2014

Computer Graphics OpenGL Semester 1 Project

Willem Mouton – H00180920

Charlie van Zyl – H00180839

CTI Education Group,
Durbanville,
South Africa

Table of Contents

Executive Summary

The aim of this project was to use all the fore letters of our names, Charlie William van Zyl and Willem Georg Mouton, and construct a 3 dimensional animal with them. Unfortunately we had two W's in our list of letters, so we had to switch one of them with the next letter of the alphabet, which was X. We then chose to construct a dragon with the letters C X Z M G W, using OpenGL and the Glut library.

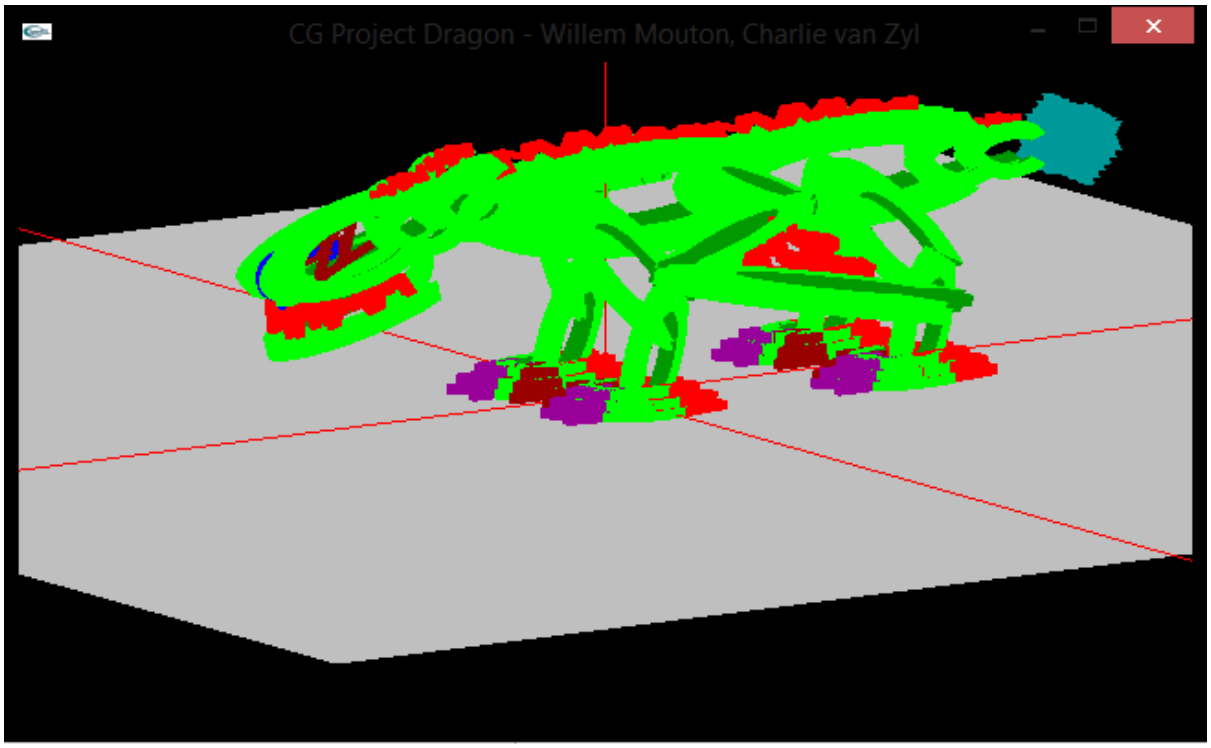


Figure 1: Our Dragon Model

After it was all done, we ended up with the dragon (as seen in the picture above), which had 24 different ways in which it can move parts of its body, and 12 different joints, one of which is a double articulated joint. The double articulated joint can be found at the jaw and head.

Project Diary

MEETING:
Date: 22 January – 24 January 2014
Place: Willem's house
Rough planning of the structure (skeleton) of the 3D model. This includes all the parts, like the head, body, neck, legs, arms, wings, and tail.

MEETING:
Date: 14 February – 16 February 2014
Place: Willem's house
Completion of the separate letters from each team member.

MEETING:
Date: 13 March 2014
Time: 12:00 – 17:00
Place: Charlie's flat
Finalize the planning for the skeletal layout of the 3D model and the design of the separate body parts.

MEETING:
Date: 19 March – 23 March 2014
Place: Willem's house
Construction of the separate body parts.

MEETING:
Date: 11 April – 13 April
Place: Willem's house
Connecting of the body parts and completion of the full body of the animal. Planning of the report.

MEETING:
Date: 17 April 2014
Place: Willem's house
Planning and implementation of the different animations for all the joints.

MEETING:
Date: 5 May 2014
Time: 11:00 – 20:00
Place: Charlie's flat
Completion of the report and the entire project.

MEETING:
Date: 9 May
Time: 10:00 – 17:00
Place: Willem's house
Last minute changes and rounding off of the entire project.

Top-level Design of Program

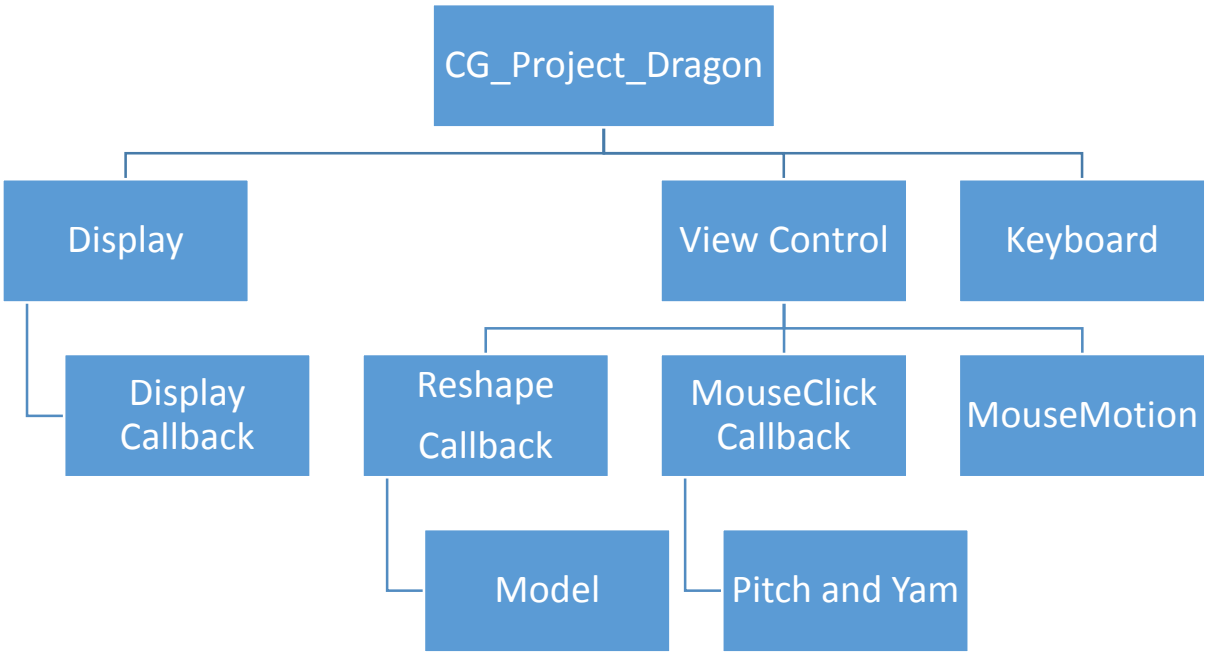


Figure 2: Top-level Design of Program

Graphical User Manual

Installation

The source code for the project is attached to this document so you can compile it buy using an IDE like Eclipse or Microsoft Visual Studio.

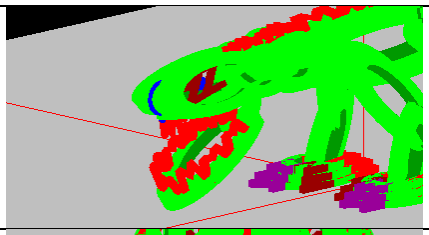
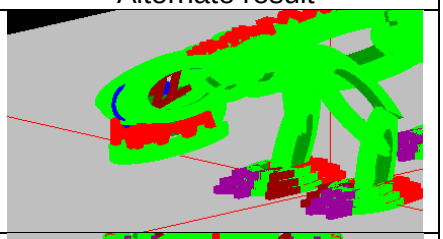
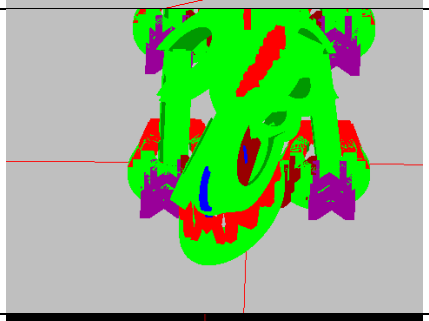
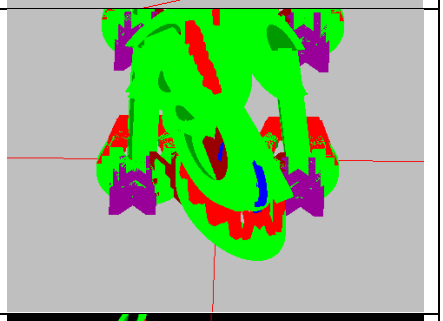
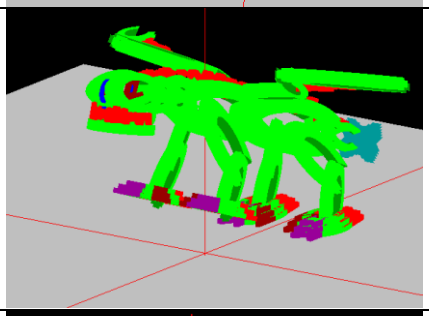
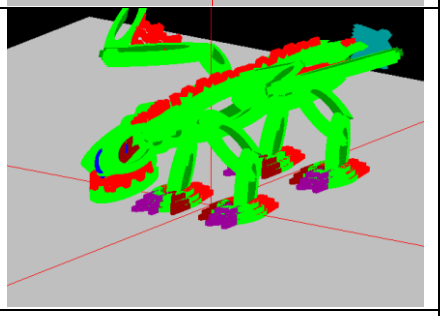
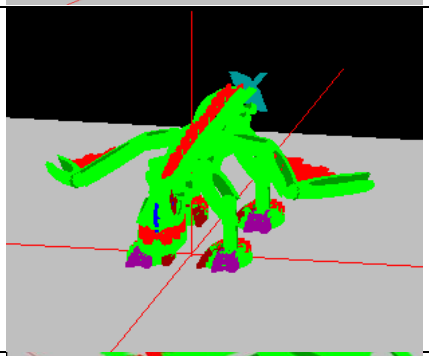
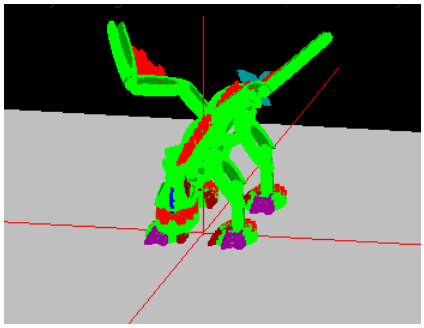
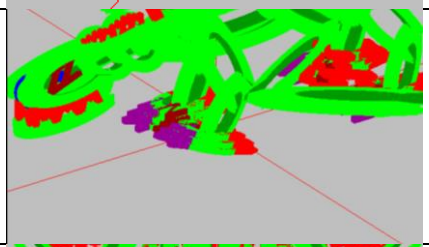
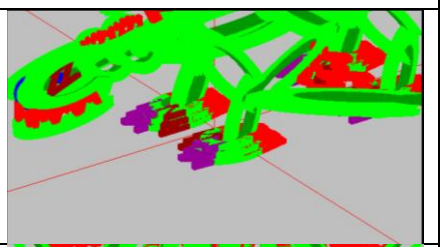
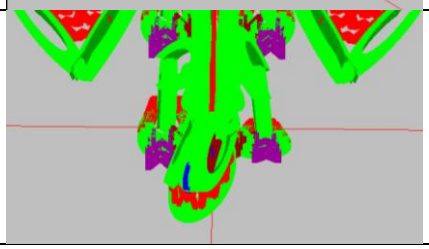
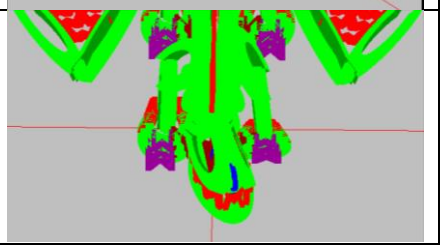
If you have access to the compiled executable file then you can just go ahead and run that.

