1. Nick Wright CSC328 Graphics Problem 3
2. Code

```
#include<windows.h>
#include<GL/glut.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>
#include<stdio.h>
#include <iostream>
#include <iomanip>
using namespace std;

/* Nick Wright
This is a 2d animation depicting polyman walking on stage from the right side of the screen,
he will do a flip, and then walk off stage. The modelview matrix is being used to perform the
actions of translation rotation and scaling*/

/*-----------------Global Variables-----------------*/
//theta = global angular value for rotation
//dx and dy = global movement values for x and y, respectively

float bodyTheta = 0, bodyDX = 7.0, bodyDY = -3.0; //global values for the body
float mouthTheta = 0, mouthDX = 7.0, mouthDY = -3.0; //global values for the mouth
float leg1Theta = 0, leg1DX = 7.0, leg1DY = -3.0; //global value for leg 1
float leg2Theta = 0, leg2DX = 7.0, leg2DY = -3.0; //global value for leg 2

int frame = 1;

void init(void);//this is a function to initialize the window in a clear color
void RenderScene(void);//this is a function to draw the scene in an opened window

//***CREATING THE LOADS AND DRAWS***
//body consists of float arrays, 2 for the top of the body and 2 for the bottom of the body
void loadBody(float[], float[], float[], float[], float[], float[]); //loads the body //adding additional
floats
void drawBody(float[], float[], float[], float[], float[], float[]); //draws the body //adding
additional floats

//mouth consists of 2 float arrays
void loadMouth(float[], float[]); //loads the mouth
void drawMouth(float[], float[]); //draws the mouth

//leg 1 and leg 2 consist of 2 float arrays
void loadLeg1(float[], float[]); //loads leg 1
```

```
void drawLeg1(float[], float[]); //draws leg 1
void loadLeg2(float[], float[]); //loads leg 2
void drawLeg2(float[], float[]); //draws leg 2

//***CREATING THE MODELVIEW MATRICIES***
