/\*\*

\* This is now going to be my HousenTable

\* Jonny Hughes

\*/

import java.awt.\*;

import javax.swing.\*;

import com.jogamp.opengl.\*;

import com.jogamp.opengl.awt.\*;

import com.jogamp.opengl.util.FPSAnimator;

import java.net.URL;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import javax.imageio.ImageIO;

import com.jogamp.opengl.awt.GLJPanel;

import com.jogamp.opengl.util.awt.ImageUtil;

import com.jogamp.opengl.util.gl2.GLUT;

import com.jogamp.opengl.util.texture.Texture;

import com.jogamp.opengl.util.texture.awt.AWTTextureIO;

public class HousenTable extends GLJPanel implements GLEventListener{

private int frameNumber = 0; // The current frame number for an animation.

private Camera camera;

private GLUT glut = new GLUT();

public float[] Tmaterial = { 0.44725F, 0.2995F, 0.2045F, 1.0F, // ambient

0.45164F, 0.27648F, 0.02648F, 1.0F, // diffuse

0.428281F, 0.25802F, 0.266065F, 1.0F, // specular

16.0F // shininess

};

private Texture[] textures = new Texture[1];

/\*\*

\* A main routine to create and show a window that contains a

\* panel of type HousenTable The program ends when the

\* user closes the window.

\*/

public static void main(String[] args) {

JFrame window = new JFrame("Hey! Take a look at this HousenTable");

HousenTable panel = new HousenTable();

window.setContentPane((Container) panel);

window.pack();

window.setLocation(50,50);

window.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

window.setVisible(true);

}

/\*\*

\* Constructor for class HousenTable.

\*/

public HousenTable() {

super( new GLCapabilities(null) ); // Makes a panel with default OpenGL "capabilities".

setPreferredSize( new Dimension(500,500) );

addGLEventListener(this); // A listener is essential! The listener is where the OpenGL programming lives.

}

//-------------------- GLEventListener Methods -------------------------

/\*\*

\* The display method is called when the panel needs to be redrawn.

\* The is where the code goes for drawing the image, using OpenGL commands.

\*/

public void display(GLAutoDrawable drawable) {

GL2 gl = drawable.getGL().getGL2(); // The object that contains all the OpenGL methods.

camera.apply(gl);

gl.glClearColor( 0, 0, 0, 1 ); // (In fact, this is the default.)

gl.glClear( GL2.GL\_COLOR\_BUFFER\_BIT | GL2.GL\_DEPTH\_BUFFER\_BIT);

//These are some custom objects which are used to define shapes to draw

Floor floor = new Floor();

HouseBox house = new HouseBox();

/\*\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

This is the main drawing part

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\*/

//Here is the HousenTable

//Light

float[] ambientLight = { 1f, 0.f, 1.f, 0f }; // weak RED ambient

gl.glLightfv(GL2.GL\_LIGHT1, GL2.GL\_AMBIENT, ambientLight, 0);

float[] diffuseLight = { 1f,2f,1f,0f }; // multicolor diffuse

gl.glLightfv(GL2.GL\_LIGHT1, GL2.GL\_DIFFUSE, diffuseLight, 0);

float[] positionLight = {0f, 0.8f, 0f, 0f};

gl.glLightfv(GL2.GL\_LIGHT1, GL2.GL\_POSITION, positionLight, 0);

// The ground

gl.glPushMatrix();

gl.glTranslatef(0f, -0.525f, 0f);

gl.glScalef(1.25f, 1f, 1.25f);

floor.drawFloor(gl);

gl.glPopMatrix();

// House

gl.glPushMatrix();

house.drawHouse(gl);

gl.glPopMatrix();

//Table

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_AMBIENT, Tmaterial, 0 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_DIFFUSE, Tmaterial, 4 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SPECULAR, Tmaterial, 8 );

gl.glMaterialf( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SHININESS, Tmaterial[12] );

gl.glPushMatrix();

gl.glRotatef(90, 1f, 0f, 0f);

gl.glPushMatrix();

gl.glTranslatef(0f, 0f, 0.25f);

glut.glutSolidCylinder(0.03d, 0.2d, 20, 20);

gl.glPopMatrix();

gl.glPushMatrix();

gl.glTranslatef(0f, 0f, 0.25f);

glut.glutSolidCylinder(0.2d, 0.02d, 20, 20);

gl.glPopMatrix();

gl.glPopMatrix();

//The hidden Gem

gl.glPushMatrix();

gl.glTranslatef(0.0f, -0.2f, 0.0f);

gl.glRotatef(60, 1f, 0f, 1f);

gl.glScalef(0.1f, 0.1f, 0.1f);

BiPy thegem = new BiPy();

thegem.drawBiPy(gl);

gl.glPopMatrix();

} // end display()

public void init(GLAutoDrawable drawable) {

// called when the panel is created

camera = new Camera(); //This is the alternate camera

camera.lookAt(5,10,30, 0,0,0, 0,1,0);

camera.setScale(1.2);

camera.installTrackball(this);