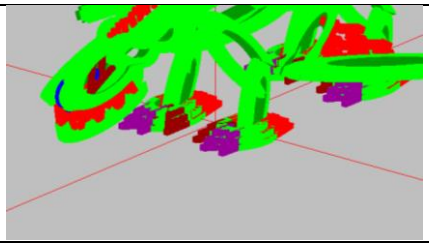
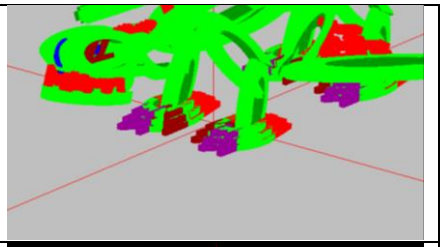
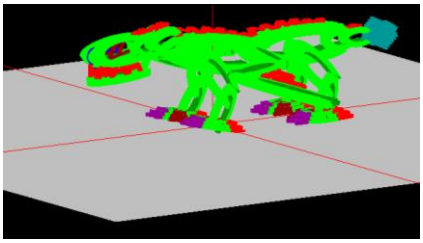
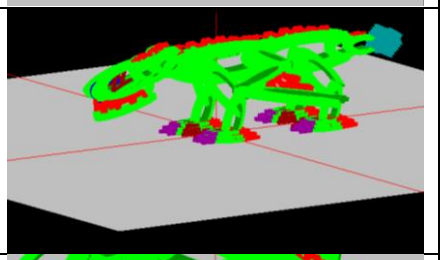
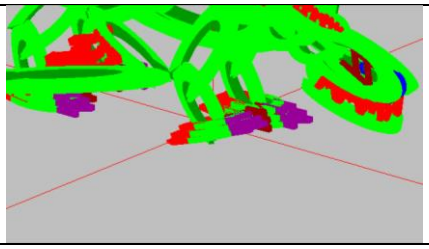
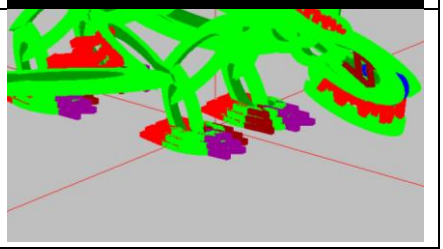
General Description

The program is fairly simple to use and the commands consists mainly out of actions that is triggered when a key on the keyboard is pressed. The model can also be rotate but clicking inside the window and dragging in the direction you wish to rotate. You can also zoom in and out by pressing the “+” to zoom in and the “-“ to zoom out.

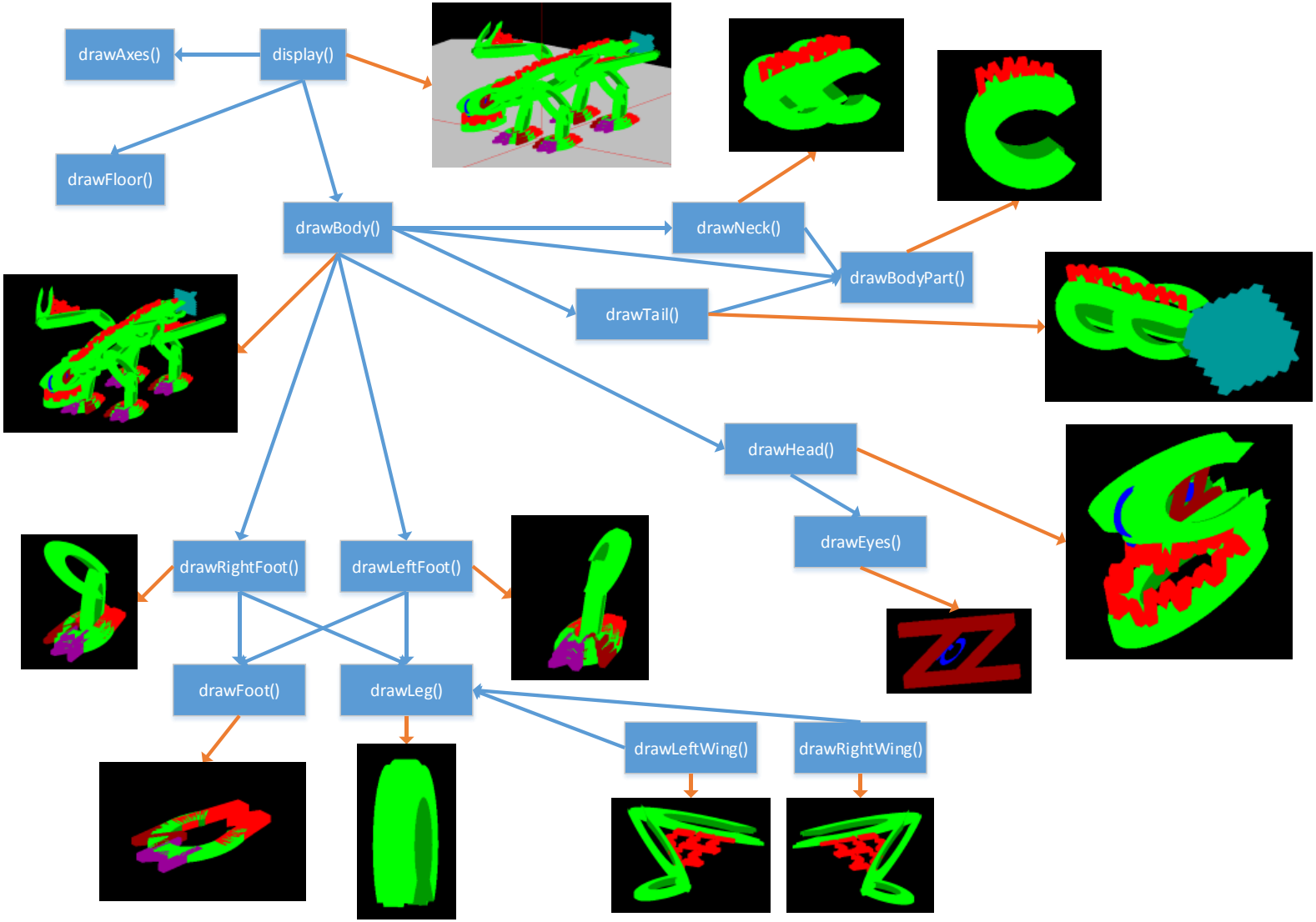
Also take note that many of the commands require the use of the “SHIFT” key to achieve the alternate action for a key.

Detailed Guide to Interact with the Model.

