Tunku Abdul Rahman University College

Jaegers Prototype

BACS 2173

Graphics Programming

Student Names	Student ID	Contribution (%)	Sign	Grade (for Tutor)
Ong Jon Shen	18WMR08498			
Ong Tun Ying	18WMR08499			
Total		100%		

Programme : RSF3

Tutorial Group : G5

Date Submitted : 6 JAN 2020

Table of Contents

Table of Contents **2**

Introduction **3**

System Specifications **5**

- Visual Studio 5
- OpenGL 5

Design Concepts **6**

- Front View 6
- Side View 8
- Back View 10

Polygon Count **12**

- Summary 12
- Head 12
- Body 13
- Arm 14
- Leg 15

User Manual **16**

- Camera Controls 16
- Animation Controls 16
- General Controls 16
- Body Controls 17
  - Head and Body 17
  - Arms and Legs 18

Personal Appraisal **21**

- ONG JON SHEN 21
- ONG TUN YING 22

Introduction

In this project, the design team is required to design a Jaeger for the movie “Pacific Rim” for their upcoming sequel. Jaegers are giant robot that have the size about 288ft (87.8m) tall and weighs about 7080 tons. Our team decided to based our new designs on the Cherno Alpha jaeger in Pacific Rim.

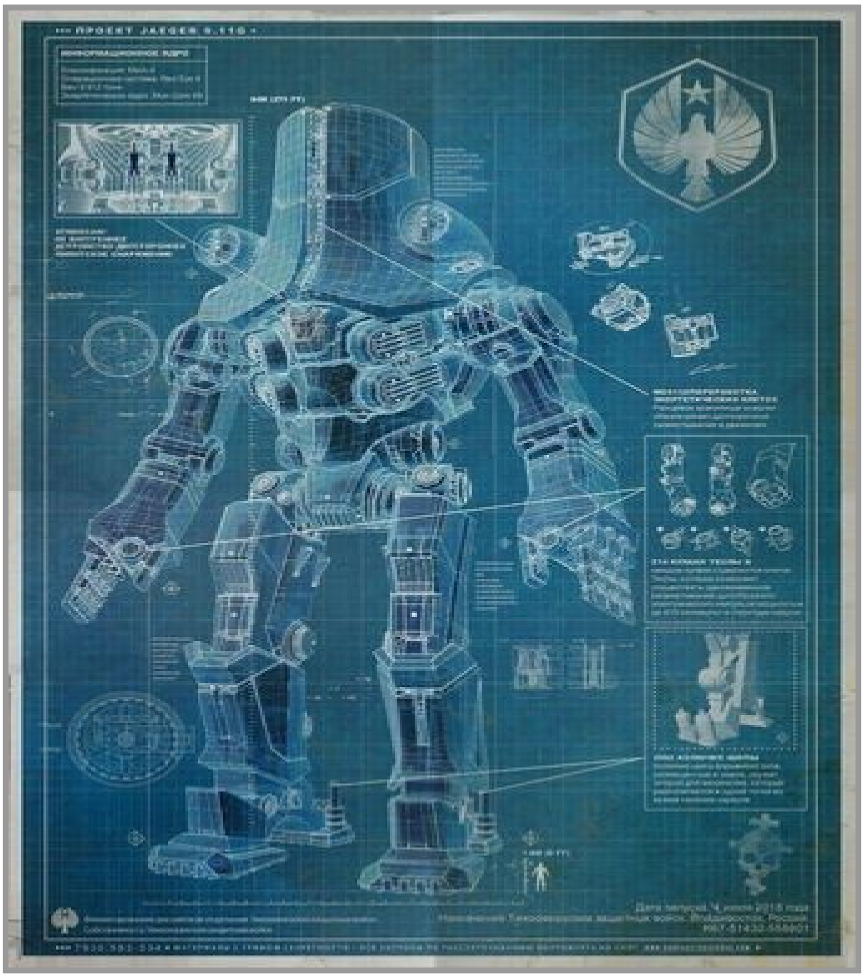


Figure 1: A blueprint schematic of the Cherno Alpha.