void bodyModel(void); //sets the MODELVIEW MATRIX for the body (rotation/translation
matrix)
void mouthModel(void); //sets the MODELVIEW MATRIX for the mouth (rotation/translation
matrix)
void leg1Model(void); //sets the MODELVIEW MATRIX for leg 1 (rotation/translation matrix)
void leg2Model(void); //sets the MODELVIEW MATRIX for leg 2 (rotation/translation matrix)

void SetupRC(void);//sets up the clear color
void TimerFunction(int);//this call back function is call each 30 ms and changes the location,
scale and rotation of the square

//Main Program
int main(int argc, char** argv)
{
        //set up the window title
        char header[] = "Polyman by Nick Wright";

        /*glutInit() initializes GLUT.  Takes the command line arguments which are used to
        initialize the native window system.
        This function must be called before any other GLUT functions.*/
        glutInit(&argc, argv);

        //set up the display mode with a single buffer and rgb colors
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

        //initialize the window size and position
        glutInitWindowSize(560, 440);
        glutInitWindowPosition(140, 20);

        //Initialize background color in window to red
        SetupRC();

        //  Open and Label Window
        glutCreateWindow(header);
        glutDisplayFunc(RenderScene);
        glutTimerFunc(30, TimerFunction, 1);

        //now draw the scene
        glutMainLoop();
```

```
	return 0;
}

//Render Scene Function
void RenderScene(void)
{
	float xdel = 0.25;

	//pattern for body icon
	//upperX and upperY are the x and y points for the upper body
	//lowerX and lowerY are the x and y points for the lower body
	//eyeX and eyeY are for the eye
	float upperX[5], upperY[5], lowerX[6], lowerY[6], eyeX[1], eyeY[1];

	//pattern for the mouth icon
	float mouthX[4], mouthY[4];

	//pattern for leg 1 icon
	float leg1X[3], leg1Y[3];

	//pattern for leg 2 icon
	float leg2X[3], leg2Y[3];

