Anthony Borza

Project 2: JOGL OpenGL

CMSC 405 7980 Computer Graphics

University of Maryland University College

Due Date: April 9, 2017

Table of Contents

Brief Description of Project:

The purpose of this project was to use Java to create a 3D scene composed of OpenGL graphic components using transformation methods. The 3D scene had to meet the following requirements: the size of the scene must be 640 X 480, there must be at least six different shapes used as well as, six different transformation methods. The scene that I created includes: a house, two trees, grass, a car, and sun. I also decided to enable lighting in my scene. All objects are created in 3D, and use the methods and libraries offered by OpenGL. The Shapes that were used to create the 3D scene are the following:

- 3D lines for the sun
- Solid Sphere for the ball of the sun
- Solid Cone for the two trees, base of the house,
- Solid Cube for the base of the trees, and roof of the house, and chimney of the house.
- 3D lines for the two windows, and door on the house
- Solid cube and Solid Torus for the car.
- GL_QUADS used to draw the grass under, and around the house.

Design and Functionality:

Project2.java

- Extends JPanel and implements GLEventListener.
- Defines, variables, methods, and classes, to be used throughout Project2.java class.
- Created a "constructor method" to do the following:
    - Creates the layout of the canvas.

    - Specifies the size (640, 480).
    - Implements GLEventListener, and ActionEventListener.
    - Creates a JCheckBox that is used for animating the 3D scene.
    - Implements actionPerformed for the Checkbox, that is surrounded by an if, else statement that controls when to start the animation of the scene, and when to stop it.
    - Creates a new JPanel called "button," and centers the button using FlowLayout, and positions it at the bottom of the canvas.
    - Creates a actionPerformed method that is invoked when the button is selected or deselected.
    - Creates a new camera for the Camera.java class that was provided to us in the weekly readings.
    - Sets the scale of the camera to five.
    - Implements the TrackBall.java class that was provided to us in the weekly readings, and takes the parameters draw, and camera. This allows you to use the wheel on your mouse to move the 3D scene.
  - Created a "display method" to do the following:
    - To be used when OpenGL needs to be redrawn.
    - Applies camera, and lighting for the scene.
    - Calls the Shapes.java class, and the methods in that class to draw the 3D objects.
  - Created a "init method" to do the following:
    - Used to apply the projection, and color of the canvas.

- Created a “reshape method” to do the following:
  - Used when the size of the GLJPanel changes. This method however, contains no code in it, as it is not necessary.
- Created a “dispose method” to do the following:
  - Method is used before the GLJPanel is destroyed. This method however, contains no code in it, as it is not necessary.
- Create a “main method” to do the following:
  - Used to run the program.
  - Used to label the frame window.
  - Used to set the content pane, and visibility.

Shapes.java

- Created a “house method” to do the following:
  - Uses methods offered by OpenGL “JOGL” to draw the house.
  - Uses a transformation methods to include: Translate, Rotate, and Scale.
  - First draws “grass” around the house
    - Implements GL_QUADS, which requires a group of four vertices, containing 3 numbers per group. This is used to position the grass, and determine its size.
  - Second draws the “bottom” of the house
    - Uses a feature offered by “JOGL” called GLUT to construct a solid cube to represent the bottom of the house.
  - Third draws the “roof” of the house

        - Uses a feature offered by "JOGL" called GLUT to construct a solid cone to represent the roof of the house.
        - Uses transformation methods: Translate, and Rotate.
    - Fourth draws the "chimney" of the house
        - Uses a feature offered by "JOGL" called GLUT to construct a solid cone to represent the roof of the house.
        - Uses transformation methods: Translate, and Scale.
    - Fifth draws the "sun" behind the house
        - Calls the sun method to draw the sun.
- Created a "door method" to do the following:
    - Uses methods offered by OpenGL "JOGL" to draw the door.
    - Implements GL_LINE_STRIP that is used to connect a group of line segments from the first vertex to the last.
    - Creates 5 groups of vertex's that are used to position, and size the door.
    - Used the translate transformation method.
- Created a "window1 method" to do the following:
    - Uses methods offered by OpenGL "JOGL" to draw the first window.
    - Implements GL_LINE_STRIP that is used to connect a group of line segments from the first vertex to the last.
    - Creates 5 groups of vertex's that are used to position, and size the window.
    - Used the translate transformation method.
- Created a "window2 method" to do the following:
    - Uses methods offered by OpenGL "JOGL" to draw the second window.