The Cherno Alpha was designed and patterned to resemble "Russian military painting schemes", and in particular, Russian tanks. Therefore, heavy additions of armour plating was incorporated into the design. The large head design was created to secure the pilots safety during engagement with the Kaiju. Moreover, a large base was created for the foot of the Cherno Alpha for increased stability as it was heavy.

Our model differs in its attack mechanisms from the Cherno Alpha by incorporating a long sword and shield to pierce and repel any Kaiju attacks. We also added a twin laser blaster for long range strikes with multiple Kaiju. Furthermore, multiple textures were included to explore different themes on the Jaeger.

System Specifications

Visual Studio

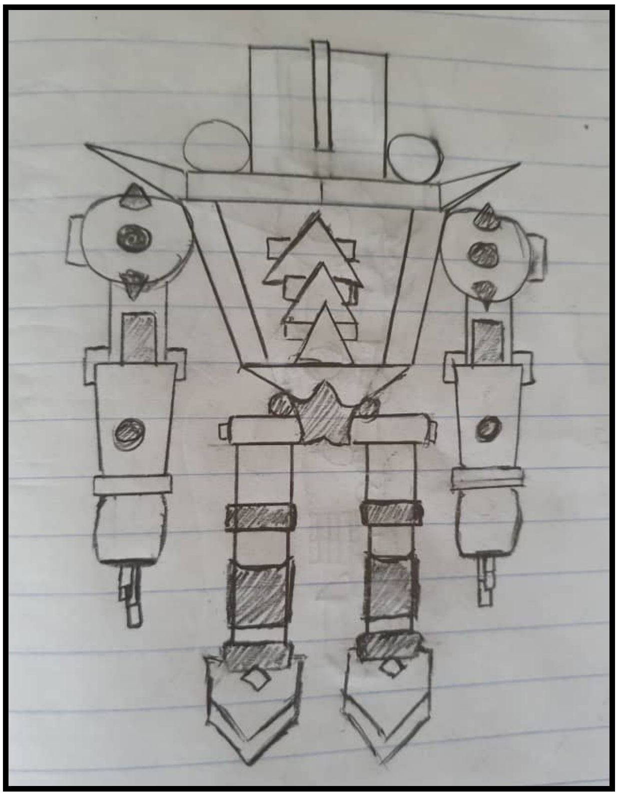
Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation and Microsoft Silverlight. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. In this project, we are using C++ to to run a window from Visual Studio. Visual studio also supports additional libraries like OpenGL.

OpenGL

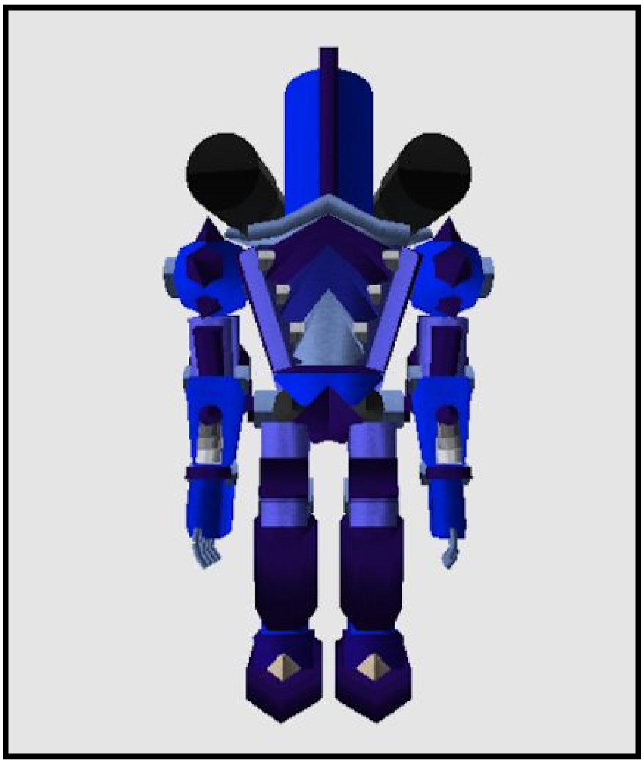
Open Graphics Library (OpenGL) is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering. OpenGL also has extension libraries such as GL, GLU, GLEE, GLEW, GLUT and GLSL to render more realistics graphics, where each libraries manage a different section from the graphics pipeline. In this project, we have used GLU to render our robot. GLU (OpenGL Utility Library) is a graphics library for OpenGL, consisting of utility functions which can be used with OpenGL. The functions mainly focus on primitive rendering and mapping between screen- and world-coordinates, etc. It provided simple, useful features which were unlikely to be supported in contemporary hardware, such as tessellating, and generating mipmaps and primitive shapes. This library especially helped in rendering 3D polygons.