	//clear the window with the current background color
	cout << "in renderscene" << endl;

	//set the current drawing color to white
	glColor3f(1.0, 1.0, 1.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();

	//set the viewport to the window dimensions
	glViewport(0, 0, 540, 440);

	//Establish the clipping volume in user coordinates
	glOrtho(-7.0, 7.0, -7.0, 7.0, 1.0, -1.0);

	//load the icons untransformed
	loadBody(upperX, upperY, lowerX, lowerY, eyeX, eyeY);
	loadMouth(mouthX, mouthY);
	loadLeg1(leg1X, leg1Y);
	loadLeg2(leg2X, leg2Y);

	//clear the window with the background color
```

```
	glClear(GL_COLOR_BUFFER_BIT);

	//set the current drawing color to white
	glColor3f(1.0, 1.0, 1.0);

	//glFlush being performed after each draw
	bodyModel(); //body modelview matrix
	drawBody(upperX, upperY, lowerX, lowerY, eyeX, eyeY);
	glFlush();
	mouthModel(); //mouth modelview matrix
	drawMouth(mouthX, mouthY);
	glFlush();
	leg1Model(); //leg 1 modelview matrix
	drawLeg1(leg1X, leg1Y);
	glFlush();
	leg2Model(); //leg 2 modelview matrix
	drawLeg2(leg2X, leg2Y);
	glFlush();

	glEnd();
	glutSwapBuffers();
	return;
}//end of renderscene

//loadBody function
void loadBody(float upperX[], float upperY[], float lowerX[], float lowerY[], float eyeX[], float
eyeY[])
{
	//this function will load the upper and lower body icons

	//set the coordinates of upper body
	upperX[0] = -9.0 / 8;	upperY[0] = 0;
	upperX[1] = -5.0 / 8;	upperY[1] = 3.0 / 4;
	upperX[2] = 5.0 / 8;	upperY[2] = 3.0 / 4;
	upperX[3] = 9.0 / 8;	upperY[3] = 0;
	upperX[4] = -9.0 / 8;	upperY[4] = 0;

	//set the coordinates of the lower body
	lowerX[0] = -7.0 / 8;	lowerY[0] = -1.0 / 2;
	lowerX[1] = -3.0 / 8;	lowerY[1] = 0;
	lowerX[2] = 9.0 / 8;	lowerY[2] = 0;
	lowerX[3] = 5.0 / 8;	lowerY[3] = -3.0 / 4;
	lowerX[4] = -5.0 / 8;	lowerY[4] = -3.0 / 4;
	lowerX[5] = -7.0 / 8;	lowerY[5] = -1.0 / 2;
```

```
	//set the coordinates of the eye
	eyeX[0] = -1.0 / 2;			eyeY[0] = 1.0 / 2;