        - Implements GL_LINE_STRIP that is used to connect a group of line segments from the first vertex to the last.
        - Creates 5 groups of vertex's that are used to position, and size the window.
        - Used the translate transformation method.
    - Created a "tree method" to do the following:
        - First draws the "base" of the tree
            - Uses methods offered by OpenGL "JOGL" to draw the first tree.
            - Uses a feature offered by "JOGL" called GLUT to construct a solid cube to represent the base of the tree.
        - Second draws the "top" of the tree
            - Uses a feature offered by "JOGL" called GLUT to construct a solid cone to represent the top of the tree.
            - Uses transformation methods: Translate, Rotate, and Scale.
    - Created a "tree2 method" to do the following:
        - First draws the "base" of the tree
            - Uses methods offered by OpenGL "JOGL" to draw the second tree.
            - Uses a feature offered by "JOGL" called GLUT to construct a solid cube to represent the base of the tree.
        - Second draws the "top" of the tree
            - Uses a feature offered by "JOGL" called GLUT to construct a solid cone to represent the top of the tree.
            - Uses transformation methods: Translate, Rotate, and Scale.
    - Created a "car method" to do the following:

- First draws the “body” of the car
    - Uses methods offered by OpenGL “JOGL” to draw the car.
    - Uses a feature offered by “JOGL” called GLUT to construct a solid cube to represent the body of the car.
    - Used transformation method: Translate
- Second draws the “first wheel” of the car
    - Uses methods offered by OpenGL “JOGL” to draw the first wheel.
    - Uses a feature offered by “JOGL” called GLUT to construct a solid torus to represent the first wheel of the car.
    - Used transformation method: Translate
- Third draws the “second wheel” of the car
    - Uses methods offered by OpenGL “JOGL” to draw the second wheel.
    - Uses a feature offered by “JOGL” called GLUT to construct a solid torus to represent the second wheel of the car.
    - Used transformation method: Translate
- Fourth draws the “third wheel” of the car
    - Uses methods offered by OpenGL “JOGL” to draw the third wheel.
    - Uses a feature offered by “JOGL” called GLUT to construct a solid torus to represent the third wheel of the car.
    - Used transformation method: Translate
- Fifth draws the “fourth wheel” of the car
    - Uses methods offered by OpenGL “JOGL” to draw the fourth wheel.

        - Uses a feature offered by "JOGL" called GLUT to construct a solid torus to represent the fourth wheel of the car.
        - Used transformation method: Translate
  - Created a "sun method" to do the following:
    - Implements a for loop to create the 13 rays around, or lines around the sun.
    - Calls the sphere method
  - Created a "sphere method" to do the following:
    - Uses methods offered by OpenGL "JOGL" to draw the circle around the sun rays.
    - Uses a feature offered by "JOGL" called GLUT to construct a solid sphere to represent the circle around the rays.

Camera.java

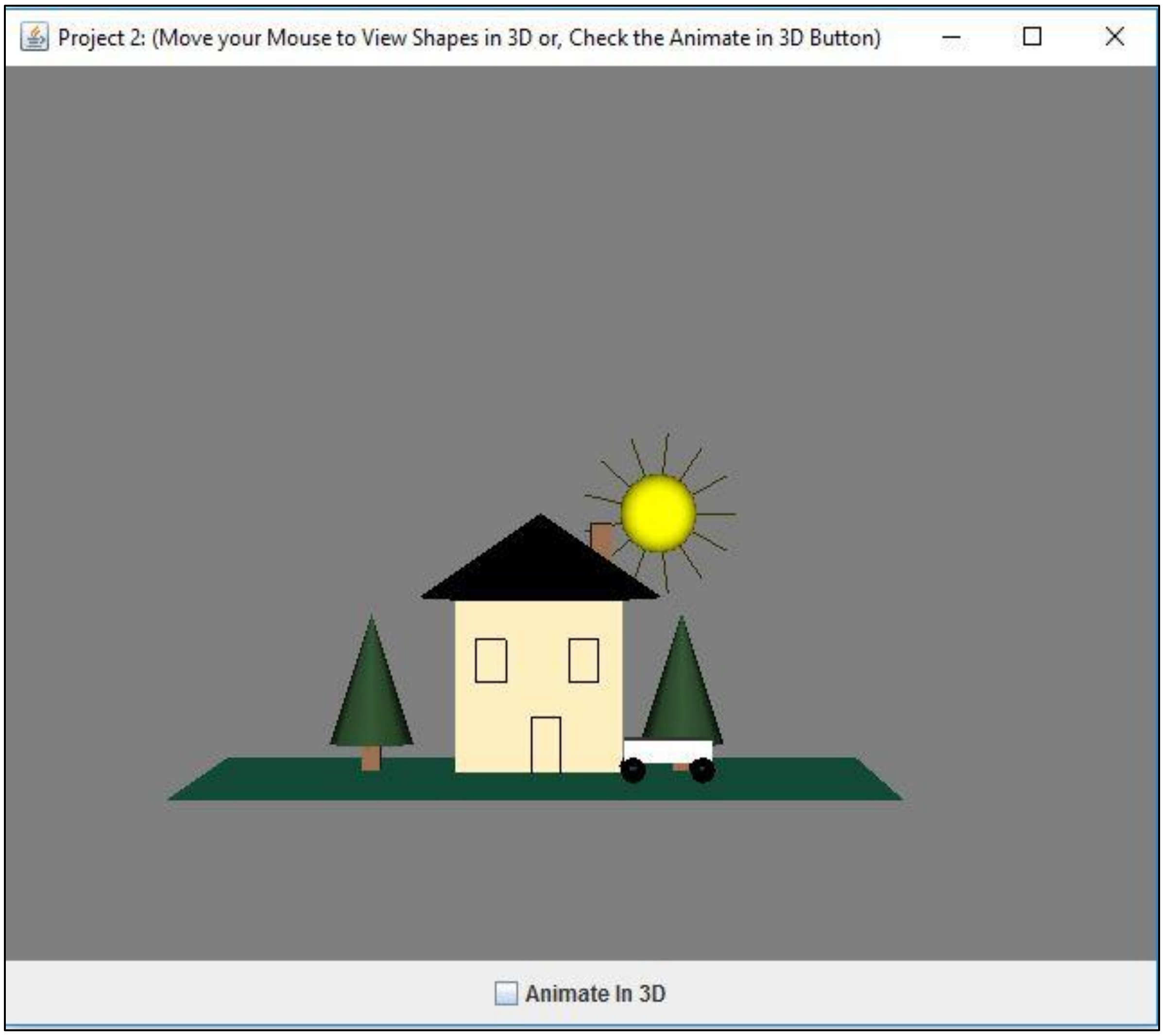
  - This class is provided in the readings, and was used to encapsulate the information needed to define a viewing transform and projection for the 3D scene. Retrieved from: http://math.hws.edu/graphicsbook/source/jogl/Camera.java

