Project C: Litt Up
EECS 351: Kevin Chen, kjc004

A Brief Intro:

This project demonstrates further application of concepts covered in EECS 351: Intro to Graphics. Building on top of Project B, we turn our attention to the application of lighting and materials in our WebGL world. Object components now have a "material" composition that defines their interactions (e.g. being lit up and reflected off of) with light. We also examine various algorithms for computing shading.

The world consists of the same five objects as my last project, with one tweak: the pendulum object now has a sphere on its end instead of an icosahedron to better demonstrated specular reflections.

User Manual:

Instructions are displayed at the bottom of the rendered index.html webpage, much like project B.

The WASD keys control how the observer sees. In particular, AS determine horizontal rotation (e.g. turning), while WD determines vertical (e.g. looking up and down). Then, J and K keys respective zoom out and in on where the observer is looking. New to this project is H and L, which allow for sideways strafing of the camera. In addition, I've added various controls for the lights — in the top right, datGUI controls properties of the overhead light, and various buttons can toggle on/off the headlights, switch shading + lighting methods, and half moition.

Results:

FIGURE 1. All lights on.
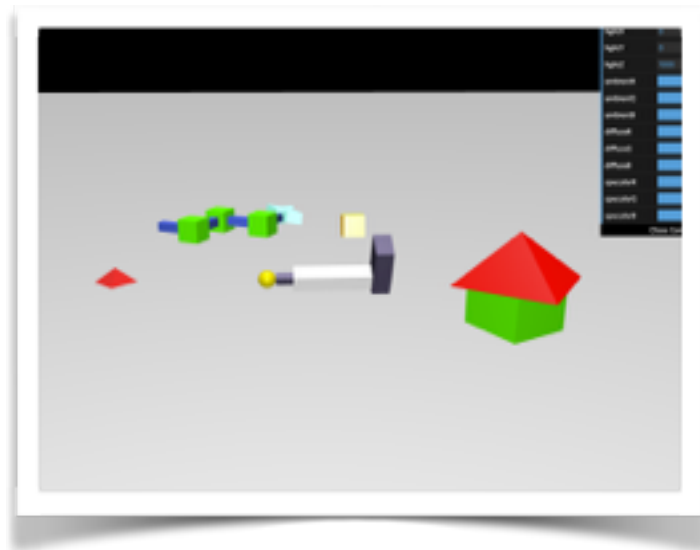
I've included five shapes from my last project, animated (Fig. 1). The cube in the back right is made of brass, the house an emerald base and ruby roof (expensive!), the pyramid red plastic, the pendulum obsidian (slider, connector), chrome (handle), and shiny gold (sphere). Finally, the jointed object is made with a combination of emerald, blue plastic, and turquoise. I mostly selected these colors because I liked the way they looked. In this first image, all the lights are on; we can adjust the
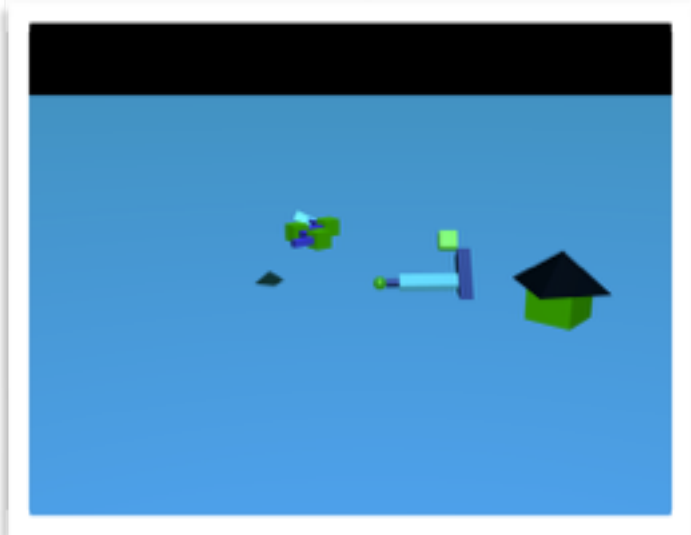
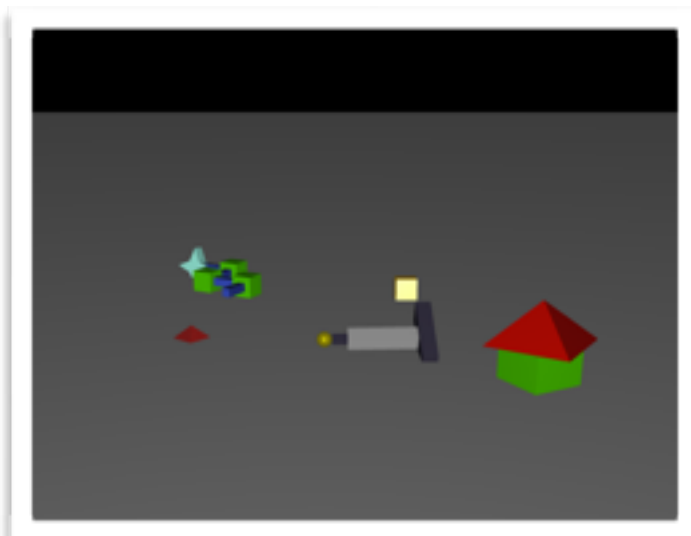**FIGURE 2.** Decreased ambient and diffuse red colors.

FIGURE 3. Overhead light off.

overhead light via datGUI on the side (Fig. 2), or turn it off altogether (Fig 3.)

By default, we use Phong shading and Phong-Blinn lighting, but we've also implemented Gouraud shading (Fig. 4). To note: Gouraud shading with both lights on is fairly blinding, so we've turned off the headlights in the image. Phong vs. Phong-Blinn lighting differences are basically not noticeable, so we omit that image. You can examine "fshader.esgl" file for implementation details (basically using the reflect function).
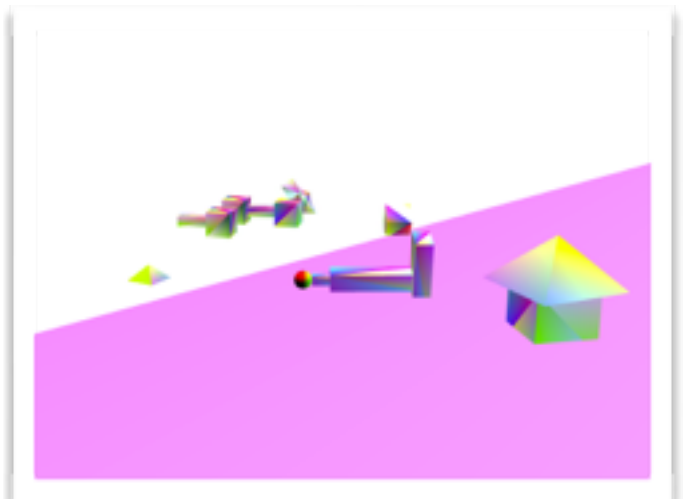

FIGURE 4. Gouraud shading

I use the same shapes as before, so I've once again included the same scene graph as the previous assignment.