}//end of loadBody

//loadMouth function
void loadMouth(float mouthX[], float mouthY[])
{
	//this function will load the mouth icons

	//set the coordinates of the mouth
	mouthX[0] = -9.0 / 8;	mouthY[0] = 0;
	mouthX[1] = -3.0 / 8;	mouthY[1] = 0;
	mouthX[2] = -7.0 / 8;	mouthY[2] = -1.0 / 2;
	mouthX[3] = -9.0 / 8;	mouthY[3] = 0;

	/*My solution to make the mouth hidden:
	if the x value of mouth is at 0, make the polygon a shape that cannot be drawn,
	this way the shape is "hidden"*/
	if (mouthDX == 0)
	{
		mouthX[0] = 0;  mouthY[0] = 0;
		mouthX[1] = 0;  mouthY[1] = 0;
		mouthX[2] = 0;  mouthY[2] = 0;
		mouthX[3] = 0;  mouthY[3] = 0;
	}
}//end of loadMouth

//loadLeg1 function
void loadLeg1(float leg1X[], float leg1Y[])
{
	//this function will load leg 1

	//setting the coordinates of leg 1
	leg1X[0] = -1.0 / 4;	leg1Y[0] = -1.0 / 2;
	leg1X[1] = -1.0 / 4;	leg1Y[1] = -1.0;
	leg1X[2] = -1.0 / 2;	leg1Y[2] = -1.0;
}//end of loadLeg1

//loadLeg2 function
void loadLeg2(float leg2X[], float leg2Y[])
{
	//this function will load leg 2
```

```
	//setting the coordinates of leg 2
	leg2X[0] = 1.0 / 4;		leg2Y[0] = -1.0 / 2;
	leg2X[1] = 1.0 / 4;		leg2Y[1] = -1.0;
	leg2X[2] = 0.0;			leg2Y[2] = -1.0;
}//end of loadLeg2

//function drawBody
void drawBody(float upperX[], float upperY[], float lowerX[], float lowerY[], float eyeX[], float
eyeY[])
{
	//this function will draw the icon at the transformed position
	int i;
	cout << "in drawBody" << endl;

	//drawing upper
	glBegin(GL_LINE_STRIP);
	//move to first point in upper icon
	glVertex2f(upperX[0], upperY[0]);
	//now draw the rest of upper
	for (i = 1; i <= 4; i++)
	{
		glVertex2f(upperX[i], upperY[i]);
	}
	glEnd();
	//filling upper
	//set the shading color to yellow
	glColor3f(1.0, 1.0, 0.0);
	glShadeModel(GL_FLAT);
	//redraw the polygon
	glBegin(GL_POLYGON);
	//the colored polygon must be redrawn to render it
	for (i = 0; i <= 3; i++)
	{
		glVertex2f(upperX[i], upperY[i]);
	}
	glEnd();