TrackBall.java

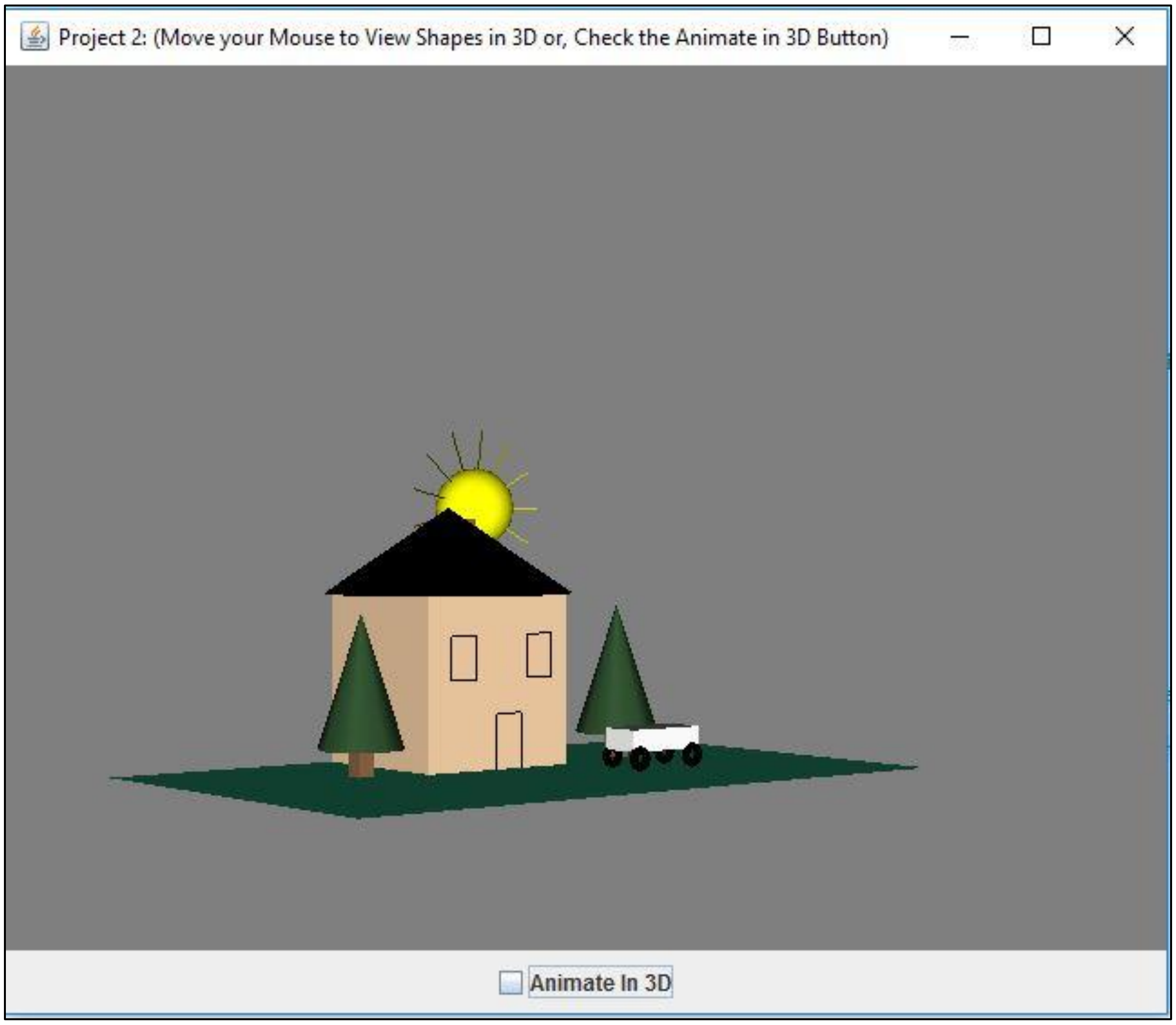
  - This class is provided in the readings, and was used to allow the user to use their mouse to rotate and view the 3D scene. Retrieved from: http://math.hws.edu/eck/cs424/s10/lab7/graph3d/TrackBall.java

