Anthony Borza

Project 4: WebGL 3D

CMSC 405 7980 Computer Graphics

University of Maryland University College

Due Date: May 7, 2017

Table of Contents

Brief Description of Project:

The goal of this project was to create a unique 3D animated scene composed of WebGL graphic components. The scene uses lighting, textures, frame buffers and multiple objects. This scene required that the size is 640 x 480, includes at least 10 different objects, and uses multiple lighting effects, and texture effects on different materials. This scene also includes checkboxes and buttons. The frame buffers are used to organize the memory resources that are needed to render the scene.

Design and Functionality:

Project 4.html

- Implements shades for both vertex and fragment.
  - This is used to hold attributes for coords, normal, texture cords, and eyeCoords.
  - Defines material and lighting properties to be used for the scene.
  - Defines a lightEquation method, as well as main method for handling the lighting for daytime and nighttime in the scene.

- Implements the following scripts to be used in the main html file:
  - gl-matrix-min.js
  - trackball-rotator.js
  - basic-object-models-IFS.js

- Implements several variables for handling the following:
  - Location of coords attributes from the shader section of the program
  - Locations for uniform matrices, projection, normal matrix
  - Locations for the material and lighting properties
  - Defines projection matrix for mat4, and mat3
  - Defines objects that are to be created using the function createModel
  - Defines a matrix stack for implementing hierarchical graphics
  - Defines the rotation of the sun about the z-axis

- Implements a draw method which handles the following:
  - The color, perspective, and projection
  - The lights method

    - The world method

- Creates a setSpotlightDirection method to set up direction vector of a light, in eye coordinates.

- Creates a setLightPosition method to set up the position of a light, in eye coordinates.

- Creates a load texture method to load a texture. (Could not get this to work)

- Creates a lights method to control the lighting in the scene

- Creates a world method to add everything to the scene, which includes:
    - Adds grass to the scene which uses a disk
    - Adds a house to the scene
    - Adds a right, left, and 4 small trees to the scene
    - Adds a lamp to the scene
    - Adds house features to the scene
    - Adds person to the scene

- Creates a house method to include the following:
    - The roof of the house using a cone
    - The base of the house using a cube
    - The chimney of the house using a cube
    - The satellite pole on the house using a cylinder
    - The satellite head using a cone

- Creates a tree method to include the following:
    - The trunk of the tree using a cylinder
    - The top of the tree using a cone

- Creates a lamp method to include the following:
    - The lamp stand of the lamp using a cylinder
    - The top piece of the lamp stand using a cone

- Creates a house Features method to include the following:
    - The door for the house using a cube
    - The door knob for the house using a sphere
    - Two windows for the house using a cube

- Creates a person method to include the following:
    - Two legs for the person using a cylinder
    - The body for the person using a cube
    - The neck for the person using a cylinder

  - The head for the person using a cube
  - 2 arms for the person using a cylinder
  - 2 eyes for the person using a sphere
  - 2 eye pupils for the person using a sphere
  - A nose for the person using a sphere
  - A mouth for the person using a cube
  - Two shoes for the person using a cube

- Creates a person method to include the following:
  - Four legs for the dog using a cylinder
  - Body of the dog using a cylinder
  - Neck for the dog using a cylinder
  - Head for the dog using a sphere
  - Tail for the dog using a cylinder
  - 2 ears for the dog using a cone
  - 2 eyes for the dog using a sphere

- Creates a pushMatrix method to push a copy of the current modelview matrix onto the matrix stack.

- Creates a popMatrix to restore the modelview matrix to a value popped from the matrix stack

- Creates a createModel method that holds the data for the IFD, which is from the JavaScript file: basic-object-models-IFS.js.

- Creates a createProgram method that is used in the WebGL context gl, and returns the identifier for that program.

- Creates a initGL method that initialize the WebGL context

- Creates a frame method

- Creates a setAnimating method and is used to start the animation of the scene

- Creates a init method initialization function that will be called when the page has loaded.

- Create html features to include headings, buttons, checkboxes, paragraphs, and so on.

gl-matrix-min.js

- A JavaScript file from the text that is used with WebGL, and is used for high performance matrix and vector operations.