	//drawing lower
	glBegin(GL_LINE_STRIP);
	//move to first point in lower icon
	glVertex2f(lowerX[0], lowerY[0]);
	//now draw the rest of upper
	for (i = 1; i <= 5; i++)
	{
```

```
                glVertex2f(lowerX[i], lowerY[i]);
        }
        glEnd();
        //filling lower
        //set the shading color to yellow
        glColor3f(1.0, 1.0, 0.0);
        glShadeModel(GL_FLAT);
        //redraw the polygon
        glBegin(GL_POLYGON);
        //the colored polygon must be redrawn to render it
        for (i = 0; i <= 4; i++)
        {
                glVertex2f(lowerX[i], lowerY[i]);
        }
        glEnd();

        //drawing the eye
        //set the color to black
        glColor3f(0.0, 0.0, 0.0);
        //setting the point size to 3
        glPointSize(3);
        glBegin(GL_POINTS);
        glVertex2f(eyeX[0], eyeY[0]);
        glEnd();

        return;
}//end of drawBody

//drawMouth
void drawMouth(float mouthX[], float mouthY[])
{
        //this function will draw the icon at the transformed position
        int i;
        cout << "in drawMouth" << endl;

        //drawing mouth
        glColor3f(1.0, 1.0, 0.0);
        glBegin(GL_LINE_STRIP);
        //move to first point in upper icon
        glVertex2f(mouthX[0], mouthY[0]);
        //now draw the rest of upper
        for (i = 1; i <= 3; i++)
        {
                glVertex2f(mouthX[i], mouthY[i]);
```

```
	}
	glEnd();
	//filling mouth
	//set the shading color to yellow
	glColor3f(1.0, 1.0, 0.0);
	glShadeModel(GL_FLAT);
	//redraw the polygon
	glBegin(GL_POLYGON);
	//the colored polygon must be redrawn to render it
	for (i = 0; i <= 2; i++)
	{
		glVertex2f(mouthX[i], mouthY[i]);
	}
	glEnd();

	return;
}

//drawLeg1
void drawLeg1(float leg1X[], float leg1Y[])
{
	//this function will draw the icon at the transformed position
	int i;
	cout << "in drawLeg1" << endl;

	//drawing leg 1
	glBegin(GL_LINE_STRIP);
	//move to first point in upper icon