Design Concepts

Front View

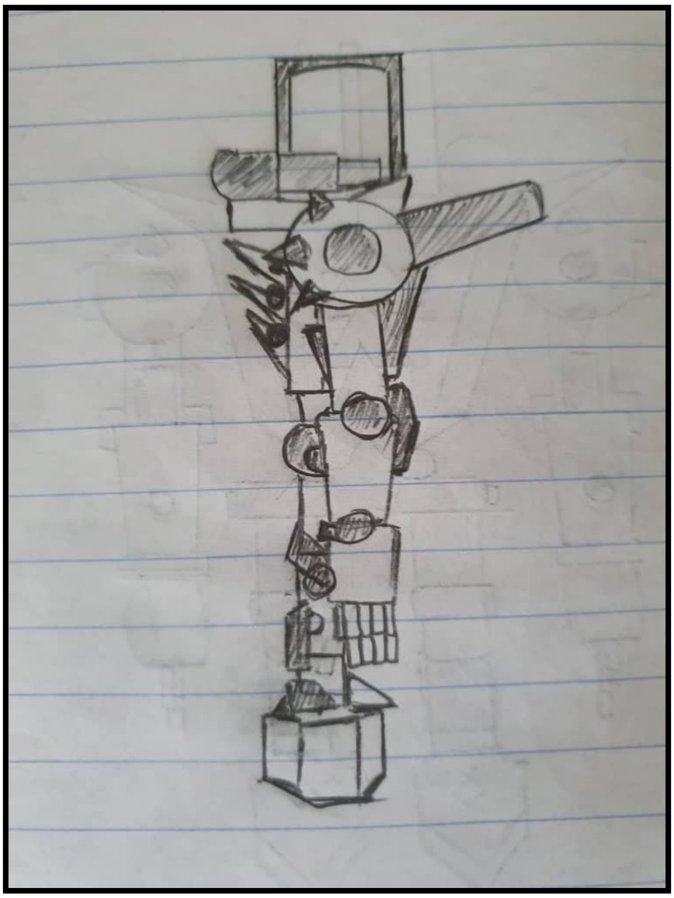


Initial Sketch of the front view.

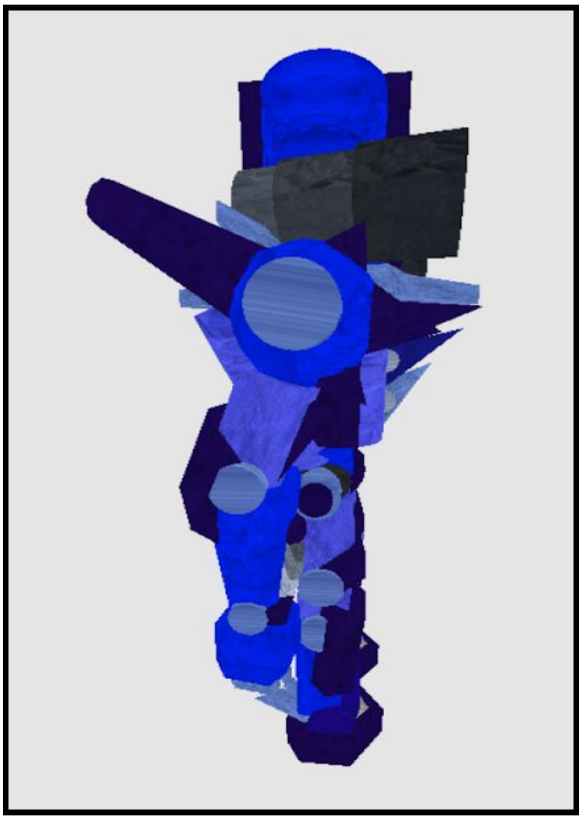


Front View of the Final Model.