Action	Result	Alternate result
q = Opens the mouth Q = Closes the mouth		
w = Turns the neck left W = Turns the neck right		
E = Drops the body e = Lifts the body		
R = Turns the wings vertically up r = Turns the wings vertically down		
t = Moves the left leg forward T = Moves the left leg backwards		
a = Turns the head left A = Turns the head right		

z = Rotates head vertically down Z = Rotates head vertically up		
d = Turns the body vertically up in the middle D = Turns the body vertically down in the middle		
g = Lifts the right leg up G = Drops the right leg down		

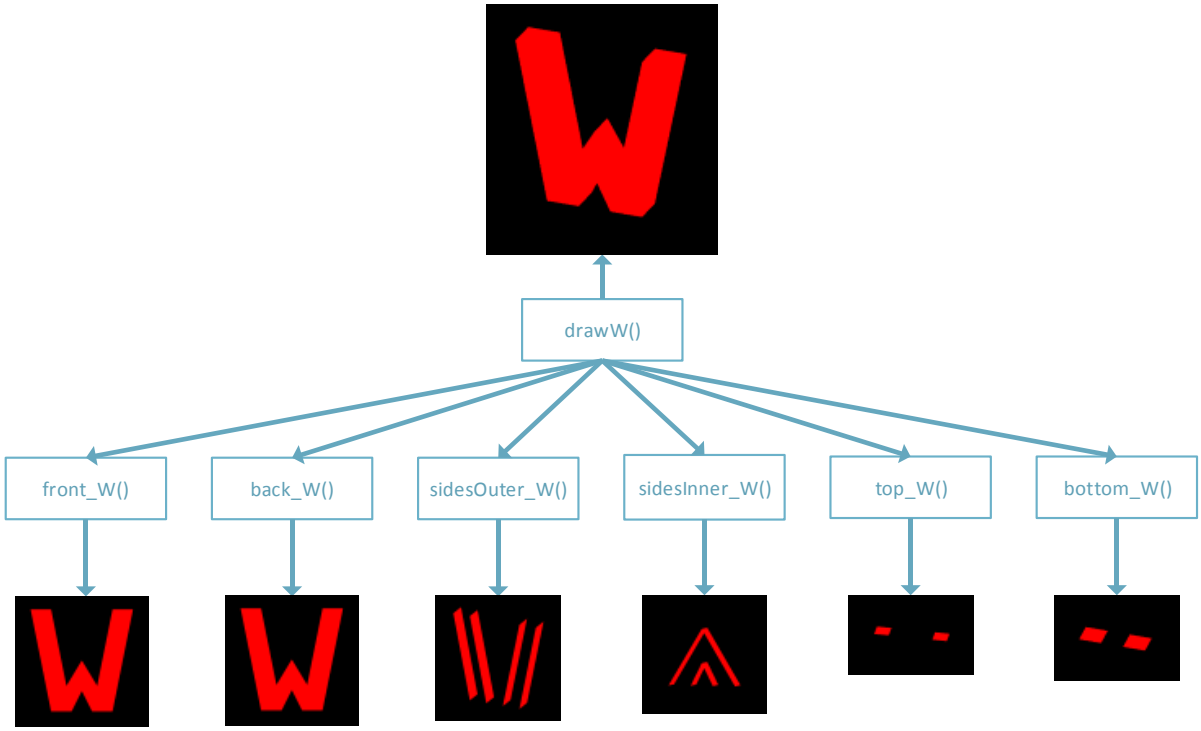
Top-Level Scene Graph



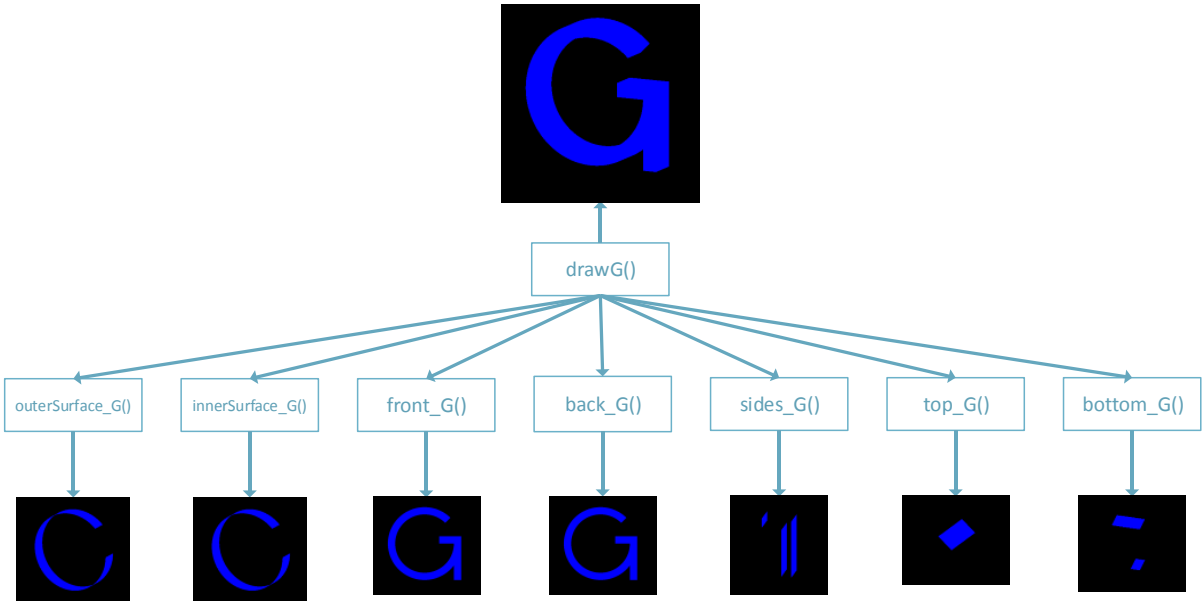
Scene Graphs of the Initials

Willem Mouton

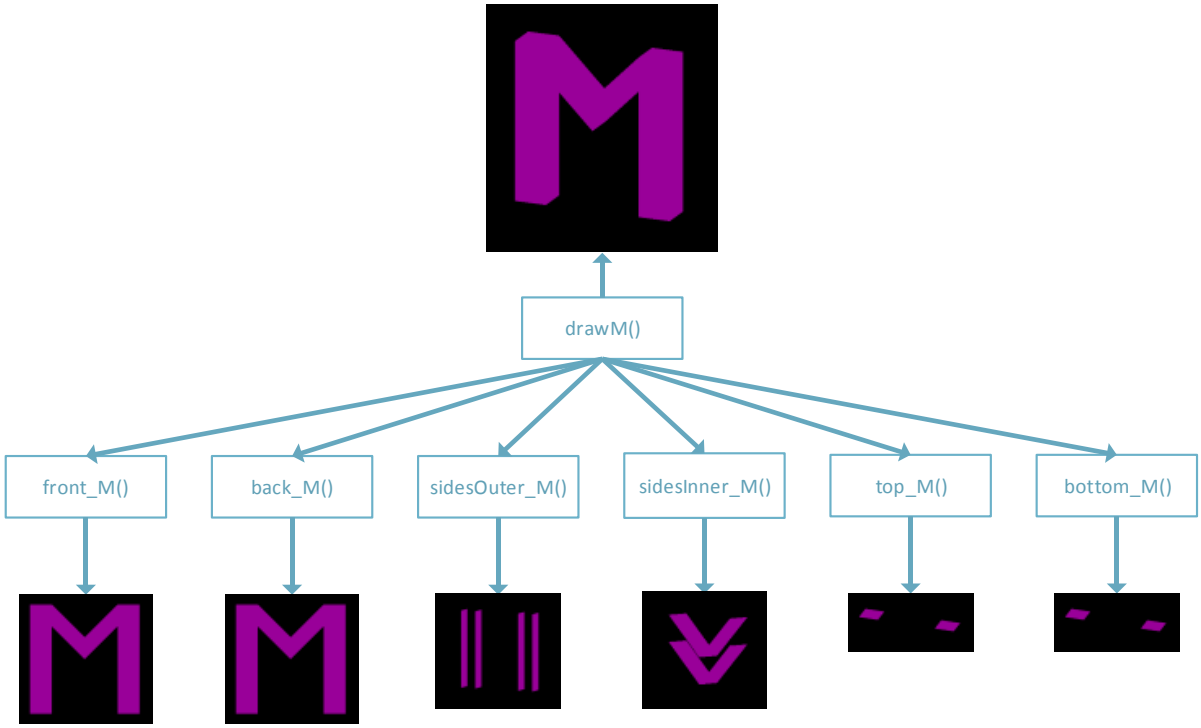
Letter W



Letter G

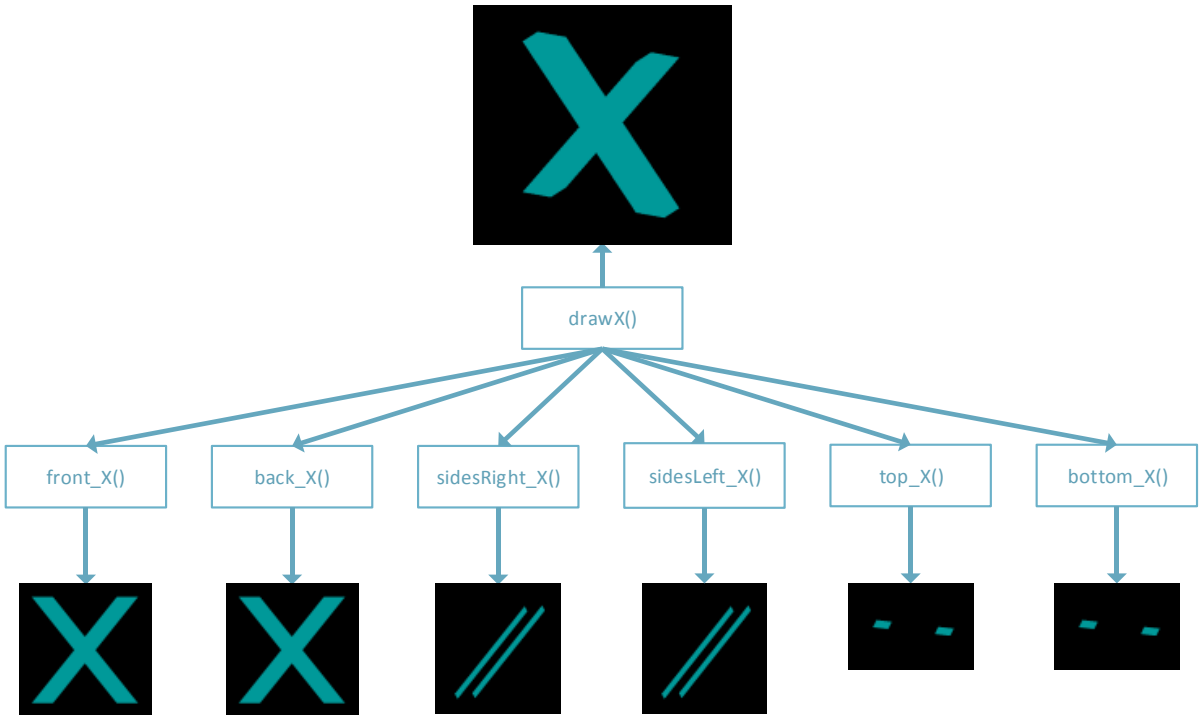


Letter M

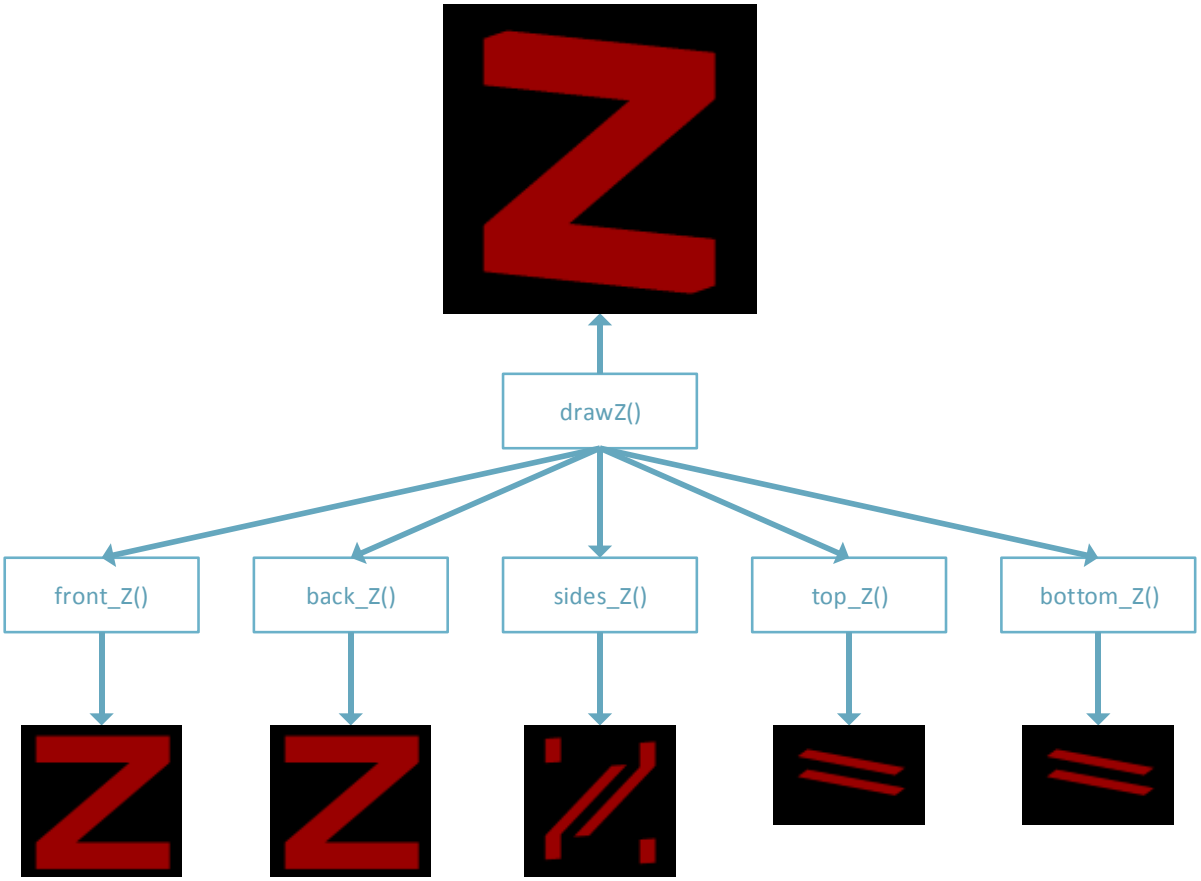


Charlie van Zyl

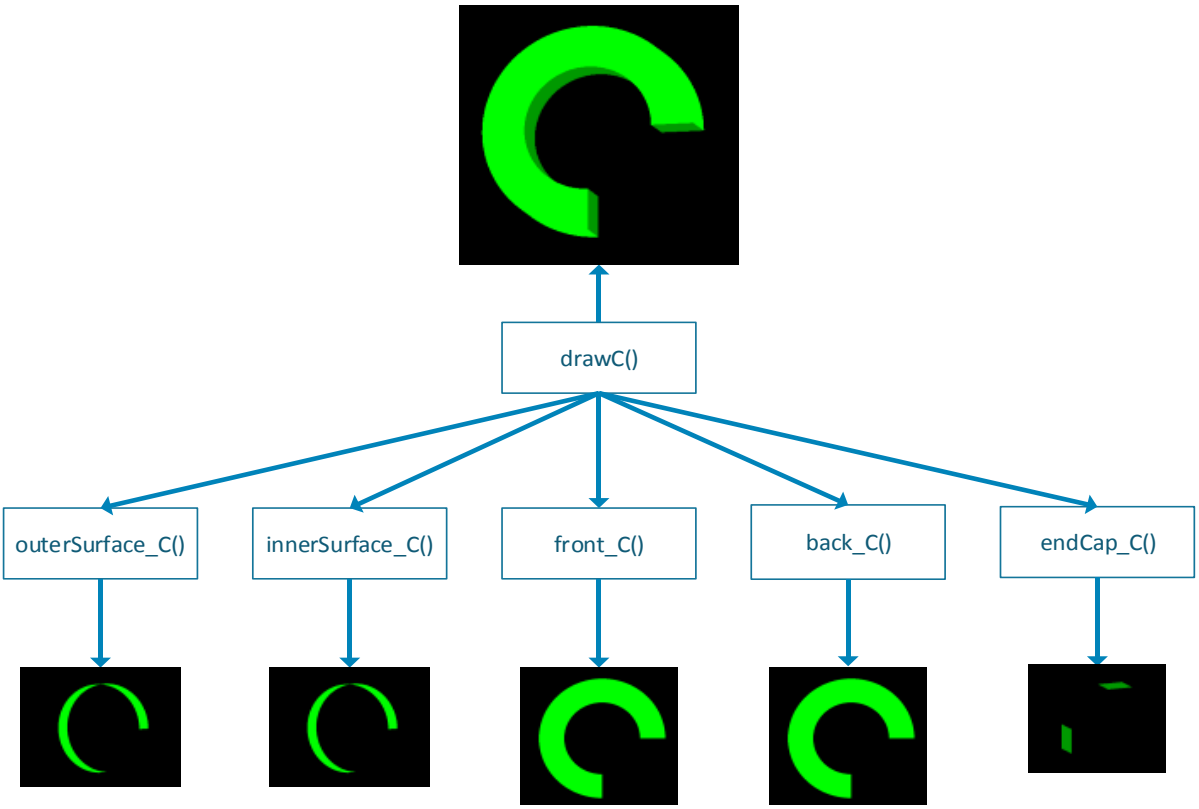
Letter X



Letter Z



Letter C



Conclusions

Willem Mouton's conclusions

If I were to point out the biggest challenge that I faced in this project, I would have to say it the sheer number of information that you need to keep track of. I must admit that this project helped me allot to better understand how to create graphics and control it, whilst also improving way I program in objective C.

If I were to do something differently next time than I would pay more attention to how objects move in the 3D space, as well as how to improve efficiency in my coding structure.

I would have liked to add a Graphical User Interface to help the user better understand the program. This will help increate the user experience and allow for deeper customization when it comes to the transformations and vertices. One last thing that I'd wish to add would have been more automation in the model for example simulating a fluent walking or flying motion.

As stated above I would defiantly add a GUI, but I would like to improve the overall code structure and make it into a more robust program, as well as increase the number of joints that can be animated, and lastly insert some automated animations.

Charlie van Zyl's conclusions