Screenshots:

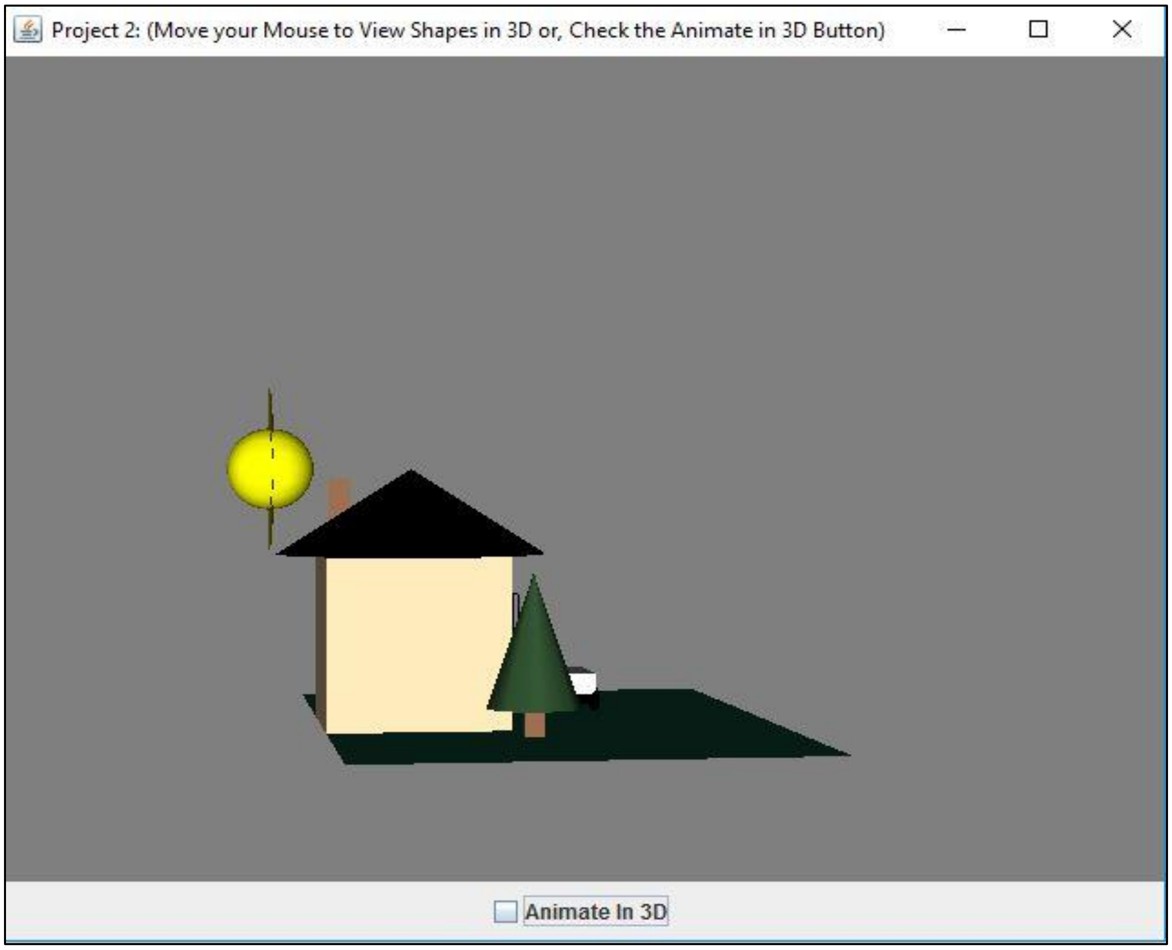
First Image: Standard View of the House


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