	glVertex2f(leg1X[0], leg1Y[0]);
	//now draw the rest of leg 1
	for (i = 1; i <= 2; i++)
	{
		glVertex2f(leg1X[i], leg1Y[i]);
	}
	glEnd();
}//end of draw leg 1

//drawLeg2
void drawLeg2(float leg2X[], float leg2Y[])
{
	//this function will draw the icon at the transformed position
	int i;
	cout << "in drawLeg2" << endl;
```

```
	//drawing leg 2
	glBegin(GL_LINE_STRIP); //try making this GL_LINES (WAS GL_LINE_STRIP)
	//move to first point in upper icon
	glVertex2f(leg2X[0], leg2Y[0]);
	//glVertex2f(leg2X[1], leg2Y[0]);
	//glVertex2f(leg2X[2], leg2Y[2]);
	//now draw the rest of leg 2
	for (i = 1; i <= 2; i++)
	{
		glVertex2f(leg2X[i], leg2Y[i]);
	}
	glEnd();
}//end of drawleg2

//function bodyModel
void bodyModel()
{
	//float bodyTheta = 0, bodyDX = -6.0, bodyDY = -3.0;
	//sets the modelviel matrix for the body
	cout << "in bodyModel" << endl;
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	glTranslatef(bodyDX, bodyDY, 0.0);
	glRotatef(bodyTheta, 0.0, 0.0, 1.0);// note that the angle theta is in degrees, not radians

	return;
}//end of bodyModel

//function mouthModel
void mouthModel()
{
	//float mouthTheta = 0, mouthDX = -6.0, mouthDY = -3.0;
	//sets the modelviel matrix for the mouth
	cout << "in mouthModel" << endl;
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	glTranslatef(mouthDX, mouthDY, 0.0);
	glRotatef(mouthTheta, 0.0, 0.0, 1.0);// note that the angle theta is in degrees, not
radians

	return;
}//end of mouthModel

//function leg1Model
```

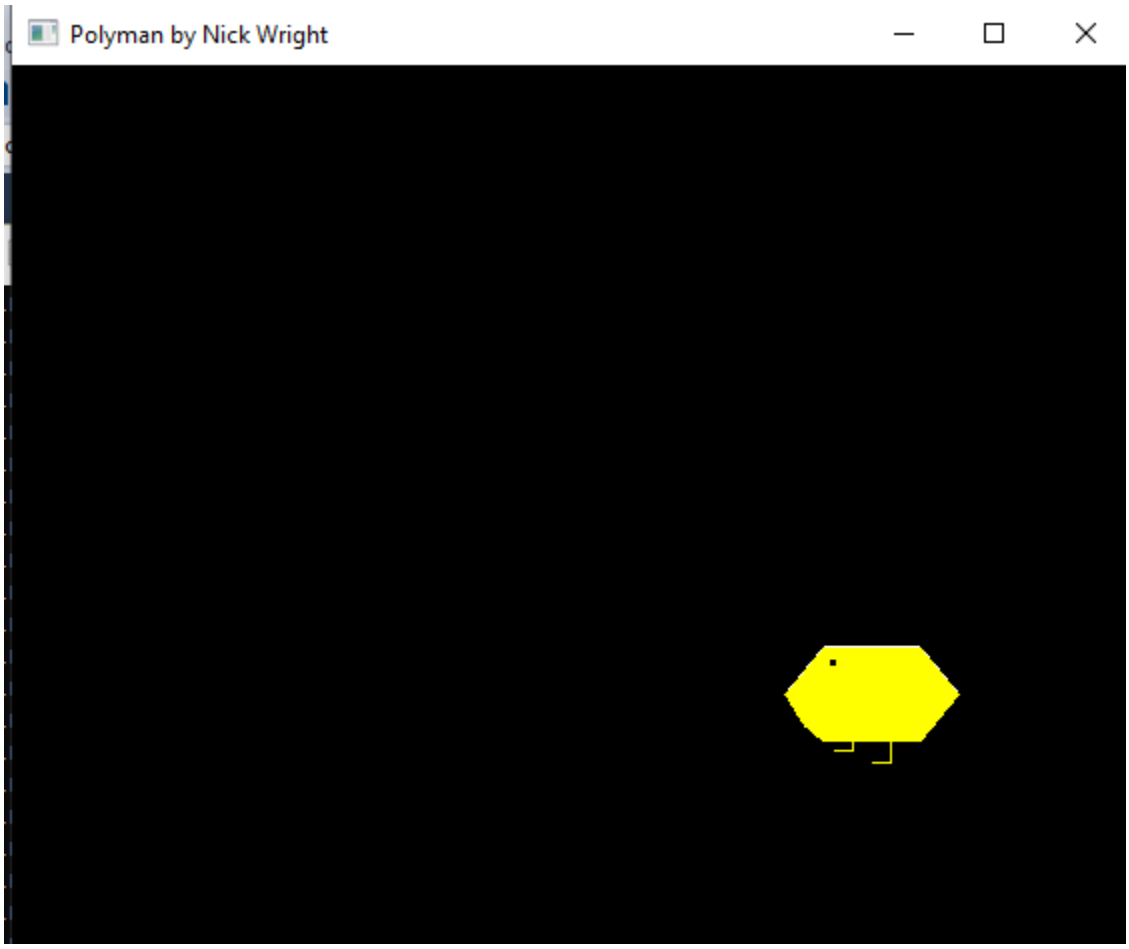
```
void leg1Model()
{
	//float leg1Theta = 0, leg1DX = -6.0, leg1DY = -3.0;
	//sets the modelviel matrix for leg1
	cout << "in leg1Model" << endl;
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	glTranslatef(leg1DX, leg1DY, 0.0);
	glRotatef(leg1Theta, 0.0, 0.0, 1.0);// note that the angle theta is in degrees, not radians

	return;
}//end of leg1Model

//function leg2Model
void leg2Model()
{
