Assignment #2
CS174A – Introduction to Computer Graphics - Fall 2014

DUE: October 29, 2011 by 11:55PM

YOU MUST UPLOAD THE FOLLOWING IN SOME TYPE OF CONTAINER (TAR,ZIP,etc)
- A README DESCRIBING WHAT YOU DID AND DID NOT DO
- THE SOURCE CODE FOR YOU SUBMISSION – INCUDE EVERYTHING WE NEED TO COMPILE AND RUN YOUR SUBMISSION

100 POINTS AVAILABLE / PARTIAL CREDIT MAY BE AWARDED

The following is **required** for the assignment and provides 100 points of basic credit:

1. Get a simple WebGL capable window to display without error – 5 points.

2. Implementation of the various shader codes needed without error – 10 points.

3. Display eight (8) cubes using a perspective projection at (+-10,+-10,+-10) from the origin each in a different color. Your initial camera position should encompass all eight cubes. Provide a key command 'c' that cycles the color of all cubes on each key press. – 30 points.

4. Implement a camera navigation system using the keyboard. Up, down arrow control the altitude of the camera and left and right arrow control the heading (azimuth) of the camera. The letters i, j, k and m control forward, left, right and backwards respectively. You can make each press move 1 degree or 0.25 units. Translation should be relative to the camera's current rotation. Include a key to reset the view back to the start position – 40 points.

5. Using the keys 'n' and 'w' make the *horizontal* field of view narrower or wider. Using the key 'c' display an orthographic projection of a cross-hair in the center of the screen – 10 points.

6. Implement the assignment in a clean and understandable manner. You can use whatever coding style you prefer but your code must be readily understandable for grading (e.g. comments) – 5 points.

Extra Credit

1. Instance each of the cubes from the same data and implement the cube(s) as a single triangle strip – 5 points.

2. Smoothly, continuously and individually either rotate or scale each of the cubes while you application is running. The rotation speed should be constant and the scale should only vary by ten percent – 10 points.

3. Implement your navigation system using quaternions - 10 points.