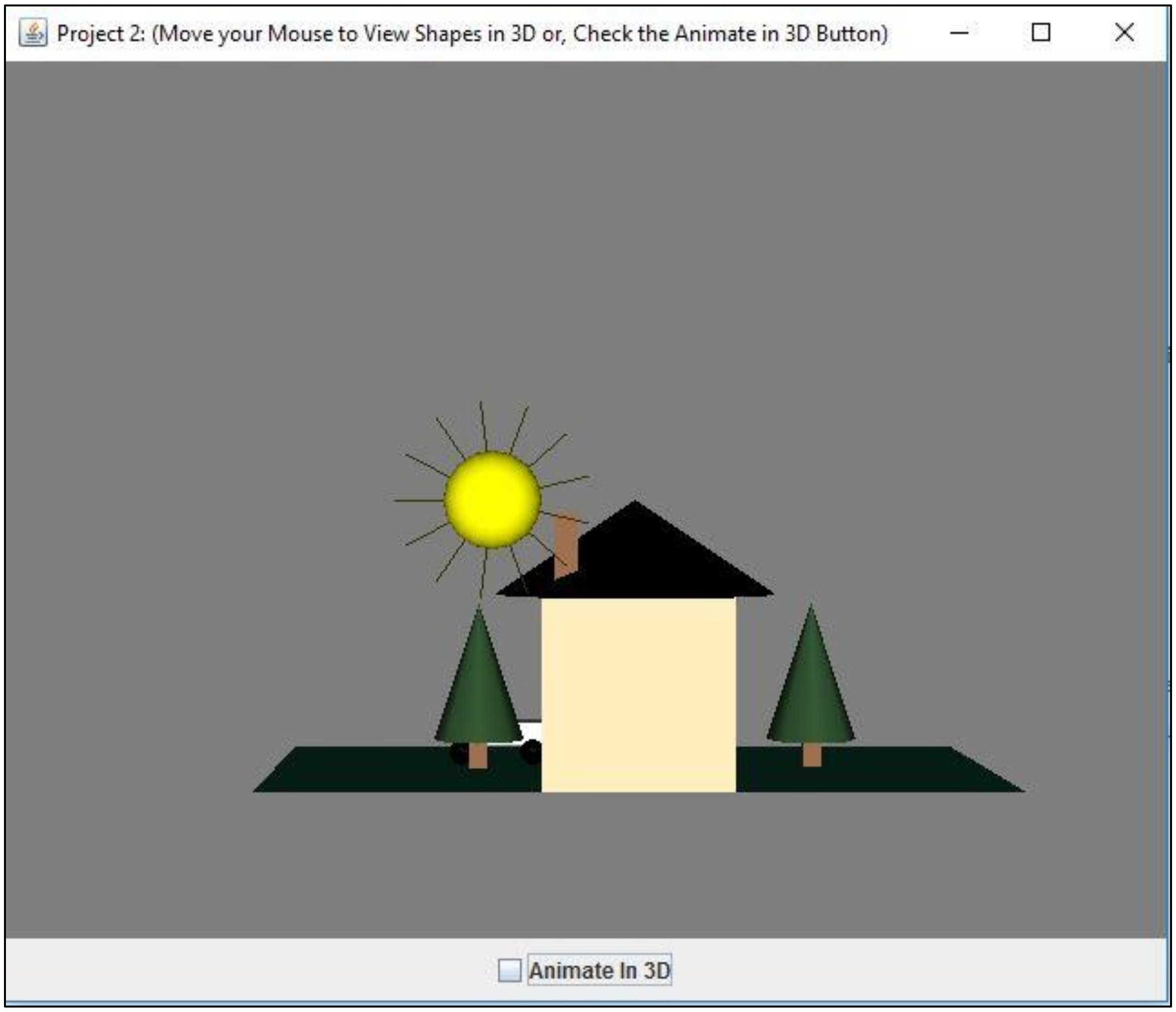
Second Image: Angle View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