Side View

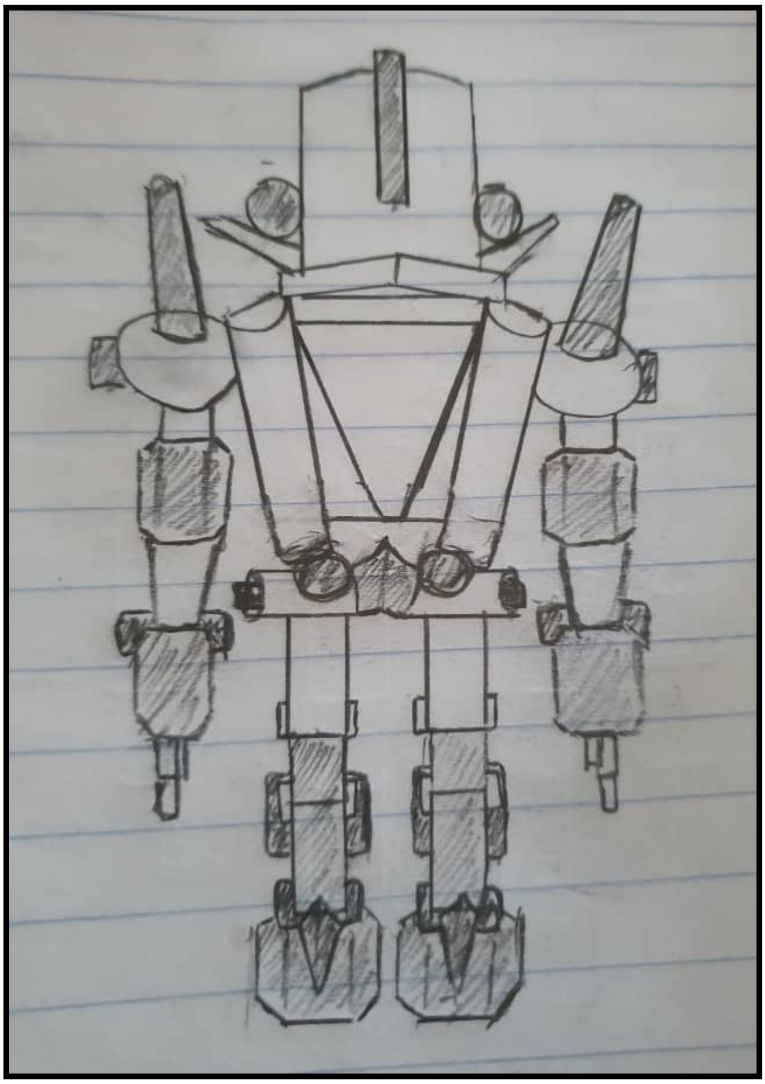


Initial Sketch of the Side View.

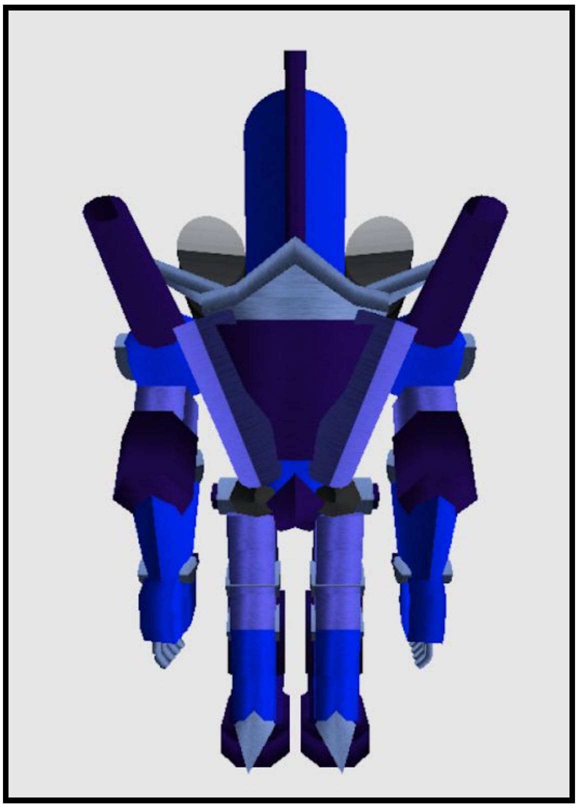


Side View of the Final Model.