I haven't learned much from the project, except maybe that to be a 3D modelling programmer, you have to be creative in your work, and also expand your horizons, be open to new ideas, and never give up if something doesn't work the first time.

Next time I would only change the amount of times we came together, if it is at all possible, to work together more.

I would only improve upon the amount of detail in the design of the animal/monster that we have created, to ensure quality work that I can be proud of.

There is no next year, but if there were going to be a next year, we should definitely construct a work plan, to make sure we finish all goals on the time set by the work plan. This will improve efficiency and leave time for other things, like socialization.

References

Study Material

All Code examples that was made available to us.
All report examples and educational slideshow that was made available to us.

Internet

Stack Overflow
http://stackoverflow.com

Books

Edward, A. Dave, S. *Interactive Computer Graphics a Top-Down Approach With Shader-Based openGL*. 6th ed.

Source Code

CG_Project_Dragon.c

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : CG_Project_Dragon.c
 */

#include <stdio.h>
#include <stdlib.h>
#include <gl\glut.h>
#include <math.h>
#include <stdbool.h>

//Header-------------
#include "vertices.h"
#include "LetterC.h"
#include "LetterG.h"
#include "LetterW.h"
#include "LetterM.h"
#include "LetterX.h"
#include "LetterZ.h"

#define WIDTH 800
#define HEIGHT 800

static float win_theta[3] = {0.0, 0.0, 0.0};
static float win_zome = 0.6;
float pitch0, yaw0;
int mouseX0, mouseY0;

//Rotation Commands for parts
//kbtX, kbtY, kbtZ, kbrX, kbrY, kbrZ, kbsX, kbsY, kbsZ
float kbtX;
float kbtY;
float kbtZ;

float kbrX;
float kbrY;
float kbrZ;

float kbsX = 1;
float kbsY = 1;
float kbsZ = 1;

float increment = 0.1;

bool MousePressed;
bool flymode = false;

float spawnhight = 2.0;

//Pivot Points
float joint_jaw = 0;
float joint_head[2] = { 0.0, 0.0 };
float joint_neck[2] = { 0.0, 0.0 };
float joint_leftLeg = 0;
float joint_rightLeg = 0;
```

```
float joint_wings = 10;
float joint_tail[2] = {90.0, 0.0};
float joint_frontBody = 0;
float joint_backBody = 0;