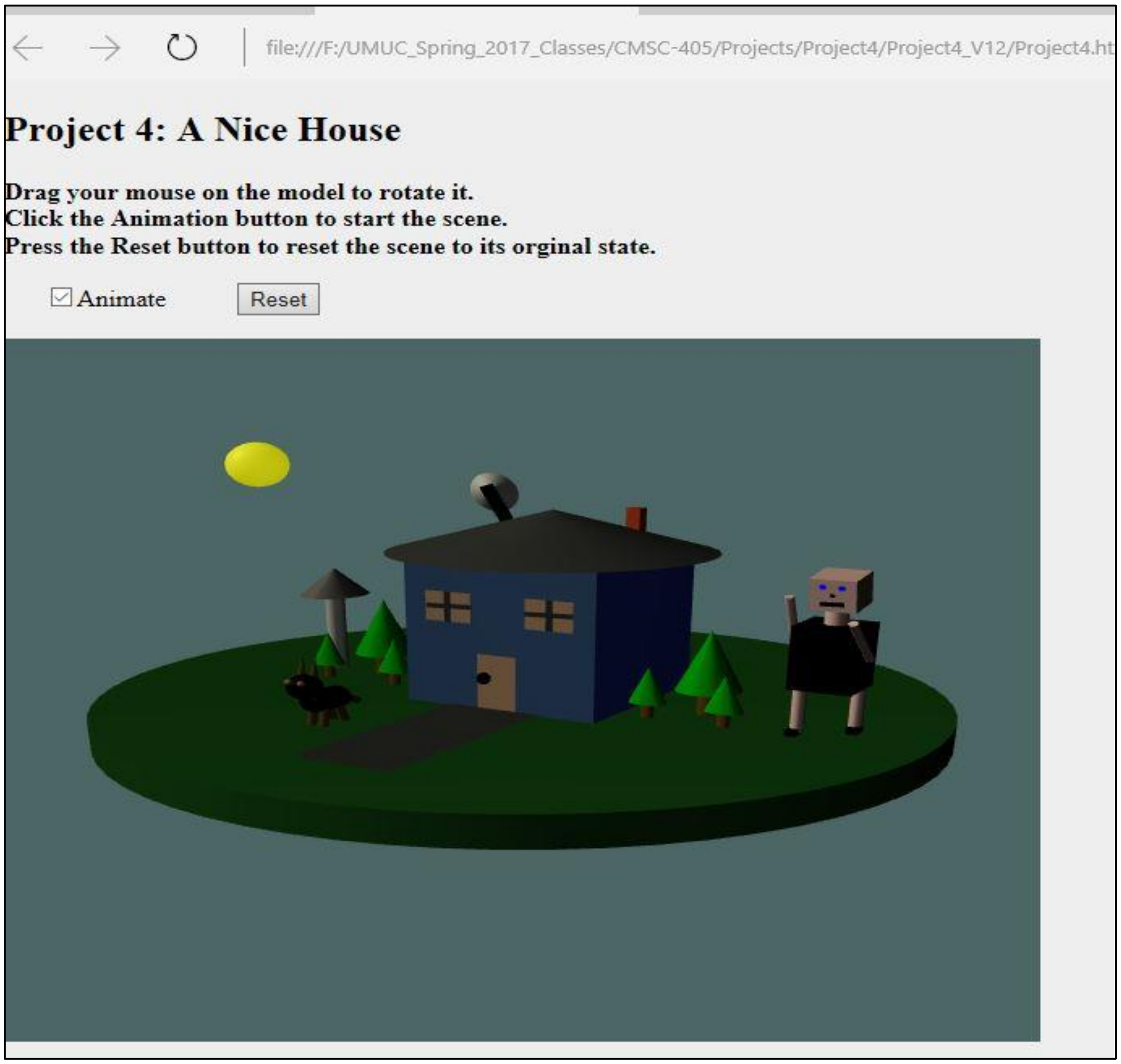
trackball-rotator.js

- A JavaScript file from the text that is used with WebGL, and is used to implement a trackball-like mouse rotation of a WebGL scene about the origin.

basic-object-models-IFS.js

- A JavaScript file from the text that is used with WebGL, and is used to create models in an IFS format that can be drawn using gl.drawElements with primitive type gl.TRIANGLES.