	//float leg2Theta = 0, leg2DX = -6.0, leg2DY = -3.0;
	//sets the modelviel matrix for leg2
	cout << "in leg2Model" << endl;
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	glTranslatef(leg2DX, leg2DY, 0.0);
	glRotatef(leg2Theta, 0.0, 0.0, 1.0);// note that the angle theta is in degrees, not radians

	return;
}//end of leg2Model

//function SetupRC
// Setup the rendering state
void SetupRC(void)
{
	// this function sets the clear color of an open window and clears the open window
	// Set clear color to blue
	glClearColor(0.0, 0.0, 1.0, 1.0);
	return;
}//end of setuprc

//function timer
void TimerFunction(int value)
{
	//this call back function is called each 30 ms and changes the location, scale and rotation
of the polygons
	switch (frame)
	{
```

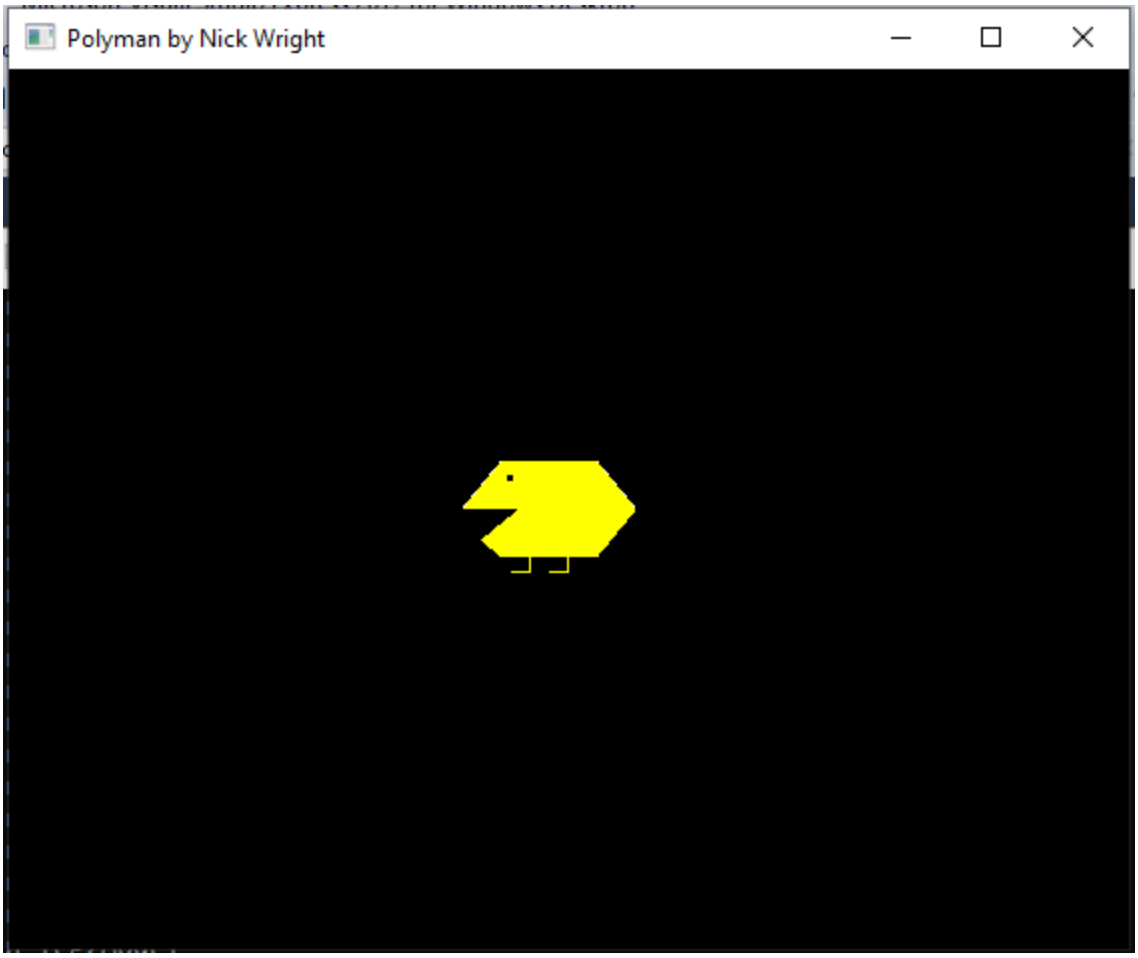
```
case 1:
        //frame 1 polyman starts at the right (7, -3) and walks to the middle (0,-3)

        //body parameters
        bodyDX -= 0.15;
        //mouth parameters
        mouthDX -= 0.15;
        //leg1 parameters
        leg1DX -= 0.15;
        //leg2 parameters
        leg2DX -= 0.15;

        //if else statement to make the legs move up and down
        if (leg1DY > -3) {
                leg1DY -= 0.1;
                leg2DY += 0.1;
        }
        else {
                leg1DY += 0.1;
                leg2DY -= 0.1;
        }

        //use body position to change frame
        if (bodyDX <= 0)
        {
                bodyDX = 0;
                mouthDX = 0;
                leg1DX = 0;
                leg2DX = 0;
                leg1DY = -3.0;
                leg2DY = -3.0;
                frame = 2;
        }
        break;

case 2:
        //frame 2 polyman opens his mouth and jumps into the air (y = 5)

        //body parameters
        bodyDY += 0.2;
        //leg1 parameters
        leg1DY += 0.2;
        //leg2 parameters
        leg2DY += 0.2;
```

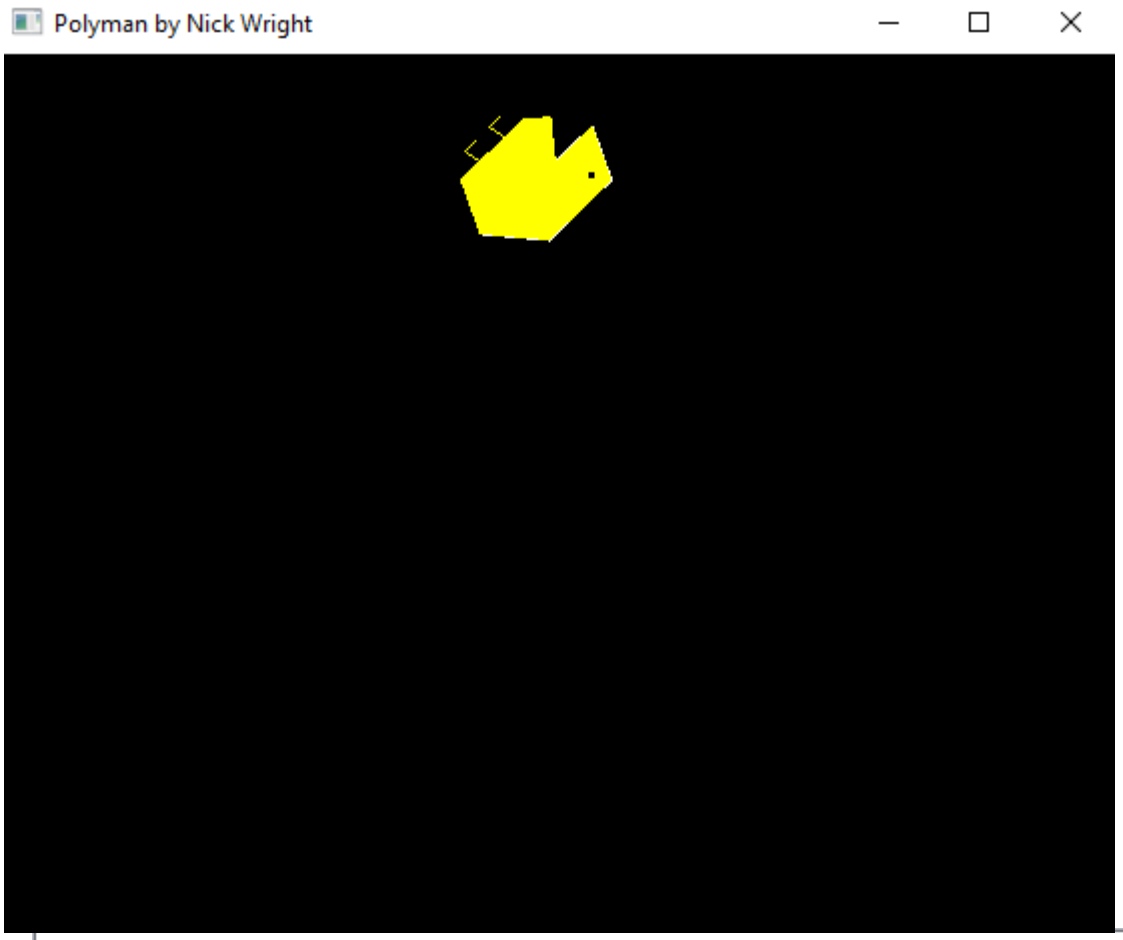
```
		if (bodyDY > 5.0)