//Other Functions--------------------------------------------------
void printPoints (){
	system("CLS");// Clear the console screen

	printf("Current Increment: ");
	printf("%3.2f\n", increment);

	printf("\nTranslate X: ");
	printf("%3.2f\n", kbtX);
	printf("Translate Y: ");
	printf("%3.2f\n", kbtY);
	printf("Translate Z: ");
	printf("%3.2f\n", kbtZ);

	printf("\nRotate X: ");
	printf("%3.2f\n", kbrX);
	printf("Rotate Y: ");
	printf("%3.2f\n", kbrY);
	printf("Rotate Z: ");
	printf("%3.2f\n", kbrZ);

	printf("\nScale X: ");
	printf("%3.2f\n", kbsX);
	printf("Scale Y: ");
	printf("%3.2f\n", kbsY);
	printf("Scale Z: ");
	printf("%3.2f\n", kbsZ);

	printf("\n\n (");
	printf("%3.1f, ", kbtX);
	printf("%3.1f, ", kbtY);
	printf("%3.1f, ", kbtZ);
	printf("%3.1f, ", kbrX);
	printf("%3.1f, ", kbrY);
	printf("%3.1f, ", kbrZ);
	printf("%3.1f, ", kbsX);
	printf("%3.1f, ", kbsY);
	printf("%3.1f", kbsZ);
	printf(")");
}
void printMenu (){
	printf("List of keys:");
//	printf("\n 1  - Make the dragon fly.");
	printf("\nQ/q - Open/Close mouth.");
	printf("\nA/a - Rotate head horizontally.");
	printf("\nZ/z - Rotate head vertically.");
	printf("\nW/w - Turns the neck horizontally.");
	printf("\nS/s - Rotates the neck vertically.");
	printf("\nE/e - Drops and lowers the back body part.");
	printf("\nD/d - Drops and lowers the front body part.");
	printf("\nR/r - Rotates wings vertically.");
	printf("\nT/t - Rotates the left leg, forwards and backwards.");
	printf("\nG/g - Rotates the left right, forwards and backwards.");
	printf("\nF/f - Spins the tail.");
	printf("\nV/v - Rotates the tail clockwise or anti-clockwise");
	printf("\n+/- - Zoom in and out.");
```

```
	printf("\nUse mouse to rotate the dragon.");

}
void transform(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz){
	glTranslatef(tx, ty, tz);
	glRotatef(rx, 1, 0, 0);
	glRotatef(ry, 0, 1, 0);
	glRotatef(rz, 0, 0, 1);
	glScalef(sx, sy, sz);

}

//Smaller Parts--------------------------------------------------
void drawBodyPart(){
	glPushMatrix();
		drawW(0.1, 0.4, 0.0, 0.0, 0.0, 170.0, 0.3, 0.3, 0.3);
		drawW(-0.1, 0.4, 0.0, 0.0, 0.0, -180.0, 0.3, 0.4, 0.3);
		drawW(-0.3, 0.4, 0.0, 0.0, 0.0, -160.0, 0.3, 0.3, 0.3);
		drawC(0.0, 0.0, 0.0, 30.0, 0.0, 45.0, 1.0, 1.0, 1.0);
		drawC(0.0, 0.0, 0.0, -30, 0.0, 45.0, 1.0, 1.0, 1.0);
	glPopMatrix();
}
void drawEyes () {
	glPushMatrix(); //Eye
		transform(-0.2, 0.8, 0.2, -21.0, 0.0, 9.0, 0.4, 0.4, 0.5);
		drawG(-0.1, 0, 0, 0, 180, 49, 0.2, 0.5, 1);
		drawZ(-0.4, 0, 0, 0, 0, 0, 1, 1, 1);
		drawZ(0.4, 0, 0, 0, 0, 0, 1, 1, 1);
	glPopMatrix();
}

//Body Parts-----------------------------------------------------
void drawFoot () {
	drawZ(-0.4, 0.0, -0.3, 90.0, 0.0, 0.0, 0.6, 0.4, 0.5);
	drawM(-0.6, 0.0, 0.1, 90.0, 0.0, 90.0, 0.5, 0.7, 0.5);
	drawW(0.5, 0.0, 0.0, 90.0, 0.0, 90.0, 1.0, 1.0, 0.5);
	drawC(0.0, 0.0, 0.0, 90.0, 0.0, 45.0, 1.5, 0.5, 1.0);
}
void drawLeg(){
	drawC(0.0, 0.0, 0.0, 0.0, 0.0, -45.0, 0.7, 1.5, 1.8);
	drawC(0.0, 0.0, 0.0, 0.0, 0.0, -45.0, 0.8, 1.6, 1.0);
}
void drawRightFoot () {
	drawFoot();

	glPushMatrix();
		transform(0.0, 0.1, 0.0, 0.0, 0.0, 0.0, 0.9, 1.0, 0.9);
		drawFoot();

		glPushMatrix();
			transform(0.0, 0.1, 0.0, 0.0, 0.0, 0.0, 0.8, 1.0, 0.8);
			drawFoot();

			glPushMatrix();
				transform(0.4, 0.3, 0.0, -7.0, 7.0, -30.0, 1.0,
1.1, 1.0);
				drawLeg();

				glPushMatrix();
```

```
                                transform(-0.6, 0.6, -0.2, 2.0, -23.0, 80.0,
1.0, 1.6, 1.0);
                                drawLeg();
                            glPopMatrix();
                        glPopMatrix();
                glPopMatrix();
        glPopMatrix();
}
void drawLeftFoot () {
        transform(0.0, 0.0, 0.0, 180.0, 0.0, 0.0, 1.0, 1.0, 1.0);
        drawFoot();

        glPushMatrix();
                transform(0.0, -0.1, 0.0, 0.0, 0.0, 0.0, 0.9, 1.0, 0.9);
                drawFoot();

                glPushMatrix();
                        transform(0.0, -0.1, 0.0, 0.0, 0.0, 0.0, 0.8, 1.0, 0.8);
                        drawFoot();

                        glPushMatrix();
                            transform(0.4, -0.3, 0.0, 7.0, -7.0, 210.0, 1.0,
1.1, 1.0);
                            drawLeg();

                            glPushMatrix();
                                transform(0.6, 0.6, -0.2, -2.0, 23.0, -80.0,
1.0, 1.6, 1.0);
//                          transform(kbtX, kbtY, kbtZ, kbrX, kbrY, kbrZ, kbsX,
kbsY, kbsZ);
//                          transform(-0.6, 0.6, -0.2, 2.0, -23.0, 80.0, 1.0,
1.6, 1.0);
                                drawLeg();
                            glPopMatrix();
                        glPopMatrix();
                glPopMatrix();
        glPopMatrix();
}
void drawRightWing (){
        glPushMatrix();
                transform(0.0, 0.0, 0.0, -90.0, 0.0, 0.0, 0.6, 1.0, 0.8);
                drawLeg();

                glPushMatrix();
                        glPushMatrix();
                        transform(-2.2, -0.2, 0.0, 0.0, 0.0, -96.0, 0.6, 3.3,
1.0);
                        drawLeg();

                            glPushMatrix();
                                transform(2.1, -0.0, 0.0, 0.0, 0.0, 103.0,
0.4, 3.0, 0.9);
                                drawLeg();

                            glPopMatrix();
                        glPopMatrix();

                        drawW(-2.2, -0.7, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                        drawW(-1.8, -1.2, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                        drawW(-1.5, -0.7, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                        drawW(-1.4, -1.6, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
```

```
                drawW(-1.1, -1.1, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-0.7, -0.6, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
            glPopMatrix();
        glPopMatrix();

}
void drawLeftWing (){
        glPushMatrix();
        transform(0.0, 0.0, 0.0, 90.0, 0.0, 0.0, 0.6, 1.0, 0.8);
            drawLeg();

            glPushMatrix();

                glPushMatrix();
                transform(-2.2, -0.2, 0.0, 0.0, 0.0, -96.0, 0.6, 3.3,
1.0);
                drawLeg();

                    glPushMatrix();
                        transform(2.1, -0.0, 0.0, 0.0, 0.0, 103.0,
0.4, 3.0, 0.9);
                        drawLeg();

                    glPopMatrix();
                glPopMatrix();

                drawW(-2.2, -0.7, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-1.8, -1.2, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-1.5, -0.7, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-1.4, -1.6, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-1.1, -1.1, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
                drawW(-0.7, -0.6, 0.0, 0.0, 0.0, 240.0, 0.8, 0.5, 0.9);
            glPopMatrix();
        glPopMatrix();

}
void drawTail(){
        glPushMatrix();
            transform(0.0, 0.0, -0.9, 0.0, -90.0, 0.0, 1.0, 0.7, 1.0);
            drawBodyPart();
        glPopMatrix();

        glPushMatrix();
            glPushMatrix();
                glPushMatrix();
                    transform(0.0, 0.0, -0.3, 0.0, -90.0, 0.0, 1.0,
0.6, 1.0);
                    drawBodyPart();
                glPopMatrix();

                glPushMatrix();
                    drawX(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.3, 0.3, 0.5);
                    drawX(0.0, 0.0, 0.1, 0.0, 0.0, 0.0, 0.4, 0.4, 0.5);
                    drawX(0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.5, 0.5, 0.5);
                    drawX(0.0, 0.0, 0.3, 0.0, 0.0, 0.0, 0.6, 0.6, 0.5);
                    drawX(0.0, 0.0, 0.4, 0.0, 0.0, 0.0, 0.7, 0.7, 0.5);
                    drawX(0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.8, 0.8, 0.5);
                    drawX(0.0, 0.0, 0.6, 0.0, 0.0, 0.0, 0.7, 0.7, 0.5);
                    drawX(0.0, 0.0, 0.7, 0.0, 0.0, 0.0, 0.6, 0.6, 0.5);
                    drawX(0.0, 0.0, 0.8, 0.0, 0.0, 0.0, 0.5, 0.5, 0.5);
                    drawX(0.0, 0.0, 0.9, 0.0, 0.0, 0.0, 0.4, 0.4, 0.5);
```

```
                        drawX(0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.3, 0.3, 0.5);
                    glPopMatrix();
                glPopMatrix();
            glPopMatrix();


}
void drawNeck () {
            glPushMatrix();
                transform(0.9, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.9, 1.6);
                drawBodyPart();
            glPopMatrix();
            glPushMatrix();
                transform(0.3, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.8, 1.4);
                drawBodyPart();
            glPopMatrix();
}
void drawHead() {
            //Nose
            glPushMatrix();
                drawG(-0.8, 0.7, 0.0, 0.0, 0.0, -52.0, 0.5, 0.5, 0.4);
                drawEyes();
            glPopMatrix();
            //Head - Left
            glPushMatrix();
                transform(0.0, 0.4, -0.9, 49.0, 0.0, 0.0, 1.0, 1.0, 1.0);
                drawC(0.0, 0.9, 0.2, -18.0, 0.0, 45.0, 2.2, 0.8, 1.0);
                drawEyes();
            glPopMatrix();

            //Head - Right
            glPushMatrix();
                drawC(0.0, 0.9, 0.2, -18.0, 0.0, 45.0, 2.2, 0.8, 1.0);
                drawEyes();
            glPopMatrix();