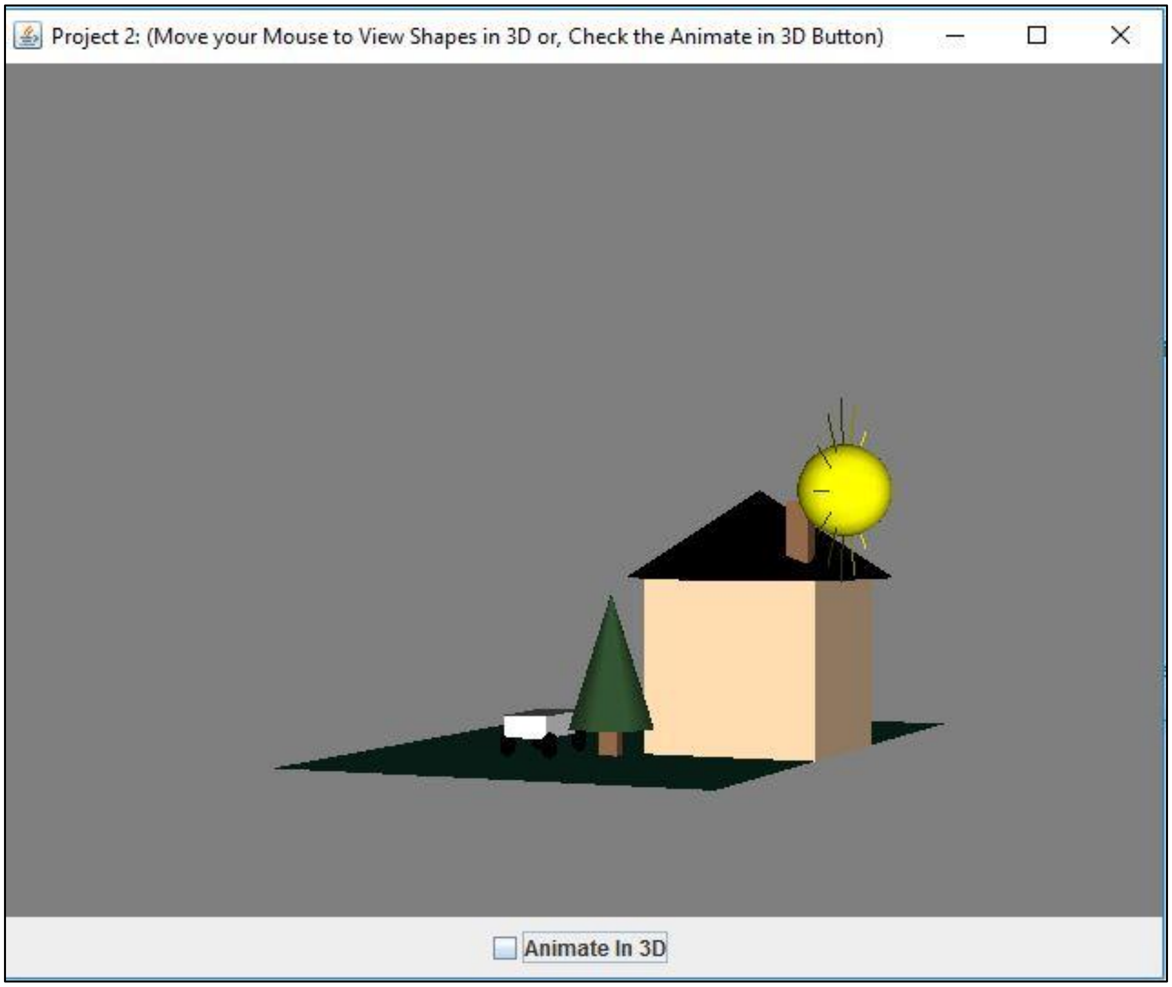
Third Image: Side View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

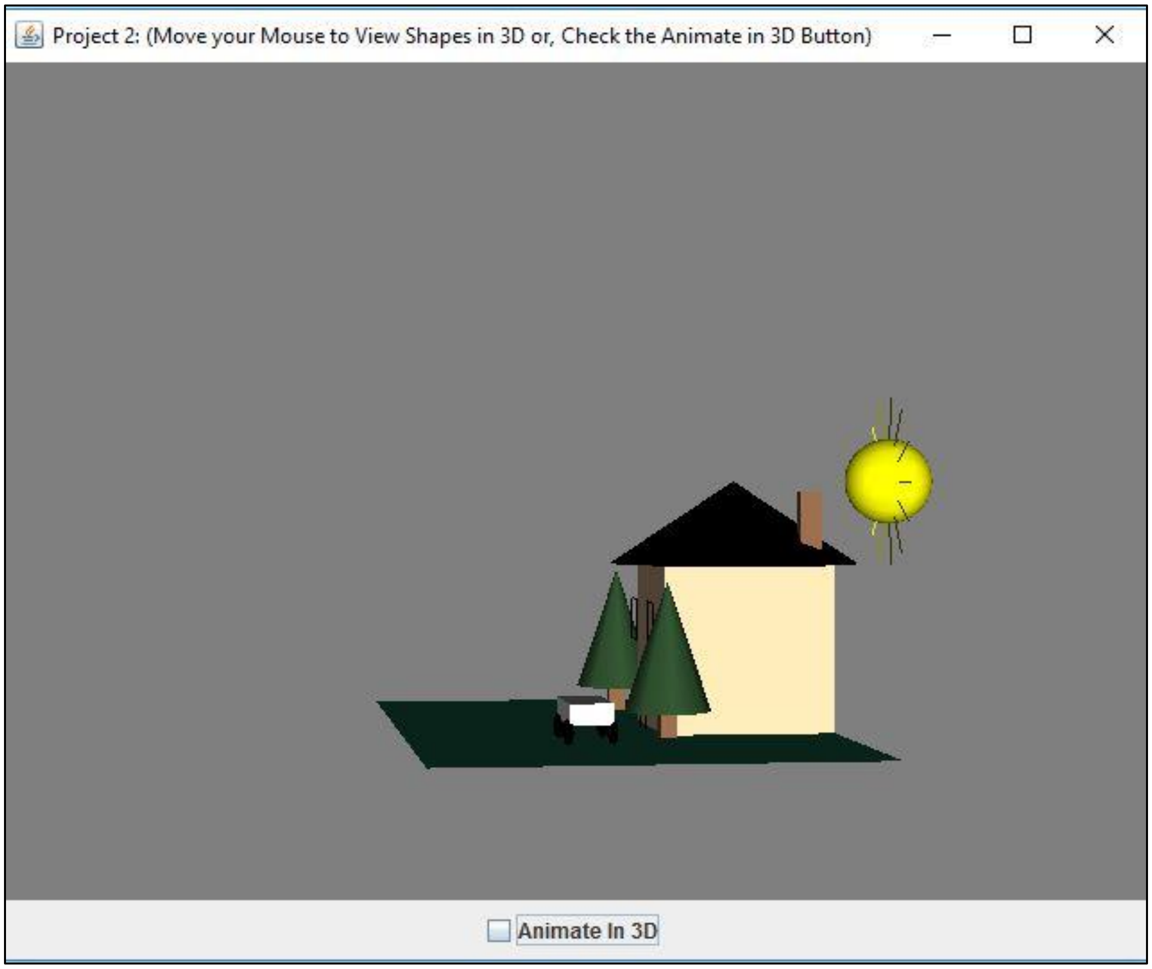
Fourth Image: Back View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

