Due: 2/20 (11:59pm)

Requirements:

- Write a WebGL program that allows user to view, zoom and rotate obj model (Fig. 1). Name your source code `hw1.html` and `hw1.js`. The program should meet the following requirements:
  - See the accompanying video. Your program must basically look and work the same as in the video.
  - The program must access and visualize `buddha2.obj` with URL: `http://www.cs.umsl.edu/~kang/htdocs/models/buddha2.obj`
  - You may notice that the obj viewer we have developed in class doesn't work properly on `buddha2.obj`. Your job is to figure out why and make necessary modifications to make it work on this model.
  - Note that `buddha2.obj` must be texture mapped with the associated texture image.
  - Add `dat.gui` components to let user switch between 3 shaders: *Gouraud*, *Phong*, and *Flat + Wireframe* rendering (see video).

What to submit:

- Submit all your **source files (.html, .js)** that are needed for compilation, including **library files/folders**. *Missing library files/folders will incur point deduction.*
- Make sure your **library folder/files** are in the right location relative to your main program (.html), such that when your main program (.html) is clicked as is, it should run without problem. *Failure to do so will incur point deduction.*

How to submit:

- Use Canvas Assignment Submission system to submit your source files.
- Make sure to zip all your files/folders into `hw1.zip`, then submit your `hw1.zip` as a single file.

Policy

- Do all the assignments on *Chrome Development Tools* using HTML, JavaScript, and GLSL ES.

Figure 1: buddha2.obj

- At the top of each source file, provide comments specifying the author, date, and a brief description of the file.
- Source code must contain enough comments here and there to make it easy enough to follow. Insufficient comments could lead to point deduction.
- Incomplete program will get almost no credit (e.g., program does not run due to compile errors or program terminates prematurely due to run-time errors).
- *Thou shall not covet thy neighbor's code*. If identical (or nearly identical) submissions are found among students, every student involved will get automatic zero for the assignment. The same goes for copying existing code from online source.
- If a student makes multiple submissions, only the last submission will be considered valid.