Back View



Initial Sketch of the Back View.



Back View of the Final Model.

Polygon Count

Summary

Component	Quantity	Component Polygon Count	Total Polygon Count
Head	1	49	49
Body	1	73	73
Arm	2	164	328
Leg	2	77	154
Grand Total:			604

Head

Part Name	Quantity	Shape Type	Shape Polygon Count	Part Polygon Count
Helmet	1	Closed Cylinder	3	3
Peacock	1	Cuboid	6	6
Head Plate	1	Decagonal (10) Prism	12	12
Plate Wing	2	Triangular (3) Prism	5	10
Laser Launcher Head	2	Closed Cylinder	3	6
Laser Launcher Middle	2	Closed Cylinder	3	6
Laser Launcher Tail	2	Closed Cylinder	3	6
Subtotal:				49

Body

Part Name	Quantity	Shape Type	Shape Polygon Count	Part Polygon Count
Main Body	1	Hexagonal (6) Prism	8	8
Chest Side Tube	2	Closed Cylinder	3	6
Front Chest Plate	3	4-Sided Pyramid	5	15
Chest Middle Tube	3	Closed Cylinder	3	9
Back Chest Plate	1	4-Sided Pyramid	5	5
Back Side Tubes	2	Closed Cylinder	3	6
Bowel	2	Sphere	1	2
Waist	1	Dodecagonal (12) Prism	14	14
Big Waist Pipe	1	Closed Cylinder	3	3
Small Waist Pipe	1	Closed Cylinder	3	3
Butt Bones	2	Cone	1	2
Subtotal:				73

Arm

Part Name	Quantity	Shape Type	Shape Polygon Count	Part Polygon Count
Shoulder	1	Sphere	1	1
Cylinder connector	3	Closed Cylinder	3	9
Spikes	3	Open Cylinder	1	3
Booster cylinder	1	Open Cylinder	1	1
Elbow	1	Hexagonal (6) Prism	8	8
Upper arm	1	Cuboid	6	6
Lower arm	1	Open Cylinder	1	1
Lower arm sphere	1	Sphere	1	1
Lower arm plate	3	Cuboid	6	18
Lower arm connector	1	Cuboid	6	6
Palm	1	Octagonal (8) Prism	10	10
Upper Fingers	5	Octagonal (8) Prism	10	50
Lower Fingers	5	Octagonal (8) Prism	10	50
Subtotal:				164

Leg

Part Name	Quantity	Shape Type	Shape Polygon Count	Part Polygon Count
Bone	2	Sphere	1	2
Upper Leg	1	Cuboid	6	6
Upper Plate	1	Cuboid	6	6
Upper Joint	1	Cuboid	6	6
Upper Cylinder Joint	1	Closed Cylinder	3	3
Lower Leg	1	Cuboid	6	6
Lower Shield Plate	1	Octagonal (8) Prism	10	10
Lower ankle	1	Cuboid	6	6
Lower arm plate	3	Cuboid	6	18
Lower Leg Joint Cylinder	1	Cylinder	3	3
Feet	1	Heptagon (7) Prism	9	9
Lower Feet Front Spike	1	Open Cylinder	1	1
Lower Feet Back Spike	1	Open Cylinder	1	1
Subtotal:				77

User Manual

Camera Controls

Mouse Button	**Response**
Hold Left Button + Drag	Rotate X and Y axis at the origin.
Hold Right Button + Drag	Translate X and Y.
Mouse Wheel Up	Zoom in.
Mouse Wheel Down	Zoom out.