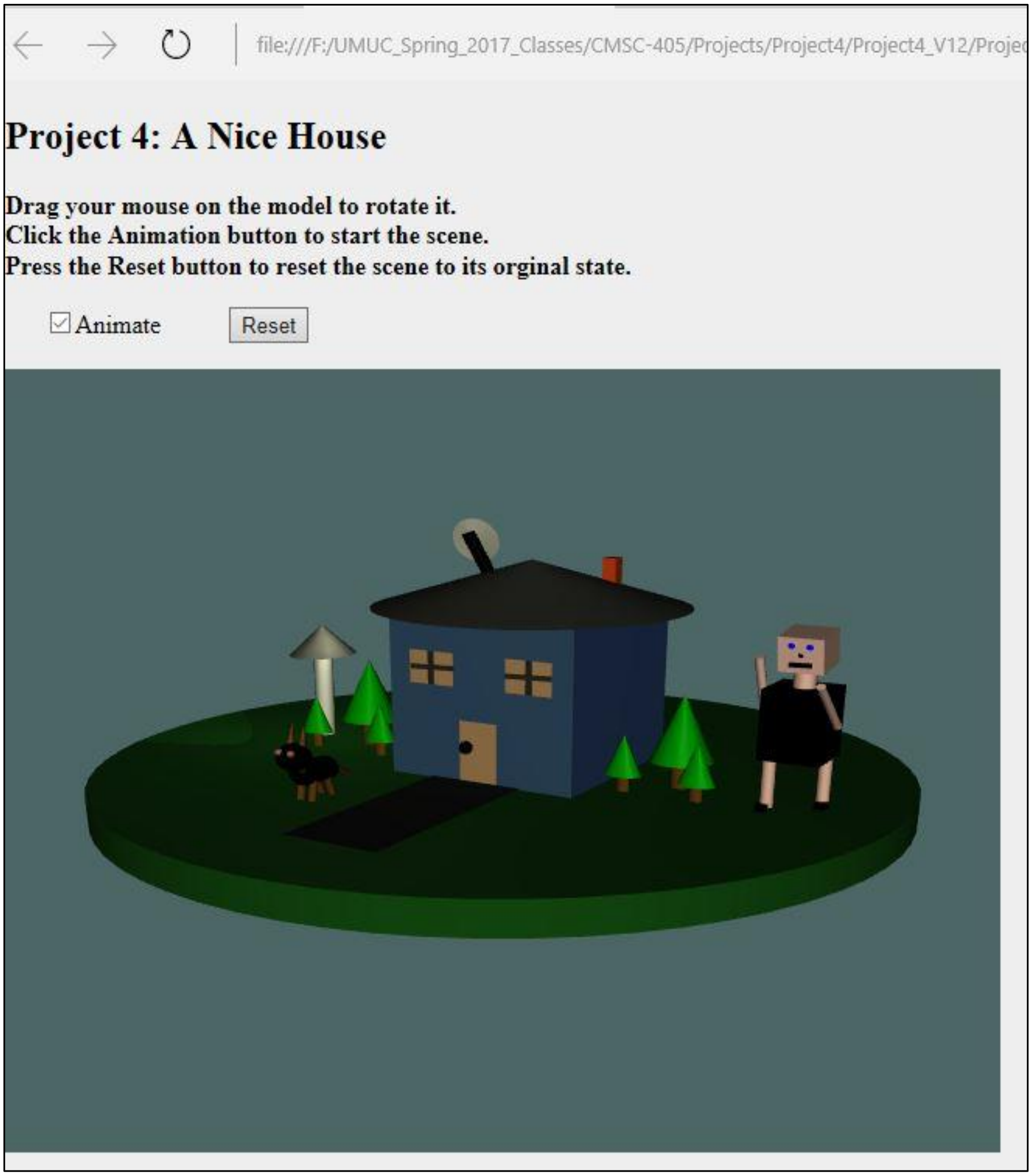
Screenshots:



file:///F:/UMUC_Spring_2017_Classes/CMSC-405/Projects/Project4/Project4_V12/Project4.ht