            //Top Jaw
            glPushMatrix();
                transform(0.0, 0.5, 0.0, 180.0, 0.0, 0.0, 0.8, 1.0, 0.7);
                //Teeth - Front
                drawW(-0.9, 0.2, 0.0, 0.0, 90.0, 180.0, 0.4, 0.3, 0.5);
                //Teeth - Right
                drawW(-0.7, 0.2, 0.3, 180.0, 24.0, 0.0, 0.5, 0.5, 0.5);
                drawW(-0.3, 0.2, 0.4, 180.0, 9.0, 0.0, 0.6, 0.3, 0.5);
                drawW(0.2, 0.2, 0.4, 0.0, 6.0, 180.0, 0.6, 0.3, 0.5);
                //Teeth - Left
                drawW(-0.7, 0.2, -0.3, 0.0, 24.0, 180.0, 0.5, 0.5, 0.5);
                drawW(-0.3, 0.2, -0.4, 0.0, 9.0, 180.0, 0.6, 0.3, 0.5);
                drawW(0.2, 0.2, -0.4, 0.0, -6.0, 180.0, 0.6, 0.3, 0.5);
                //Jaw
                drawC(0.0, 0.0, 0.0, 90.0, 0.0, 45.0, 2.2, 1.0, 1.0);
            glPopMatrix();
            //Bottom Jaw Connection
            glPushMatrix();
                drawC(1.1, 0.2, 0.0, 90.0, 27.0, 45.0, 0.6, 1.2, 0.7);
            glPopMatrix();
            //Bottom Jaw
            glPushMatrix();
                transform(0.6, 0.0, 0.0, 0.0, 0.0, joint_jaw, 1.0, 1.0, 1.0);
//Joint Needed
                transform(-0.6, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0);
```

```
            //Teeth - Front
            drawW(-0.9, 0.2, 0.0, 0.0, 90.0, 180.0, 0.4, 0.3, 0.5);
            //Teeth - Right
            drawW(-0.7, 0.2, 0.3, 180.0, 24.0, 0.0, 0.5, 0.5, 0.5);
            drawW(-0.3, 0.2, 0.4, 180.0, 9.0, 0.0, 0.6, 0.3, 0.5);
            drawW(0.2, 0.2, 0.4, 0.0, 6.0, 180.0, 0.6, 0.3, 0.5);
            //Teeth - Left
            drawW(-0.7, 0.2, -0.3, 0.0, 24.0, 180.0, 0.5, 0.5, 0.5);
            drawW(-0.3, 0.2, -0.4, 0.0, 9.0, 180.0, 0.6, 0.3, 0.5);
            drawW(0.2, 0.2, -0.4, 0.0, -6.0, 180.0, 0.6, 0.3, 0.5);
            //Jaw
            drawC(0.0, 0.0, 0.0, 90.0, 0.0, 45.0, 2.2, 1.0, 1.0);
      glPopMatrix();
}


//Draw Main Body-----------------------------------------------
void drawBody() {
      glPushMatrix();
            transform(2.0, spawnhight, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0);
            glPushMatrix();
                  //Draw Back Legs----------------------------------------
-----------
                  glPushMatrix();
                        transform(1.0, -2.0, 0.8, 0.0, 0.0, 0.0, 1.0, 1.0,
1.0);
                        drawRightFoot();
                  glPopMatrix();

                  glPushMatrix();
                        transform(1.0, -2.0, -0.8, 0.0, 0.0, 0.0, 1.0, 1.0,
1.0);
                        drawLeftFoot();
                  glPopMatrix();

                  glPushMatrix();
                        transform(0.9, 0.0, 0.0, 0.0, 0.0, joint_backBody,
1.0, 1.0, 1.0);

                        glPushMatrix();
                              //Draw Back Body----------------------------
----------------
                              glPushMatrix();
                                    transform(0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
4.0, 1.2, 1.2);
                                    drawBodyPart();
                              glPopMatrix();

                              glPushMatrix();
                                    transform(2.2, 0.0, 0.0, joint_tail[1],
joint_tail[0], 0.0, 1.0, 1.0, 1.0);

                                    //Draw Tail---------------------------
-----------------
                                    drawTail();
                              glPopMatrix();

                              glPushMatrix();
                                    transform(-2.6, 0.0, 0.0, 0.0, 0.0,
joint_frontBody, 1.0, 1.0, 1.0);
```

```
                                    glPushMatrix();
                                          //Draw Front Body----------------
-------------------
                                          glPushMatrix();
                                                transform(0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 4.0, 1.2, 1.2);
                                                drawBodyPart();
                                          glPopMatrix();

                                          //Draw Wings---------------------
-------------------
                                          glPushMatrix();
                                                transform(0.0, 0.0, 0.0, -
joint_wings, 0.0, 0.0, 1.0, 1.0, 1.0);
                                                transform(0.9, 0.4, 0.9,
0.0, 34.0, 0.0, 1.0, 1.0, 1.0);
                                                drawRightWing();
                                          glPopMatrix();
                                          glPushMatrix();
                                                transform(0.0, 0.0, 0.0,
joint_wings, 0.0, 0.0, 1.0, 1.0, 1.0);
                                                transform(0.9, 0.4, -0.9,
0.0, -34.0, 0.0, 1.0, 1.0, 1.0);
                                                drawLeftWing();
                                          glPopMatrix();

                                          //Draw Front Legs----------------
-------------------
                                          glPushMatrix();
                                                transform(0.0, 0.0, 0.0,
0.0, 0.0, joint_leftLeg, 1.0, 1.0, 1.0);
                                                transform(-0.5, -2.0, 0.8,
0.0, 0.0, 0.0, 1.0, 1.0, 1.0);
                                                drawRightFoot();
                                          glPopMatrix();
                                          glPushMatrix();
                                                transform(0.0, 0.0, 0.0,
0.0, 0.0, joint_rightLeg, 1.0, 1.0, 1.0);
                                                transform(-0.5, -2.0, -0.8,
0.0, 0.0, 0.0, 1.0, 1.0, 1.0);
                                                drawLeftFoot();
                                          glPopMatrix();

                                          glPushMatrix();
                                                transform(-2.9, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0, 1.0, 1.0);
                                                transform(0.0, 0.0, 0.0,
0.0, joint_neck[0], joint_neck[1], 1.0, 1.0, 1.0);

                                                //Draw Neck----------------
---------------------
                                                drawNeck();

                                                glPushMatrix();
                                                      transform(0.0, 0.0,
0.0, 0.0, joint_head[0], joint_head[1], 1.0, 1.0, 1.0);
                                                      transform(-0.6, -0.7,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0);

                                                      //Draw Head----------
-----------------------
```

```
                                                                drawHead();
                                                        glPopMatrix();
                                                glPopMatrix();
                                        glPopMatrix();
                                glPopMatrix();
                        glPopMatrix();
                glPopMatrix();
        glPopMatrix();
    glPopMatrix();
}

//void fly() {
//    if (flymode) {
//        int i = 0;
//        for (i = 0; i <= 20; i++) {
//
//            if (joint_wings > -20)
//                joint_wings -= 2;
//            if (joint_wings < 36)
//                joint_wings += 2;
//        }
//    } else {
//        flymode = false;
//    }
//}
//Draw Other-------------------------------------------------------
void drawFloor () {
    glPushMatrix();
        glTranslatef(0, -0.05, 0); //draw slightly below y=0 so we can
see grid
        glBegin(GL_POLYGON);
            glColor3f(.75,.75,.75);
            glVertex3f(-10,0,10);
            glVertex3f(-10,0,-10);
            glVertex3f(10,0,-10);
            glVertex3f(10,0,10);
        glEnd();
    glPopMatrix();
}

void drawAxes() {
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_LINES); //x axis
    glVertex3f(-15.0, 0.0, 0.0);
    glVertex3f(15.0, 0.0, 0.0);
    glEnd();

    glBegin(GL_LINES); //y axis
    glVertex3f(0.0, -2.0, 0.0);
    glVertex3f(0.0, 15.0, 0.0);
    glEnd();

    glBegin(GL_LINES); //z axis
    glVertex3f(0.0, 0.0, -15.0);
    glVertex3f(0.0, 0.0, 15.0);
    glEnd();
}

//Input Functions-------------------------------------------------------
void mouseMotion(int x, int y)
{
```

```
	// Called when the Mouse is moved with left button down
	win_theta[0] = pitch0 + (y - mouseY0);
	win_theta[1] = yaw0 + (x - mouseX0);

	glutPostRedisplay();
}

void mouseClick(int button, int state, int x, int y) {
//	 Called on button press or release
	switch (state) {
	case GLUT_DOWN:
		MousePressed = true;
		pitch0 = win_theta[0];
		yaw0 = win_theta[1];
		mouseX0 = x;
		mouseY0 = y;
		break;
	default:
	case GLUT_UP:
		MousePressed = false;
		break;
	}
}
/*
void keyboardDev(unsigned char key, int x, int y) {
	switch (key) {
	case 'z':
		kbrX++;
		break;
	case 'Z':
		kbrX--;
		break;
	case 'x':
		kbrY++;
		break;
	case 'X':
		kbrY--;
		break;
	case 'c':
		kbrZ++;
		break;
	case 'C':
		kbrZ--;
		break;
	case 'v':
		kbrX += 10;
		break;
	case 'V':
		kbrX -= 10;
		break;
	case 'b':
		kbrY += 10;
		break;
	case 'B':
		kbrY -= 10;
		break;
	case 'n':
		kbrZ += 10;
		break;
	case 'N':
		kbrZ -= 10;
```

```
        break;
    case 'a':
        kbtX = kbtX + increment;
        break;
    case 'A':
        kbtX = kbtX - increment;
        break;
    case 's':
        kbtY = kbtY + increment;
        break;
    case 'S':
        kbtY = kbtY - increment;
        break;
    case 'd':
        kbtZ = kbtZ + increment;
        break;
    case 'D':
        kbtZ = kbtZ - increment;
        break;
    case 'q':
        kbsX = kbsX + increment;
        break;
    case 'Q':
        kbsX = kbsX - increment;
        break;
    case 'w':
        kbsY = kbsY + increment;
        break;
    case 'W':
        kbsY = kbsY - increment;
        break;
    case 'e':
        kbsZ = kbsZ + increment;
        break;
    case 'E':
        kbsZ = kbsZ - increment;
        break;
    case ',':
        increment = increment + 0.1;
        break;
    case '.':
        increment = increment - 0.1;
        break;
    case '+':
        win_zome+= 0.2;
        break;
    case '-':
        win_zome-= 0.2;
        break;
    case 'm':
        kbtX = 0;
        kbtY = 0;
        kbtZ = 0;
        kbrX = 0;
        kbrY = 0;
        kbrZ = 0;
        kbsX = 1;
        kbsY = 1;
        kbsZ = 1;
        increment = 0.1;
        break;
```