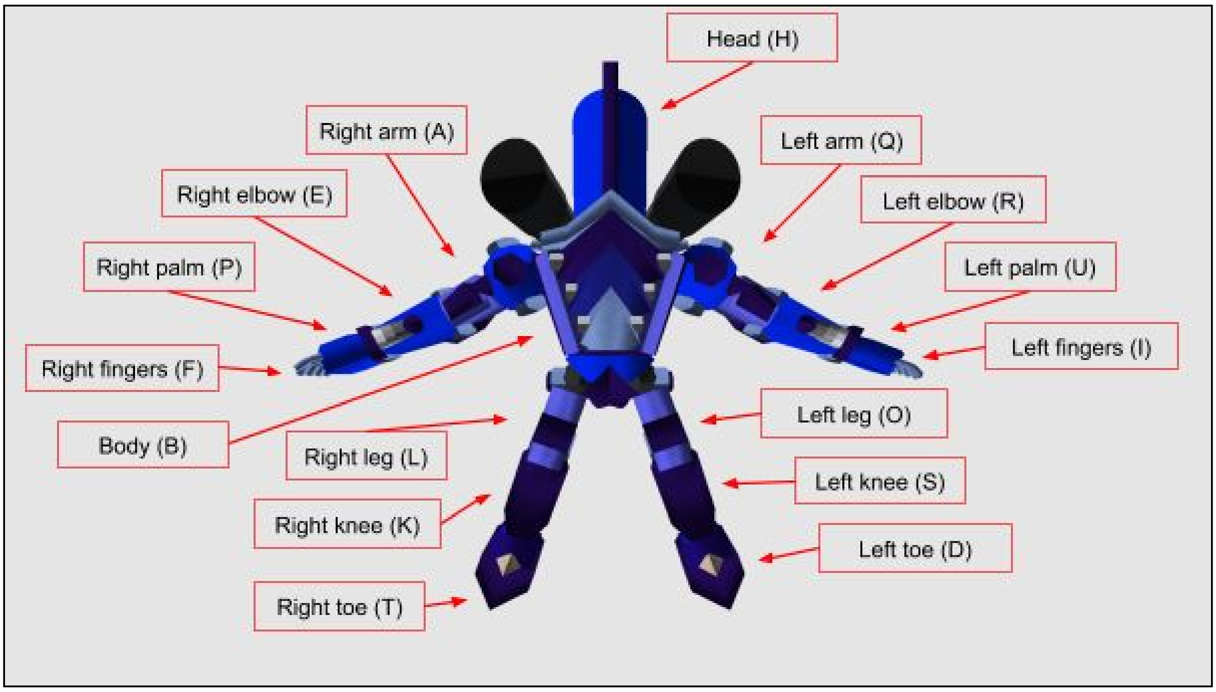
Animation Controls

Key	**Response**
1	Bow.
2	Walk.
3	Fly.
4	Hold sword and shield, then block and attack.
5	Shoot lasers.

General Controls

Key	**Response**
Ctrl	Change robot texture.
Enter / Return	Change background texture.
Spacebar	Reset all.
Tab	Toggle lighting.
9	Toggle perspective/orthographic perspective.
0	Toggle music.
Esc	Quit.

Body Controls



Head and Body

Parts	Keys	Response
Head	H → up	Robot looks upwards
	H → down	Robot looks downwards
	H → left	Robot looks to left
	H → right	Robot looks to right
Body	B → up	Body including arms moves upwards
	B → down	Body including arms moves downwards
	B → left	Body including arms moves leftwards
	B → right	Body including arms moves rightwards

Arms and Legs

Parts	Keys	Response	Parts	Keys	Response
Right Arm	A → up	Move right arms upwards	Left Arm	Q → up	Move left arms upwards
	A → down	Move right arms downwards		Q → down	Move left arms downwards
	A → left	Move right arms leftwards		Q → left	Move left arms leftwards
	A → right	Move right arms rightwards		Q → right	Move left arms rightwards
Right Elbow	E → up	Move right lower hand upwards	Left Elbow	R → up	Move left lower hand upwards
	E → down	Move right lower hand downwards		R → down	Move left lower hand downwards
	E → left	Move right lower hand leftwards		R → left	Move left lower hand leftwards
	E → right	Move right lower hand rightwards		R → right	Move left lower hand rightwards
Right Palm	P → up	Move right palm upwards	Left Palm	U → up	Move left palm upwards
	P → down	Move right palm downwards		U → down	Move left palm downwards
	P → left	Move right palm leftwards		U → left	Move left palm leftwards
	P → right	Move right palm rightwards		U → right	Move left palm rightwards

