files that you wish to hand in to this directory. The assignment can be handed in using `handin cs314 a3`.
- Demonstrate your code to a TA in the lab, or at another to-be-arranged time.

We look forward to showing off the creative results from this assignment in class!