```
        case ' ':
                printPoints();
                break;
                //Official actions
        case 'p': //Rotate Jaw
                if (joint_jaw < 35)
                        joint_jaw += 5;
                break;
        case 'P': //Rotate Jaw
                if (joint_jaw > 0)
                        joint_jaw -= 5;
                break;
        case 'o': //Rotate Head Y
                if (joint_head[0] < 15)
                        joint_head[0] += 5;
                break;
        case 'O': //Rotate Head Y
                if (joint_head[0] > -15)
                        joint_head[0] -= 5;
                break;
        case 'i': //Rotate Head Z
                if (joint_head[1] < 30)
                        joint_head[1] += 5;
                break;
        case 'I': //Rotate Head Z
                if (joint_head[1] > -35)
                        joint_head[1] -= 5;
                break;
        case 'u': //Rotate Neck Y
                        if (joint_neck[0] < 25)
                                joint_neck[0] += 5;
                        break;
                case 'U': //Rotate Neck Y
                        if (joint_neck[0] > -25)
                                joint_neck[0] -= 5;
                        break;
                case 'y': //Rotate Neck Z
                        if (joint_neck[1] < 100)
                                joint_neck[1] += 5;
                        break;
                case 'Y': //Rotate Neck Z
                        if (joint_neck[1] > -30)
                                joint_neck[1] -= 5;
                        break;
        }
        glutPostRedisplay();
        printPoints();
}
*/
void keyboard(unsigned char key, int x, int y){
        switch (key) {
        case '+':
                win_zome += 0.2;
                break;
        case '-':
                win_zome -= 0.2;
                break;
        case '1':
                flymode = true;
                break;
        //Official actions
```

```
case 'q': //Rotate Jaw
      if (joint_jaw < 35)
            joint_jaw += 5;
      break;
case 'Q': //Rotate Jaw
      if (joint_jaw > 0)
            joint_jaw -= 5;
      break;
case 'a': //Rotate Head Y
      if (joint_head[0] < 15)
            joint_head[0] += 5;
      break;
case 'A': //Rotate Head Y
      if (joint_head[0] > -15)
            joint_head[0] -= 5;
      break;
case 'z': //Rotate Head Z
      if (joint_head[1] < 30)
            joint_head[1] += 5;
      break;
case 'Z': //Rotate Head Z
      if (joint_head[1] > -35)
            joint_head[1] -= 5;
      break;
case 'w': //Rotate Neck Y
      if (joint_neck[0] < 20)
            joint_neck[0] += 5;
      break;
case 'W': //Rotate Neck Y
      if (joint_neck[0] > -20)
            joint_neck[0] -= 5;
      break;
case 's': //Rotate Neck Z
      if (joint_neck[1] < 15)
            joint_neck[1] += 5;
      break;
case 'S': //Rotate Neck Z
      if (joint_neck[1] > -15)
            joint_neck[1] -= 5;
      break;
case 'e': //Rotate Back Body Z
      if (joint_backBody > -70)
            joint_backBody -= 5;
      break;
case 'E': //Rotate Back Body Z
      if (joint_backBody < 0)
            joint_backBody += 5;
      break;
case 'd': //Rotate Front Body Z
      if (joint_frontBody > -10)
            joint_frontBody -= 5;
      break;
case 'D': //Rotate Front Body Z
      if (joint_frontBody < 0)
            joint_frontBody += 5;
      break;
case 'r': //Rotate Wings
      if (joint_wings > -20)
            joint_wings -= 2;
      break;
case 'R': //Rotate Wings
```

```
            if (joint_wings < 36)
                  joint_wings += 2;
            break;
      case 't': //Rotate Left Leg
            if (joint_leftLeg > -20)
                  joint_leftLeg -= 2;
            break;
      case 'T': //Rotate Left Leg
            if (joint_leftLeg < 40)
                  joint_leftLeg += 2;
            break;
      case 'g': //Rotate Left Leg
            if (joint_rightLeg > -20)
                  joint_rightLeg -= 2;
            break;
      case 'G': //Rotate Left Leg
            if (joint_rightLeg < 40)
                  joint_rightLeg += 2;
            break;
      /*case 'f': //Rotate Tail Y
            if (joint_tail[0] < 110)
                  joint_tail[0] += 5;
            break;
      case 'F': //Rotate Tail Y
            if (joint_tail[0] > -70)
                  joint_tail[0] -= 5;
            break;*/
      case 'v': //Rotate Tail Z
            if (joint_tail[1] < 100)
                  joint_tail[1] += 5;
            break;
      case 'V': //Rotate Tail Z
            if (joint_tail[1] > 0)
                  joint_tail[1] -= 5;
            break;
      }
      glutPostRedisplay();
}

void reshapeCallBack(int w, int h){
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
            glLoadIdentity();
            if (w <= h)
                  glOrtho(-4.0, 4.0, -4.0 * (float) h / (float) w,
                        4.0 * (float) h / (float) w, -10.0, 10.0);
            else
                  glOrtho(-2.0 * (float) w / (float) h,
                        2.0 * (float) w / (float) h, -2.0, 2.0, -10.0,
10.0);
//          gluLookAt(0, 0, 1, 0, 0, 0, 0, 1, 0);
    glMatrixMode(GL_MODELVIEW);
}

void viewControll() {
      //Rotate everything
      glRotatef(win_theta[0], 1.0, 0.0, 0.0);
      glRotatef(win_theta[1], 0.0, 1.0, 0.0);
      glRotatef(win_theta[2], 0.0, 0.0, 1.0);

      //zoom (NB glOrtho projection)
```

```
	glScalef(win_zome,win_zome,win_zome);
}

void display() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glLoadIdentity();

	viewControll();

	drawBody();

	drawFloor();
	drawAxes();

//	fly();
	glFlush();
	glutSwapBuffers();

}

int main(int argc, char ** args) {
	printMenu();
	glutInit(&argc, args);
	glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH);
	glutInitWindowSize(WIDTH, HEIGHT);
	glutCreateWindow("CG Project Dragon - Willem Mouton, Charlie van
Zyl"); //create window
	glEnable(GL_DEPTH_TEST);

	glutReshapeFunc(reshapeCallBack);
	glutDisplayFunc(display); //display function

	//Input Functions.
	glutKeyboardFunc(keyboard);
	glutMouseFunc(mouseClick);
	glutMotionFunc(mouseMotion);

	glutMainLoop(); //call registered functions
	return 0;
}
```

Vertices.h

```
/*
 Project    : Computer Graphics OpenGL Semester 1 Project
 Author     : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document   : Vertices.h
 */
float colours[8][3] = {
	{1, 0, 0},
	{0.6, 0, 0},
	{0, 1, 0},
	{0, 0.6, 0},
	{0, 0, 1},
	{0, 0, 0.6},
	{0, 0.6, 0.6},
	{0.6, 0, 0.6},
};
```

```
float vertices[100][3] = { { 0.5, 0.5, 0.1 }, { 0.3, 0.5, 0.1 },
            { 0.1, -0.5, 0.1 }, { 0.3, -0.5, 0.1 }, { 0.0, -0.3, 0.1 },
            { 0.0, 0.0, 0.1 }, { -0.5, 0.5, 0.1 }, { -0.3, 0.5, 0.1 },
            { -0.1, -0.5, 0.1 }, { -0.3, -0.5, 0.1 }, { 0.0, -0.3, 0.1 },

            { 0.5, 0.5, -0.1 }, { 0.3, 0.5, -0.1 }, { 0.1, -0.5, -0.1 },
            { 0.3, -0.5, -0.1 }, { 0.0, -0.3, -0.1 }, { 0.0, 0.0, -0.1 },
            { -0.5, 0.5, -0.1 }, { -0.3, 0.5, -0.1 }, { -0.1, -0.5, -0.1 },
            { -0.3, -0.5, -0.1 }, { 0.0, -0.3, -0.1 },

            { 0.5,0.3,0.1 }, { -0.5,0.3,0.1 }, { 0.5,-0.5,0.1 },
            { -0.5,-0.5,0.1 }, { -0.5,-0.3,0.1 }, { 0.5,-0.3,0.1 },
          { 0.2,0.3,0.1 }, { -0.5,-0.3,0.1 }, { -0.2,-0.3,0.1 },

            { 0.5,0.3,-0.1 }, { -0.5,0.3,-0.1 }, { 0.5,-0.5,-0.1 },
            { -0.5,-0.5,-0.1 }, { -0.5,-0.3,-0.1 }, { 0.5,-0.3,-0.1 },
            { 0.2,0.3,-0.1 }, { -0.5,-0.3,-0.1 }, { -0.2,-0.3,-0.1 },

            { 0.3, 0.3, 0.1 }, { 0.0, 0.2, 0.1 }, { -0.3, 0.3, 0.1 },

            { 0.3,0.3,-0.1 }, { 0.0,0.2,-0.1 }, { -0.3,0.3,-0.1 },

            { 0.5,0.1,0.1 }, { 0.2,0.1,0.1 }, { 0.2,0.0,0.1 },
            { 0.5,0.0,0.1 }, { 0.4,0.0,0.1 }, { 0.4,-0.5,0.1 },

            { 0.5,0.1,-0.1 }, { 0.2,0.1,-0.1 }, { 0.2,0.0,-0.1 },
            { 0.5,0.0,-0.1 }, { 0.4,0.0,-0.1 }, { 0.4,-0.5,-0.1 },

            { 0.3,0.0,0.1 }, { 0.3,0.0,-0.1 },
            { 0.0,-0.5,-0.1 }, { 0.0,-0.5,0.1 },
};

void polygon(int a, int b, int c, int d, int col) {
      glBegin(GL_POLYGON);
      glColor3fv(colours[col]);
      glVertex3fv(vertices[a]);
      glVertex3fv(vertices[b]);
      glVertex3fv(vertices[c]);
      glVertex3fv(vertices[d]);
      glEnd();
}

void curve(float startPoint, float endPoint, float r1, float r2, float a,
            float b, int col) {