Right Fingers	F → up	Move right middle fingers outwards	Left Fingers	I → up	Move left middle fingers outwards
	F → down	Move right middle fingers inwards		I → down	Move left middle fingers inwards
	F → left	Move right distal fingers inwards		I → left	Move left distal fingers inwards
	F → right	Move right distal fingers outwards		I → right	Move left distal fingers outwards
Right Leg	L → up	Move right leg upwards	Left Leg	O → up	Move left leg upwards
	L → down	Move right leg downwards		O → down	Move left leg downwards
	L → left	Move right leg leftwards		O → left	Move left leg leftwards
	L → right	Move right leg rightwards		O → right	Move left leg rightwards
Right Knee	K → up	Move right lower leg upwards	Left Knee	S → up	Move left lower leg upwards
	K → down	Move right lower leg downwards		S → down	Move left lower leg downwards
	K → left	Move right lower leg leftwards		S → left	Move left lower leg leftwards
	K → right	Move right lower leg rightwards		S → right	Move left lower leg rightwards
Right Toe	T → up	Move right toe upwards	Left Toe	D → up	Move left toe upwards

	T → down	Move right toe downwards		D → down	Move left toe downwards
	T → left	Move right toe leftwards		D → left	Move left toe leftwards
	T → right	Move right toe rightwards		D → right	Move left toe rightwards

Personal Appraisal

ONG JON SHEN

Firstly, I appreciate the opportunity to have taken this subject (Graphics Programming) that allows to me to enhance my skills and knowledge of the fundamental of graphics. I have learnt a lot in coding polygons and objects into the computer windows, moving around the objects, how objects rendering occurs in the CPU and GPU contexts and also some techniques to make the objects looks more realistics such as adding textures and lightings to the object. Besides the coding itself, the logic behind the code are also very important, I have learned a lot on how to tune each of the vertices and angles to make the robot body and it's movement as smooth and realistics as possible. Also thanks to my teammates that we are able to learn together and assist each other during the program development, else the project will not be accomplished successfully and smoothly.

There are few problems and challenges in doing this assignment. Firstly, the formation of the robot's primitive is not easy, as we have to figure out the coordinates of each vertice of the robot, this process is very time-consuming. Our solution is to add a own-created functions called "my3DPolygon" that can produce a 3D polygon by providing only vertices of one faces into the function. Besides this, positioning a polygon after rotation is time consuming and takes plenty of practise. Also, fitting polygons to match another part proves to be challenging and requires minor adjustments (pixel by pixel). The solution was to translate each new piece through x-axis and the y-axis and leave the z-axis for last (usually leave the z-axis far out to make the piece visible).

Personally, after this assignment I am more appreciative of the computer graphical industry. Therefore, in the future I hope to improve my critical thinking skills in solving unique problems. As I love coding, it is important to have good logic skills in coding programs. Moreover, I would like to learn more about project management skills such as proper time management, schedule planning, efficiency and productivity of working and also communication skills that are required to manage a team effectively.

ONG TUN YING

In this assignment, I have realized that it is important to build an organized project structure for a project that have thousands of lines of code, in our case, approximately 2000 lines. One of the ways that we have tried is to build separate functions of simple objects that can be used to draw just by passing in values. We have also learned to use C++ header files to separate the project to a few source files so that one file does not get too long and cluttered, for example, Main.cpp for all the Windows API codes and some more general OpenGL codes, while Head.cpp for OpenGL codes for the many shapes of components of the robot's head. By doing these, they have greatly eased the process of creating the model and maintaining it while it was being built.

One of the problems I faced was the process of making animations for the model. It was difficult to grasp the right angles and position for the animation to look natural and smooth, the solution was by referencing multiple posing and animation clip and we tried to mimic them.

I have also learned how to use Windows API codes to make a full screen application that also takes the aspect ratio to adjust the projection of the model. In this project, perspective projection was used and I found it necessary (instead of using orthographic projection) because we wanted to simulate a more realistic model for the robot, especially when we use the z-plane extensively to make the robot walk forward and to shoot the lasers. Besides, I have also learned how to play sound effects or music though a C library.

Part of the future plans of the project, for me, is to make the codes, especially the codes on movements and animations, even more modular so that we can easily add animations on the robot. For someone who has used a more modern engine like Unity before, OpenGL is truly an eye-opener for me to the more specific and manual ways of graphic programming, and it was a positive experience because I got to learn the very basics and a lower-level style of thinking and coding graphics.