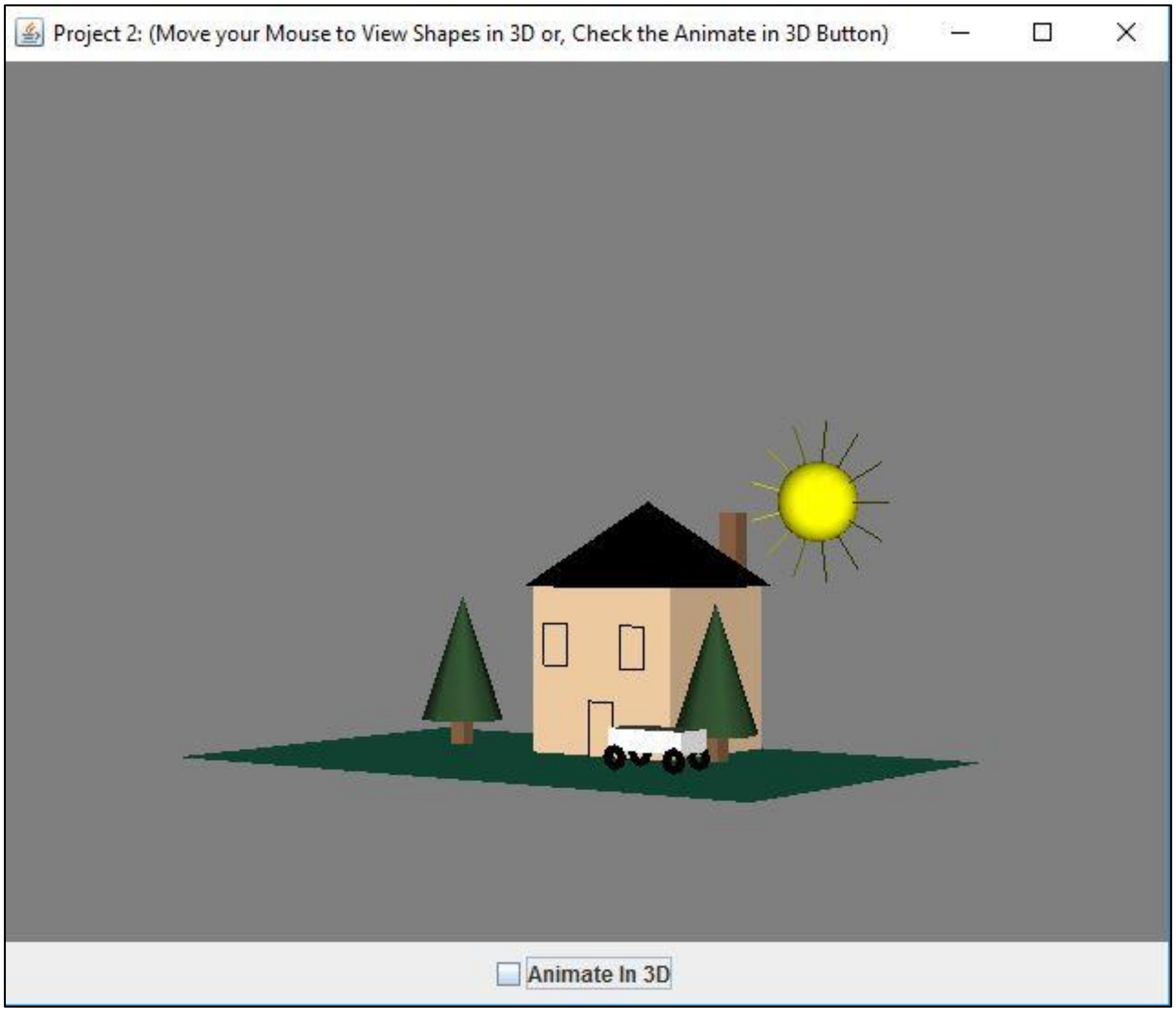
Fifth Image: Side Angle View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