Project 4: A Nice House

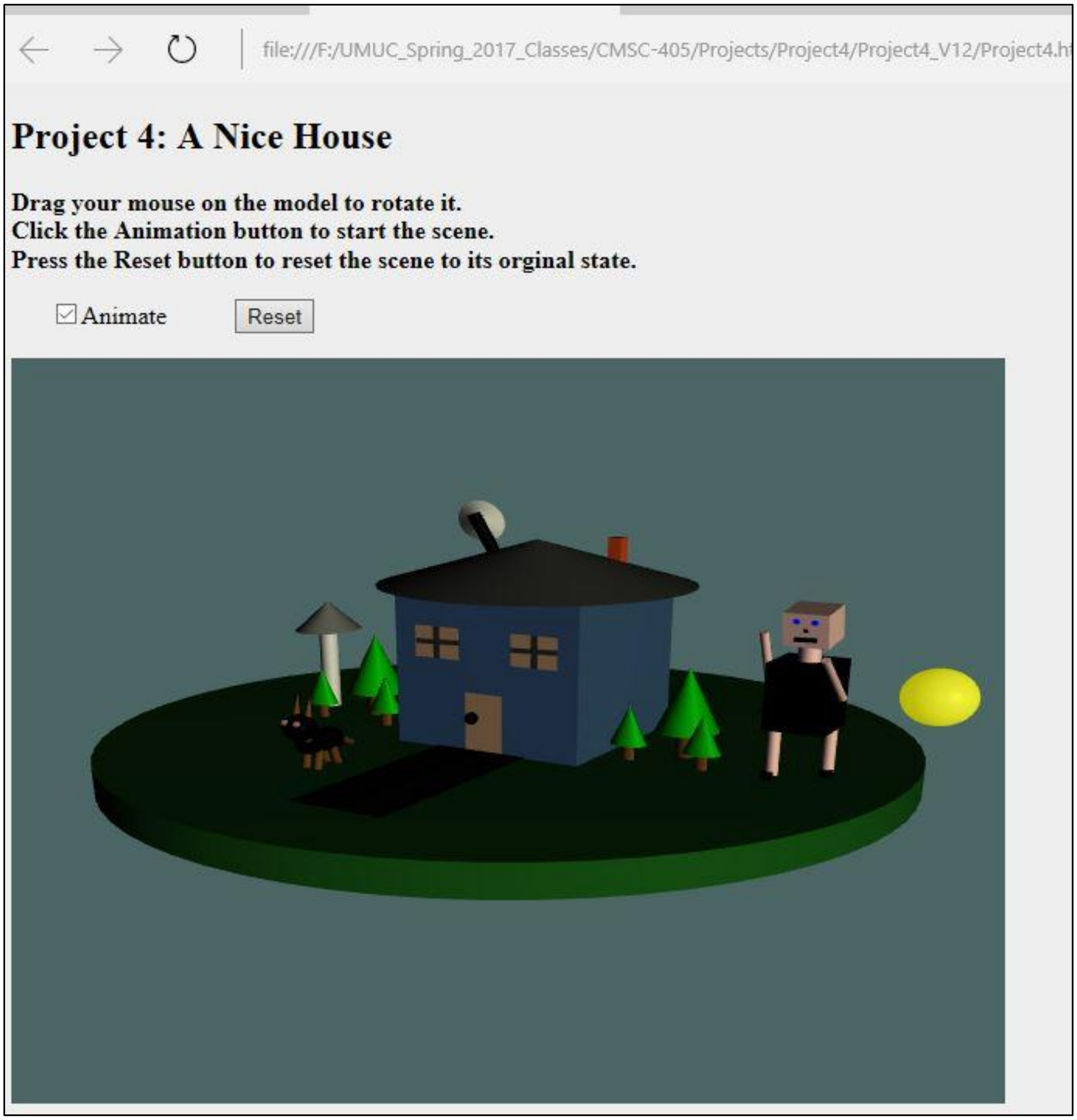
Drag your mouse on the model to rotate it.
Click the Animation button to start the scene.
Press the Reset button to reset the scene to its orginal state.



file:///F:/UMUC_Spring_2017_Classes/CMSC-405/Projects/Project4/Project4_V12/Projec

