**CSE 4200 Lab 4 - Spencer Wallace**

\*For the code only able the lines that I changed are included

**Summary: (Please see photos and code below)**

All parts completed, to my knowledge, successfully because of this I am giving myself full credit – 30/30. \***Lines discussed below.** For the polygons I struggled a while with the 9 sided polygon, I could not get it to draw all of the vertices, I think because I was accidentally drawing a convex polygon. I was able to fix this and produce a 12 sided polygon which I then scaled down to 9 sides as seen below. For the cube I estimated some points using some guesses from what I know about how a cube looks, and for the dotted lines I used 0x9999 to draw constantly alternating points. I think it looks pretty good. This same method for the cube was used for the hut, but doubled in scale – and it was also used for the chimney as well as the door. For the window I used the circle algorithm from our previous lab and drew circles with a decreasing radius but same central point until it was filled. For the roof I used four triangles which I again estimated for location and size, and I also think these turned out pretty good and that the hut was pretty colorful.

**Lines.cpp**

Graphical user interface

Description automatically generated with medium confidence

For the top lines I used 0xFFFF to draw a solid line, and then 0xF99F for the dot in between since this is able to connect with the other lines but still represent a dot.

**1111 1111 1111 1111** **1111 1001 1001 1111** **1111 1111 1111 1111**

For the bottom I used 0x4A4A as this gives exactly two spaces between the second of the group closed together and the distant dot, and then two spaces between the distant dot and the beginning of the next group.

**Code for lines.cpp**

glColor3f (1.0, 1.0, 1.0);

glEnable (GL\_LINE\_STIPPLE);

glLineStipple (1, 0xFFFF); // dash

drawOneLine (50.0, 125.0, 100.0, 125.0);

glLineStipple (1, 0xF99F); // dot

drawOneLine (100.0, 125.0, 116.0, 125.0);

glLineStipple (1, 0xFFFF); // dash

drawOneLine (116.0, 125.0, 156.0, 125.0);

glLineStipple (1, 0xF99F); // dot

drawOneLine (156.0, 125.0, 172.0, 125.0);

glLineStipple (1, 0xFFFF); // dash

drawOneLine (172.0, 125.0, 222.0, 125.0);

glLineWidth(5.0);

glLineStipple (5, 0x4A4A); // close then distant

drawOneLine (50.0, 75.0, 210, 75.0);

**Polygon.cpp**

A yellow square with a black background

Description automatically generated with low confidence

**Draw Mode with fill**

A screenshot of a computer

Description automatically generated with low confidence

**Draw Mode with point**Shape

Description automatically generated

**Draw Mode with lines**

Shape

Description automatically generated

**Green polygon is drawn clockwise, set to cull front**

A picture containing icon

Description automatically generated

**Yellow polygon is drawn counter clockwise, set to cull back**

**Code**

glClear (GL\_COLOR\_BUFFER\_BIT);

glEnable(GL\_CULL\_FACE);

glCullFace(GL\_BACK);

glPointSize(10.0);

//green poly

glColor3f (0.0, 1.0, 0.0);

glPolygonMode(GL\_FRONT, GL\_FILL);

// glFrontFace(GL\_CW);

glBegin( GL\_POLYGON );

glVertex2i( 140, 400 );

glVertex2i( 240, 350 );

glVertex2i( 240, 250 );

glVertex2i( 140, 200 );

glVertex2i( 40, 250 );

glVertex2i( 40, 350 );

glEnd();

//yellow poly

glColor3f (1.0, 1.0, 0.0);

glPolygonMode(GL\_FRONT, GL\_FILL);

// glFrontFace(GL\_CCW);

glBegin( GL\_POLYGON );

glVertex2i( 450, 400 );

glVertex2i( 375, 360 );

glVertex2i( 360, 320 );

glVertex2i( 370, 280 );

glVertex2i( 415, 240 );

glVertex2i( 485, 240 );

glVertex2i( 520, 280 );

glVertex2i( 530, 320 );

glVertex2i( 515, 360 );

glEnd();

glFlush ();

**cube.cpp**

Shape, rectangle

Description automatically generated

**Code**

glColor3f (1.0, 1.0, 1.0);

glEnable (GL\_LINE\_STIPPLE);