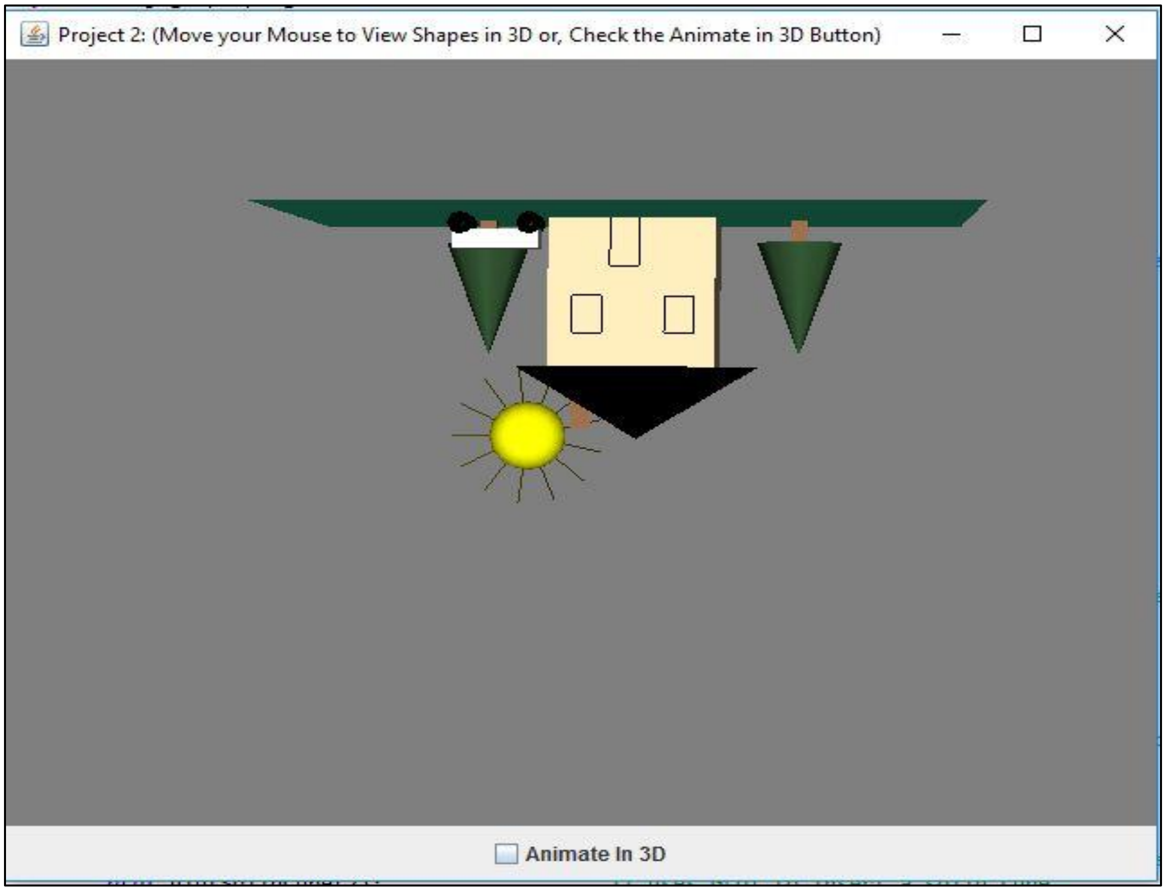
Sixth Image: Back View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

Seventh Image: Angle View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

Eight Image: Upside Down View of the House in 3D


Project 2: (Move your Mouse to View Shapes in 3D or, Check the Animate in 3D Button)
Animate In 3D

Lessons Learned:

After completing project 2, there are many things that I learned throughout the process of creating a 3D scene composed of OpenGL 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 working with the OpenGL libraries offered by java, so it took some time to understand the new functions, and methods under it. The hardest part was learning the new functions and methods, as well as, self-teaching myself some linear algebra, and how vectors, vertices, and other vector math works with programming. Once I could create the first 3D object, the easiest part was creating all of the remaining objects in the 3D scene.

Overall, this was a fun project, and I am satisfied with my program, and look forward to the next one.

References:

Eck David J. (2016). *Introduction to Computer Graphics: Camera.java Class.* Retrieved from http://math.hws.edu/graphicsbook/source/jogl/Camera.java

Eck David J. (2016). *Introduction to Computer Graphics: TackBall.java Class.* Retrieved from http://math.hws.edu/eck/cs424/s10/lab7/graph3d/TrackBall.java