//This code is from the lights around the teapot example in the book

GL2 gl = drawable.getGL().getGL2();

gl.glClearColor(0, 0, 0, 1);

gl.glEnable(GL2.GL\_DEPTH\_TEST);

gl.glEnable(GL2.GL\_LIGHTING);

gl.glEnable(GL2.GL\_LIGHT0);

gl.glEnable(GL2.GL\_LIGHT1);

gl.glEnable(GL2.GL\_NORMALIZE);

gl.glLightModeli(GL2.GL\_LIGHT\_MODEL\_TWO\_SIDE, 1);

gl.glLightModeli(GL2.GL\_LIGHT\_MODEL\_LOCAL\_VIEWER, 1);

gl.glMateriali(GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SHININESS, 64);

}

public void dispose(GLAutoDrawable drawable) {

// called when the panel is being disposed

}

public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {

// called when user resizes the window

}

}

HouseBox.java:

import java.awt.\*;

import javax.swing.\*;

import com.jogamp.opengl.\*;

import com.jogamp.opengl.awt.\*;

import com.jogamp.opengl.util.FPSAnimator;

import com.jogamp.opengl.util.awt.ImageUtil;

import com.jogamp.opengl.util.gl2.GLUT;

import com.jogamp.opengl.util.texture.Texture;

import com.jogamp.opengl.util.texture.awt.AWTTextureIO;

public class HouseBox{

public float[] material = {

0.14725F, 0.8995F, 0.2045F, 1.0F, // ambient

0.15164F, 0.87648F, 0.02648F, 1.0F, // diffuse

0.128281F, 0.855802F, 0.266065F, 1.0F, // specular

16.0F // shininess

};

public float[][] vertexList =

{ {0.5f, 0.5f, 0.5f}, //UpRight Front 0

{-0.5f, 0.5f, 0.5f}, //UpLeft Front 1

{-0.5f, -0.5f, 0.5f}, //DownLeft Front 2

{0.5f, -0.5f, 0.5f}, //DownRight Front 3

{0.5f, 0.5f, -0.5f}, //UpRight Back 4

{-0.5f, 0.5f, -0.5f}, //UpLeft Back 5

{-0.5f, -0.5f, -0.5f}, //DownRight Back 6

{0.5f, -0.5f, -0.5f}, //DownLeft Back 7

{0f, 1f, 0.5f}, //Top Front Roof 8

{0f, 1f, -0.5f}, //Top Back Roof 9

{-0.25f, 0.685f, 0.5f}, //UpMFront Roof 10

{-0.25f, 0.55f, 0.5f}, //DownMFront ROof 11

{-0.25f, 0.685f, -0.5f}, //TopM Back Roof 12

{-0.25f, 0.55f, -0.5f}, //BottmM Back Roof 13

{-0.25f, -0.5f, 0.5f}, //SlicePtL 14

{-0.25f, -0.25f, 0.5f}, //BLW 15

{-0.25f, 0.3f, 0.5f}, //TLW 16

{0.25f, 0.3f, 0.5f}, //TRW 17

{0.25f, -0.25f, 0.5f}, //BRW 18

{-0.5f, 0f, 0.5f},//MUL 19

{0.5f, 0f, 0.5f},//MUR 20

{0.25f, -0.5f, 0.5f},//SlicePtR 21

{-0.25f, -0.5f, -0.5f}, //SlicePtL 22

{-0.25f, -0.25f, -0.5f}, //BLW 23

{-0.25f, 0.3f, -0.5f}, //TLW 24

{0.25f, 0.3f, -0.5f}, //TRW 25

{0.25f, -0.25f, -0.5f}, //BRW 26

{-0.5f, 0f, -0.5f},//MUL 27

{0.5f, 0f, -0.5f},//MUR 28

{0.25f, -0.5f, -0.5f},//SlicePtR 29

};

/\*public double[][][] texCoordList = {

{{1.0, 0.8}, {0.5, 1.0}, {0.0, 0.8}, {0.0, 0.0}, {1.0, 0.0}}, //front

{{1.0, 0.8}, {0.5, 1.0}, {0.0, 0.8}, {0.0, 0.0}, {1.0, 0.0}}, //back

{{1.0, 1.0}, {0.0, 1.0}, {0.0, 0.0}, {1.0, 0.0}}, //right

{{0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}}, //bottom

{{1.0, 1.0}, {0.0, 1.0}, {0.0, 0.0}, {1.0, 0.0}}, //left

{{0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}}, //roof right

{{0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}, {0.0, 0.0}}, //roof left},

