Introduction to Computer Graphics CS 174A: Assignment 1 **Spring 2017**
Weight: 15 %
Points: 24
Collaboration: None permitted. If you discuss this assignment with others you should submit their names along with the assignment material. Using code from previous offerings of the course constitutes plagiarism and is prohibited.

Submission: Follow the instructions carefully to avoid point reductions.
Submit a zipped file (UID.zip — e.g: 313939200.zip) that includes the template files and if applicable a README file explaining which parts you only partially fulfilled, or any key actions you feel are necessary for the grader to successfully inspect your project.

Assignment:

Write a program using our javascript template (provided on the course Piazza forum), that **draws the scene shown in the Homework 1 sample videos**. The sample videos are also found on the Piazza forum under "Resources/Resources". **A diagram / blueprint showing how the hinges should hook up to each other is included on the page below**. For drawing objects, use the template's draw() function on a box (cube) and a ball (sphere) that have been submitted to the GPU, like the example code provides. For now don't worry about the details of how data is sent to the graphics card.

JavaScript has no operator overloading, so operations like +, -, *, +=, *=, etc. are unavailable for the provided matrix types. Instead use mult() to return the product of two matrices. For vector math (especially in future projects) use mult_vec() to return the product of a matrix and a vector (careful not to accidentally use mult() instead) and be aware that matrices can do transpose() and inverse(). For plain vectors you have add(), subtract(), dot(), cross(), mix(), scale_vec(), length(), and normalize().

Requirements:

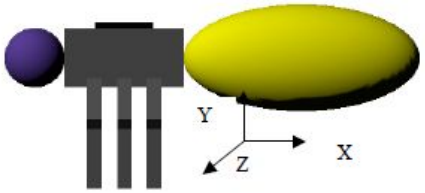
(a) You must use a hierarchical approach to model the complex objects. This applies **both** conceptually (during your order of matrix transformations) and programmatically (breaking up your code into a **hierarchy of subroutines**). **(5 Points)**
(b) Model a static ground plane. **(1 Point)**
(c) Model a tree that has a trunk made of 8 segments and a sphere for foliage. **(2 Points)**
(d) The tree must visibly sway as shown by the sample video. **(2 Points)**
(e) Trunk parts rotate around the **middle of the bottom face**. **(4 Points)**
(f) Animate the wings and legs of the wasp. The legs have **two segments each**, not one. You may use the same value for more than one angle. Pieces should rotate along the bee's lengthwise axis, and should stay connected at the corner edges where they touch one another. You must use boxes for the legs and wings to demonstrate that you can keep them in contact with the body corners like in the diagram below. **(5 Points)**
(g) The wasp flies in a circle around the vertical axis, and it should always be aligned with the tangent of the circle. The wasp must move up and down. **(5 Points)**

You must rotate objects around the correct point; i.e., where they touch the parent object matters. Pay special attention to the locations of these hinges - edges where two boxes make contact - and center your rotations along those. You will be mainly graded on that and on the fluidity of motions.

Your scene must be qualitatively similar to the one provided, but need NOT match the exact motion or dimensions or colors of the sample code. **In fact, your scene MUSTN'T exactly match anyone else's submission**. Use colors, sizes, or an added feature to customize your bee and make it unique so we do not get your submission mixed up with the others. In doing so, you must still adhere to the details of requirement (f) -- remember, two box leg segments that each hinge at a corner edge.

Hints:
(a) Create a method draw_leg() and use it for each of the legs.
(b) Read the top two classes in the .js file with example scenes, and start coding there. Use the **graphics_state.animation_time** variable whenever you need movement as a function of time.
(c) Functions of the form f(t) = a + b*sin(w*t) are useful for modeling periodic motion.

(Cont'd - page 2)


Y
Z
X