		{
			bodyDY = 5.0;
			leg1DY = 5.0;
			leg2DY = 5.0;
			frame = 3;
		}
		break;

	case 3:
		//frame 3 polyman rotates 360 degrees
		//using negtaive theta value to make him do a backflip

		//body parameters
		bodyTheta -= 5.0;
		//leg1 parameters
		leg1Theta -= 5.0;
		//leg2 parameters
		leg2Theta -= 5.0;

		if (bodyTheta <= -360.0)
		{
			frame = 4;
			bodyTheta = 0.0;
			leg1Theta = 0.0;
			leg2Theta = 0.0;
		}
		break;

	case 4:
		//frame 4 polyman lands back down on the ground (y = -3.0) polyman also
closes his mouth

		//body parameters
		bodyDY -= 0.2;
		//leg1 parameters
		leg1DY -= 0.2;
		//leg2 parameters
		leg2DY -= 0.2;

		if (bodyDY <= -3.0)
		{
			bodyDY = -3.0;
```

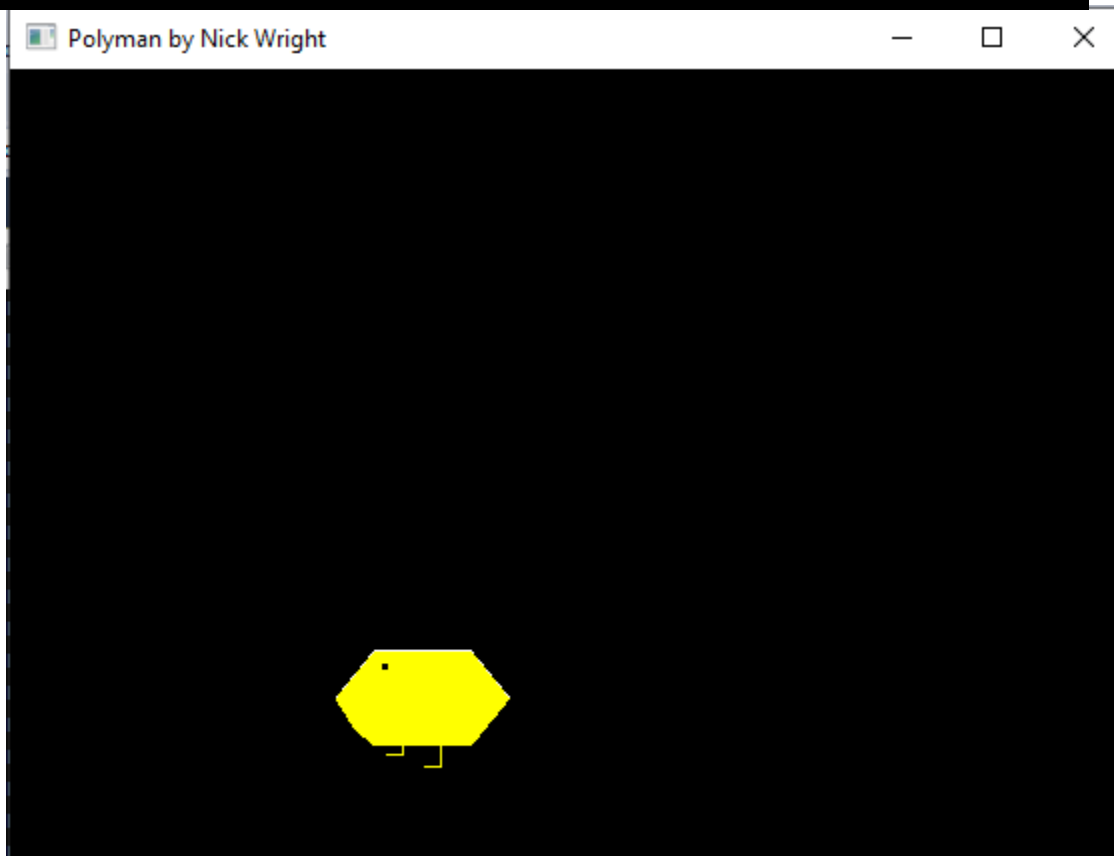
```
                leg1DY = -3.0;
                leg2DY = -3.0;
                frame = 5;
            }
            break;

        case 5:
            //frame 5 polyman walks off of the stage to the left

            //body parameters
            bodyDX -= 0.15;
            //mouth parameters
            mouthDX -= 0.15;
            //leg1 parameters
            leg1DX -= 0.15;
            //leg2 parameters
            leg2DX -= 0.15;

            //if else statement to make the legs move up and down
            if (leg1DY > -3) {
                leg1DY -= 0.1;
                leg2DY += 0.1;
            }
            else {
                leg1DY += 0.1;
                leg2DY -= 0.1;
            }

            if (bodyDX <= -6.0)
            {
                bodyDX = -6.5;
                mouthDX = -6.5;
                leg1DX = -6.5;
                leg2DX = -6.5;
                leg1DY = -3.0;
                leg2DY = -3.0;
                break;
            }
        }

        // Redraw the scene with new coordinates
        glutPostRedisplay();
        glutTimerFunc(30, TimerFunction, 1);
}
```

3. Output




Polyman by Nick Wright


Polyman by Nick Wright


Polyman by Nick Wright