      //Variables
      float theta;
      float PI = 3.142;

      glBegin(GL_QUAD_STRIP);
            glColor3fv(colours[col]);
            for (theta = startPoint; theta <= endPoint; theta++) {
                  float thetaRad = theta * PI / 180.0;
                  glVertex3f(r1 * cos(thetaRad), r1 * sin(thetaRad), a);
                  glVertex3f(r2 * cos(thetaRad), r2 * sin(thetaRad), b);
            }
      glEnd();
}
```

LetterW.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterW.h
 */
void front_W () {
      polygon(0, 1, 2, 3, 0);
      polygon(3, 2, 4, 5, 0);
      polygon(5, 10, 8, 9, 0);
      polygon(9, 8, 7, 6, 0);
}

void back_W () {
      polygon(11, 12, 13, 14, 0);
      polygon(14, 13, 15, 16, 0);
      polygon(16, 21, 19, 20, 0);
      polygon(20, 19, 18, 17, 0);
}

void sideOuter_W () {
      polygon(6, 17, 20, 9, 0);
      polygon(1, 12, 13, 2, 0);
      polygon(0, 11, 14, 3, 0);
      polygon(7, 18, 19, 8, 0);
}

void sideInner_W () {
      polygon(9, 20, 16, 5, 0);
      polygon(3, 14, 16, 5, 0);
      polygon(8, 19, 21, 10, 0);
      polygon(2, 13, 21, 10, 0);
}

void top_W () {
      polygon(6, 17, 18, 7, 0);
      polygon(0, 11, 12, 1, 0);
}

void bottom_W () {
      polygon(9, 20, 19, 8, 0);
      polygon(3, 14, 13, 2, 0);
}

void drawW(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
      glPushMatrix();

            glTranslatef(tx, ty, tz);
            glRotatef(rx, 1, 0, 0);
            glRotatef(ry, 0, 1, 0);
            glRotatef(rz, 0, 0, 1);
            glScalef(sx, sy, sz);

            //Front
            front_W ();

            //Back
            back_W ();
```

```
            //Sides - Outer
            sideOuter_W ();

            //Sides - Inner
            sideInner_W ();

            //Top
            top_W ();

            //Bottom
            bottom_W ();

      glPopMatrix();
}
```

LetterC.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterC.h
 */
void outerSurface_C(float r1) {
      curve(0, 270, r1, r1, 0.1, -0.1, 2);
}

void innerSurface_C(float r2) {
      curve(0, 270, r2, r2, 0.1, -0.1, 3);
}

void front_C(float r1, float r2) {
      curve(0, 270, r1, r2, 0.1, 0.1, 2);
}

void back_C(float r1, float r2) {
      curve(0, 270, r1, r2, -0.1, -0.1, 2);
}

void top_C() {

}

void endCap_C() {
      polygon(58, 59, 55, 49, 3);
      polygon(10, 21, 60, 61, 3);
}

void drawC(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
      //Variables
      float r1 = 0.5;
      float r2 = 0.3;

      glPushMatrix();
      glScalef(sx, sy, sz);

            glPushMatrix();
                  glTranslatef(tx, ty, tz);
```

```
                    glRotatef(rx, 1, 0, 0);
                    glRotatef(ry, 0, 1, 0);
                    glRotatef(rz, 0, 0, 1);

                    //G Part 1
                    //Outer Surface
                    outerSurface_C(r1);

                     //Inner Surface
                    innerSurface_C(r2);

                    //Front
                    front_C(r1, r2);

                    //Back
                    back_C(r1, r2);

                    //End Cap
                    endCap_C();

            glPopMatrix();
      glPopMatrix();
}
```

LetterM.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterM.h
 */
void front_M() {
      polygon(6, 7, 9, 25, 7);
      polygon(0, 1, 3, 24, 7);
      polygon(1, 40, 5, 41, 7);
      polygon(7, 42, 5, 41, 7);
}

void back_M() {
      polygon(17, 18, 20, 34, 7);
      polygon(11, 12, 14, 33, 7);
      polygon(12, 43, 16, 44, 7);
      polygon(18, 45, 16, 44, 7);
}

void sideOuter_M() {
      polygon(0, 11, 33, 24, 7);
      polygon(6, 17, 34, 25, 7);
      polygon(7, 18, 20, 9, 7);
      polygon(1, 12, 14, 3, 7);
}

void sideInner_M() {
      polygon(1, 12, 44, 41, 7);
      polygon(7, 18, 44, 41, 7);
      polygon(40, 43, 16, 5, 7);
      polygon(42, 45, 16, 5, 7);
}
```

```
void top_M() {
	polygon(6, 17, 18, 7, 7);
	polygon(0, 11, 12, 1, 7);
}

void bottom_M() {
	polygon(25, 9, 20, 34, 7);
	polygon(24, 3, 14, 33, 7);
}

void drawM(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
	glPushMatrix();

		glTranslatef(tx, ty, tz);
		glRotatef(rx, 1, 0, 0);
		glRotatef(ry, 0, 1, 0);
		glRotatef(rz, 0, 0, 1);
		glScalef(sx, sy, sz);

		//Front
		front_M();

		//Back
		back_M();

		//Sides - Outer
		sideOuter_M();

		//Sides - Inner
		sideInner_M();

		//Top
		top_M();

		//Bottom
		bottom_M();

	glPopMatrix();
}
```

LetterX.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterX.h
 */
void front_X() {
	polygon(6, 7, 24, 3, 6);
	polygon(0, 1, 25, 9, 6);
}

void back_X() {
	polygon(17, 18, 33, 14, 6);
	polygon(11, 12, 34, 20, 6);
}

void sideLeft_X() {
```

```
	polygon(18, 7, 24, 33, 6);
	polygon(17, 6, 3, 14, 6);
}

void sideRight_X() {
	polygon(12, 1, 25, 34, 6);
	polygon(11, 0, 9, 20, 6);
}

void top_X() {
	polygon(6, 7, 18, 17, 6);
	polygon(0, 1, 12, 11, 6);
}

void bottom_X() {
	polygon(25, 9, 20, 34, 6);
	polygon(24, 3, 14, 33, 6);
}

void drawX(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
	glPushMatrix();

		glTranslatef(tx, ty, tz);
		glRotatef(rx, 1, 0, 0);
		glRotatef(ry, 0, 1, 0);
		glRotatef(rz, 0, 0, 1);
		glScalef(sx, sy, sz);

		//Front
		front_X();

		//Back
		back_X();

		//Sides - Left
		sideLeft_X();

		//Sides - Right
		sideRight_X();

		//Top
		top_X();

		//Bottom
		bottom_X();

	glPopMatrix();
}
```

LetterZ.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterZ.h
 */
void front_Z() {
	polygon(6, 0, 22, 23, 1);
```

```
	polygon(24, 25, 29, 27, 1);
	polygon(22, 28, 29, 30, 1);
}

void back_Z() {
	polygon(17, 11, 31, 32, 1);
	polygon(33, 34, 38, 36, 1);
	polygon(31, 37, 38, 39, 1);
}

void sides_Z() {
	polygon(6, 17, 32, 23, 1);
	polygon(25, 34, 38, 29, 1);
	polygon(0, 11, 31, 22, 1);
	polygon(24, 33, 36, 27, 1);
	polygon(28, 37, 38, 29, 1);
	polygon(22, 31, 39, 30, 1);
}

void top_Z() {
	polygon(6, 0, 11, 17, 1);
	polygon(23, 22, 31, 32, 1);

}

void bottom_Z() {
	polygon(25, 24, 33, 34, 1);
	polygon(29, 27, 36, 38, 1);
}

void drawZ(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
	glPushMatrix();

		glTranslatef(tx, ty, tz);
		glRotatef(rx, 1, 0, 0);
		glRotatef(ry, 0, 1, 0);
		glRotatef(rz, 0, 0, 1);
		glScalef(sx, sy, sz);

		//Front
		front_Z();

		//Back
		back_Z();

		//Sides
		sides_Z();

		//Top
		top_Z();

		//Bottom
		bottom_Z();

	glPopMatrix();
}
```

LetterG.h

```
/*
 Project     : Computer Graphics OpenGL Semester 1 Project
 Author      : Willem Mouton (H00180920), Charlie van Zyl (H00180839)
 Document    : LetterG.h
 */
void outerSurface_G(float r1) {
      curve(45, 360, r1, r1, 0.1, -0.1, 4);
}

void innerSurface_G(float r2) {
      curve(45, 360, r2, r2, 0.1, -0.1, 4);
}

void front_G(float r1, float r2) {
      curve(45, 360, r1, r2, 0.1, 0.1, 4);
      polygon(46, 47, 48, 49, 4);
      polygon(49, 50, 51, 24, 4);
}

void back_G(float r1, float r2) {
      curve(45, 360, r1, r2, -0.1, -0.1, 4);
      polygon(52, 53, 54, 55, 4);
      polygon(55, 56, 57, 33, 4);
}

void sides_G() {
      polygon(46, 24, 33, 52, 4);
      polygon(50, 51, 57, 56, 4);
      polygon(47, 48, 54, 53, 4);
}

void top_G() {
      polygon(46, 47, 53, 52, 4);
}

void bottom_G() {
      polygon(24, 51, 57, 33, 4);
      polygon(49, 48, 54, 55, 4);
}

void drawG(float tx, float ty, float tz, float rx, float ry, float rz,
float sx, float sy, float sz) {
      //Variables
      float r1 = 0.5;
      float r2 = 0.4;

      glPushMatrix();

            glTranslatef(tx, ty, tz);
            glRotatef(rx, 1, 0, 0);
            glRotatef(ry, 0, 1, 0);
            glRotatef(rz, 0, 0, 1);
            glScalef(sx, sy, sz);


            //G Part 1
            //Outer Surface
            outerSurface_G(r1);

             //Inner Surface
            innerSurface_G(r2);
```

```
            //G Part 2
            //Front
            front_G(r1,r2);

            //Back
            back_G(r1,r2);

            //Sides
            sides_G();

            //Top
            top_G();

            //Bottom
            bottom_G();

      //End Cap
//    polygon(24, 51, 57, 33, 1);

      glPopMatrix();
}
```