Project 4: A Nice House

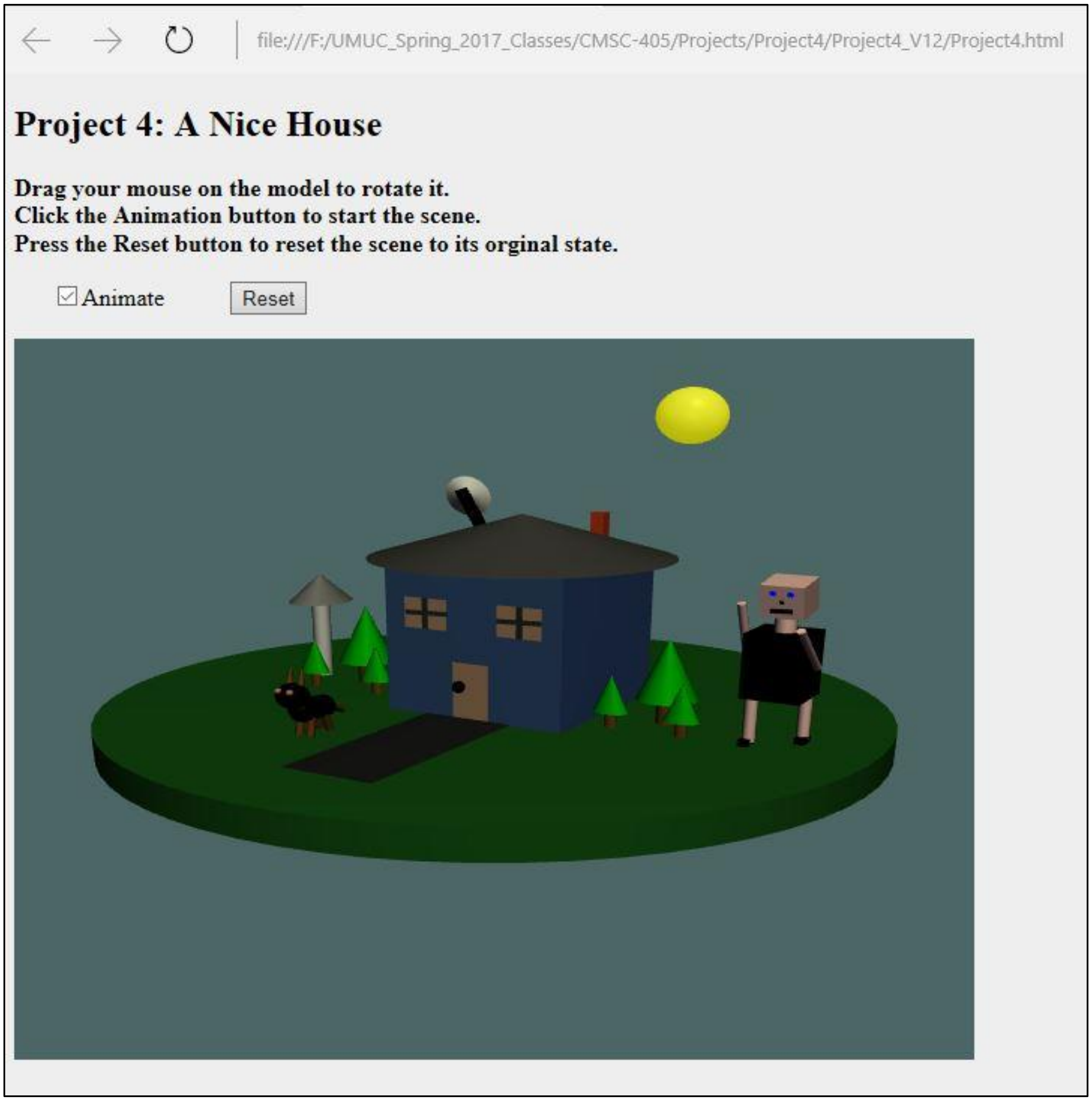
Drag your mouse on the model to rotate it.
Click the Animation button to start the scene.
Press the Reset button to reset the scene to its orginal state.



file:///F:/UMUC_Spring_2017_Classes/CMSC-405/Projects/Project4/Project4_V12/Project4.h