};\*/

public int[][] faceList =

{ {0,8,/\*10,11,\*/1,19,20}, //front

{19,2,14,16},

{17,21,3,20},

{14,21,18,15},

{5,/\*13,12,\*/9,4,28,27}, //back

{27,6,22,24},

{25,29,7,28},

{22,29,26,23},

{4,0,3,7}, //right

{7,3,2,6}, //bottom

{1,5,6,2}, //left

{4,9,8,0}, //roof right

{8,9,5,1}, //roof left1

/\*{10,12,13,11}, //roof left2

{11,13,4,1}, \*/ //roof left3

/\*THE INVERSES FOR THE INSIDE OF THE HOUSE

{20,19,1,8,0}, //front

{16,14,2,19},

{20,3,21,17},

{15,18,21,14},

{6,7,4,9,5}, //back

{7,3,0,4}, //right

{6,2,3,7}, //bottom

{2,6,5,1}, //left

{0,8,9,4}, //roof right

{1,5,9,8}, //roof left1

/\*{10,12,13,11}, //roof left2

{11,13,4,1}, \*/ //roof left3\*/

};

public float []faceColor = {0.902f, 0.075f, 0.0f};

public float Vx;

public float Vy;

public float Vz;

public float Wx;

public float Wy;

public float Wz;

public void drawHouse(GL2 gl/\*, Texture texture\*/){

//box

for (int i = 0; i < faceList.length; i++) {

Vx = vertexList[faceList[i][1]][0] - vertexList[faceList[i][0]][0];

Vy = vertexList[faceList[i][1]][1] - vertexList[faceList[i][0]][1];

Vz = vertexList[faceList[i][1]][2] - vertexList[faceList[i][0]][2];

Wx = vertexList[faceList[i][2]][0] - vertexList[faceList[i][0]][0];

Wy = vertexList[faceList[i][2]][1] - vertexList[faceList[i][0]][1];

Wz = vertexList[faceList[i][2]][2] - vertexList[faceList[i][0]][2];

gl.glBegin(GL2.GL\_POLYGON);

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_AMBIENT, material, 0 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_DIFFUSE, material, 4 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SPECULAR, material, 8 );

gl.glMaterialf( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SHININESS, material[12] );

gl.glNormal3f(Vy\*Wz - Vz\*Wy, Vz\*Wx - Vx\*Wz, Vx\*Wy - Vy\*Wx);

for (int j = 0; j < faceList[i].length; j++) {

int vertexNum = faceList[i][j]; // Index for vertex j of face i.

float[] vertexCoords = vertexList[vertexNum]; // The vertex itself.

gl.glVertex3f(vertexCoords[0], vertexCoords[1], vertexCoords[2]);

}

gl.glEnd();

}

}

}

BiPy.java:

import java.awt.\*;

import javax.swing.\*;

import com.jogamp.opengl.\*;

import com.jogamp.opengl.awt.\*;

import com.jogamp.opengl.util.FPSAnimator;

public class BiPy{

public float[] material = { 0.94725F, 0.0995F, 0.0045F, 1.0F, // ambient

0.95164F, 0.07648F, 0.02648F, 1.0F, // diffuse

0.728281F, 0.655802F, 0.666065F, 1.0F, // specular

16F, // shininess

0.2f,

};

public float[][] vertexList =

{ {0.0f, 1.0f, 0.0f}, //0

{-0.25f, 0.0f, 0.50f}, //1

{0.25f, 0.0f, 0.50f}, //2

{0.57f, 0.0f, 0.0f}, //3

{0.25f, 0.0f, -0.50f}, //4

{-0.25f, 0.0f, -0.50f}, //5

{-0.57f, 0.0f, 0.0f}, //6

{0.0f, -1.0f, 0.0f} //7

};

public int[][] faceList =

{ {0,1,2}, //0

{0,2,3}, //1

{0,3,4}, //2

{0,4,5}, //3

{0,5,6}, //4

{0,6,1}, //5

{7,2,1}, //6

{7,3,2}, //7

{7,4,3}, //8

{7,5,4}, //9

{7,6,5}, //10

{7,1,6} //11

};

public float []faceColor = {0.902f, 0.075f, 0.0f};

public float Vx;

public float Vy;

public float Vz;

public float Wx;

public float Wy;

public float Wz;

public void drawBiPy(GL2 gl){

for (int i = 0; i < faceList.length; i++) {

Vx = vertexList[faceList[i][1]][0] - vertexList[faceList[i][0]][0];

Vy = vertexList[faceList[i][1]][1] - vertexList[faceList[i][0]][1];

Vz = vertexList[faceList[i][1]][2] - vertexList[faceList[i][0]][2];

Wx = vertexList[faceList[i][2]][0] - vertexList[faceList[i][0]][0];

Wy = vertexList[faceList[i][2]][1] - vertexList[faceList[i][0]][1];

Wz = vertexList[faceList[i][2]][2] - vertexList[faceList[i][0]][2];

gl.glBegin(GL2.GL\_TRIANGLES);

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_AMBIENT, material, 0 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_DIFFUSE, material, 4 );

gl.glMaterialfv( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SPECULAR, material, 8 );

gl.glMaterialf( GL2.GL\_FRONT\_AND\_BACK, GL2.GL\_SHININESS, material[12] );

gl.glNormal3f(Vy\*Wz - Vz\*Wy, Vz\*Wx - Vx\*Wz, Vx\*Wy - Vy\*Wx);

for (int j = 0; j < faceList[i].length; j++) {

int vertexNum = faceList[i][j]; // Index for vertex j of face i.

float[] vertexCoords = vertexList[vertexNum]; // The vertex itself.

gl.glVertex3f(vertexCoords[0], vertexCoords[1], vertexCoords[2]);

}

}

gl.glEnd();

}

}