//bottom back

glLineStipple (3, 0x9999);

drawOneLine (100.0, 50.0, 50.0, 75.0);

glLineStipple (3, 0x9999);

drawOneLine (50.0, 75.0, 150, 75.0);

glLineStipple (3, 0x9999);

drawOneLine (200.0, 50.0, 150.0, 75.0);

//front

glLineStipple (1, 0xFFFF);

drawOneLine (100.0, 50.0, 200.0, 50.0);

glLineStipple (1, 0xFFFF);

drawOneLine (100.0, 150.0, 200.0, 150.0);

glLineStipple (1, 0xFFFF);

drawOneLine (100.0, 150.0, 100.0, 50.0);

glLineStipple (1, 0xFFFF);

drawOneLine (200.0, 150.0, 200.0, 50.0);

//top back

glLineStipple (3, 0x9999);

drawOneLine (100.0, 150.0, 50.0, 175.0);

glLineStipple (3, 0x9999);

drawOneLine (50.0, 175.0, 150, 175.0);

glLineStipple (3, 0x9999);

drawOneLine (200.0, 150.0, 150.0, 175.0);

//back sides

glLineStipple (3, 0x9999);

drawOneLine (50.0, 175.0, 50, 75.0);

glLineStipple (3, 0x9999);

drawOneLine (150.0, 175.0, 150.0, 75.0);

glFlush ();

**hut.cpp**

Icon

Description automatically generated

**Code**

glPointSize(10.0);

glColor3f (0.0, 1.0, 0.0);

glPolygonMode(GL\_FRONT, GL\_FILL);

// front

glBegin( GL\_POLYGON );

glVertex2i(200, 50);

glVertex2i(400, 50);

glVertex2i(400, 250);

glVertex2i(200, 250);

glEnd();

//left

glColor3f (1.0, 0.0, 0.0);

glBegin( GL\_POLYGON );

glVertex2i(200, 50);

glVertex2i(100, 100);

glVertex2i(100, 300);

glVertex2i(200, 250);

glEnd();

//top

glColor3f (0.0, 0.8, 0.9);

glBegin( GL\_POLYGON );

glVertex2i(100, 300);

glVertex2i(200, 250);

glVertex2i(400, 250);

glVertex2i(300, 300);

glEnd();

//door

glColor3f (0.7, 0.4, 0.0);

glBegin( GL\_POLYGON );

glVertex2i(280, 100);

glVertex2i(280, 50);

glVertex2i(320, 50);

glVertex2i(320, 100);

glEnd();

//roof

////front

glBegin( GL\_POLYGON );

glVertex2i(270, 300);

glVertex2i(180, 250);

glVertex2i(420, 250);

glEnd();

//left

glColor3f(0.5, 0.3, 0.0);

glBegin( GL\_POLYGON );

glVertex2i(30, 310);

glVertex2i(180, 250);

glVertex2i(270, 300);

glEnd();

//back

glColor3f(0.9, 0.6, 0.0);

glBegin( GL\_POLYGON );

glVertex2i(30, 310);

glVertex2i(310, 310);

glVertex2i(270, 300);

glEnd();

//right

glColor3f(0.9, 0.8, 0.0);

glBegin( GL\_POLYGON );

glVertex2i(420, 250);

glVertex2i(310, 310);

glVertex2i(270, 300);

glEnd();

//chimney

glColor3f( 0.3, 0.25, 0.05 );

// front

glBegin( GL\_POLYGON );

glVertex2i(200, 280);

glVertex2i(240, 280);

glVertex2i(240, 340);

glVertex2i(200, 340);

glEnd();

//left

glBegin( GL\_POLYGON );

glVertex2i(180, 300);

glVertex2i(200, 280);

glVertex2i(200, 340);

glVertex2i(180, 360);

glEnd();

//top

glBegin( GL\_POLYGON );

glVertex2i(180, 360);

glVertex2i(200, 340);

glVertex2i(240, 340);

glVertex2i(220, 360);

glEnd();

//window

glColor3f (0.1, 1.0, 0.8);

Circle(300, 190, 30);

Circle(300, 190, 25);

Circle(300, 190, 20);

Circle(300, 190, 15);

Circle(300, 190, 10);

Circle(300, 190, 5);

glFlush ();