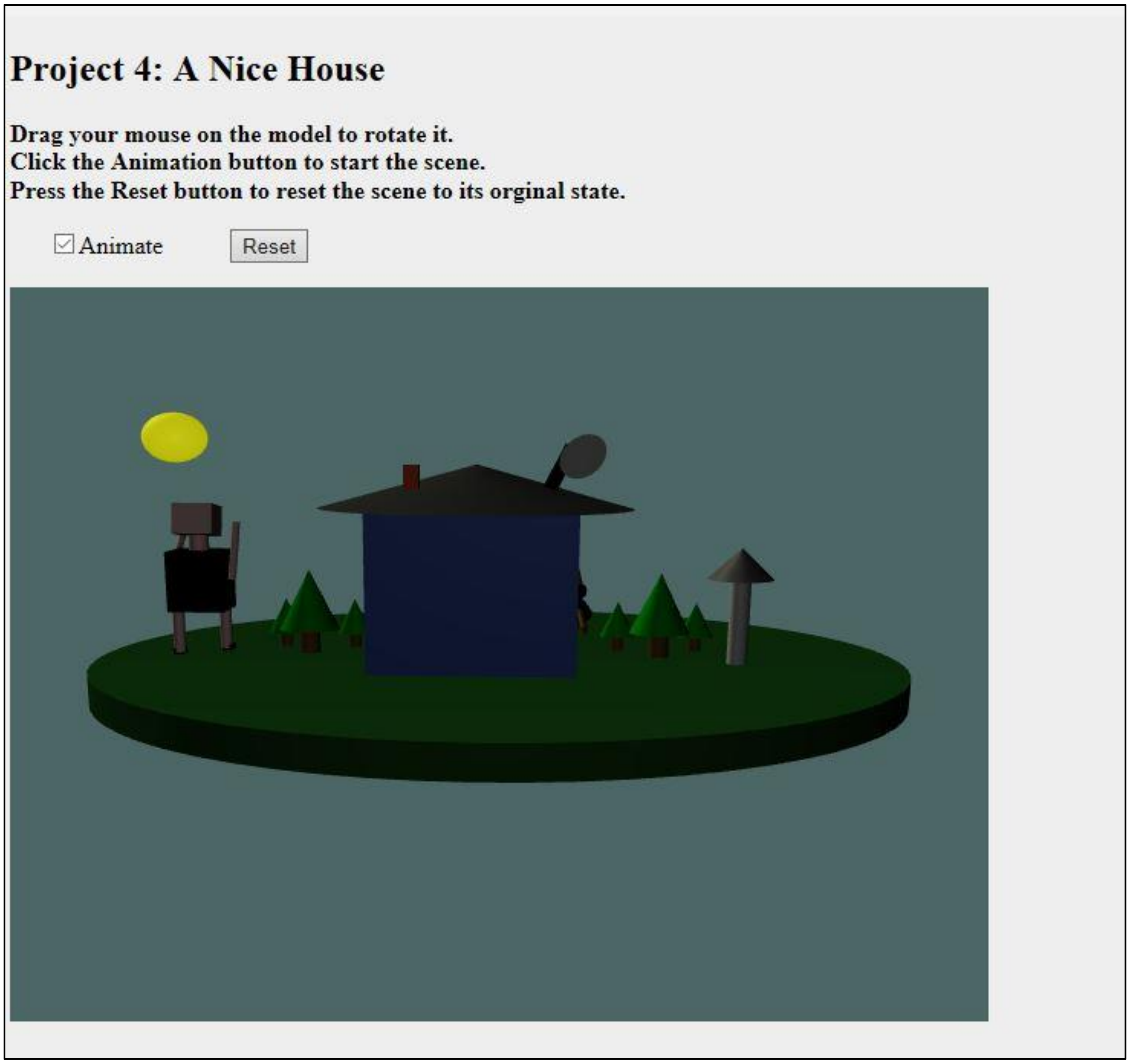
Project 4: A Nice House

Drag your mouse on the model to rotate it.
Click the Animation button to start the scene.
Press the Reset button to reset the scene to its orginal state.


file:///F:/UMUC_Spring_2017_Classes/CMSC-405/Projects/Project4/Project4_V12/Project4.html
☑ Animate
Reset

Project 4: A Nice House

Drag your mouse on the model to rotate it.
Click the Animation button to start the scene.
Press the Reset button to reset the scene to its orginal state.



Project 4: A Nice House

Drag your mouse on the model to rotate it.
Click the Animation button to start the scene.
Press the Reset button to reset the scene to its orginal state.

Lesson Learned:

After completing project 4, there are many things that I learned throughout the process of creating a 3D scene composed of WebGL graphic components. The first thing I did, like with all programing projects I am given, was to first read over the instructions, and understand what is being asked. This was my first time ever working with WebGL and its JavaScript libraries for rendering 3D graphics, so creating a 3D scene was challenging. However, the hardest part for me was creating textures for each object. I could not seem to get the images to load, and then display on the objects. As you will see in my code, I created the methods, but was unsuccessful with adding the images. Overall, I was able to successfully create a 3D scene composed of WebGL graphical components.