SECV2213-01

FUNDAMENTAL OF COMPUTER GRAPHICS

ALTERNATE ASSESSMENT PORTFOLIO

LECTURER

DR. GOH EG SU

PREPARED BY

NG JING ER (A19EC0115)

TABLE OF CONTENTS

1 . INTRODUCTION

This portfolio report is a compilation of completed assignments and projects that I have completed throughout this semester for the course SECV 2213 (Fundamental of Computer Graphics). In this report, I will explain about the contents of the assignments and project and make a brief reflection based on what and how I have applied the knowledge learnt from this course. In this portfolio report, there are 4 main sections discussing the 3 Assignments and 1 Project. Most of the assignments and project are using the Microsoft Visual Studio for coding and generate the desired output based on the question. I have more understanding of the fundamentals of computer graphics and applied the technique and skills learnt to create a graphics display using the OpenGL.

2 . ASSIGNMENT 1

2.1 INTRODUCTION FOR ASSIGNMENT 1

Assignment 1 is divided into two parts, which are Assignment 1a and Assignment 1b. Assignment 1a is a group assignment to prepare a simple report about the computer graphics application with the chosen topic “Virtual Reality and Augmented Reality”. The report briefly explained how Virtual Reality and Augmented Reality was used for special effects and the computer graphics term found based on the topic chosen. A short oral presentation is prepared and the findings are shared in the recorded presentation video. While for Assignment 1b is an individual task to produce an OpenGL program with the theme “Scene in UTM Campus”. Some basic OpenGL functions, primitives are used to generate the output that matches with the theme. Basically for the overall Assignment 1 helps the student to have a more understanding and provide a strong foundation of the computer graphics techniques and application.

2.2 SOLUTIONS AND TECHNIQUES USED

For Assignment 1a, we have carried out a simple group discussion using the social media Whatsapp, and delegated the tasks among the group members to do some research on a sample application of computer graphics. The selected application for the title Virtual Reality and Augmented Reality is Oculus Rift S. Based on the findings, some of the computer graphics techniques have been used to create special effects of virtual reality such as 3D Modelling and 3D Transformation.

For Assignment 1b, there are some simple techniques used to draw my design that suits “Scene in UTM Campus”. The scene chosen is the Masjid Sultan Ismail in UTM. Before drawing the scene using OpenGL, I used to draw a draft on a paper first and figure out which function is the most suitable for drawing the different parts of my design. Based on the draft prepared, the building drawn is built up with some simple shapes of rectangles, semicircles and triangles. To draw the shape of the building, glBegin has been invoked to delimit the vertices of the primitives specified such as GL_POINTS, GL_LINES, GL_TRIANGLES, GL_TRIANGLE_FAN, GL_QUADS and GL_POLYGON. Since the drawing is a 2D scene, the glVertex2f() command is used to specify point, line, and polygon vertices. The color of the shapes can be changed by adjusting the parameter inside the glColor3f(). glColor3f() takes 3 arguments with the order: red, green and blue. The current color set for the particular shape coordinates are associated with the vertex value when glVertex is called. All the functions for drawing are compiled in a function called display(). glutCreateWindow() is invoked to create a window named "ASSIGNMENT 1 DRAWING - Scene in UTM Campus ", with the width of 900 pixels and height of 800 pixels.

2.3 RESULTS

After the research on Virtual Reality and Augmented Reality application, Oculus Rift S, I found that Oculus Rift S is a combination of virtual reality and augmented reality gaming headset that is designed to work with a PC. Oculus Rift S is a computer graphic application that promotes the use of virtual reality (VR) which immerses the user in a computer-generated environment that simulates reality. This device uses computer modelling and simulation to enable the user interaction with artificial 3D visuals created.

As a result, I am able to implement the OpenGL primitives that display a simple output of 2D Scene in UTM Campus. Since the design is a 2D scene, therefore glVertex2f is used to specify the x and y coordinates of the vertex, where the coordinate of z is set to zero in default. The initial design of the program for Assignment 1b is too simple. Therefore, the colour of the building is changed using glColor3f which will show different intensity of color and make the output more attractive. The implementation of GL_QUADS instead of GL_POLYGON enables the colors added blended together. The background color is added into the design to improve the visual effect of the output.

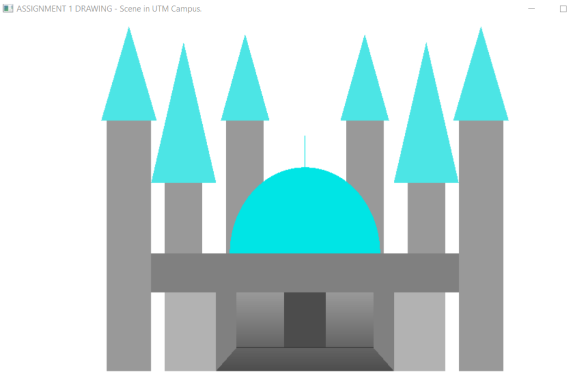

Figure 1: example of initial output

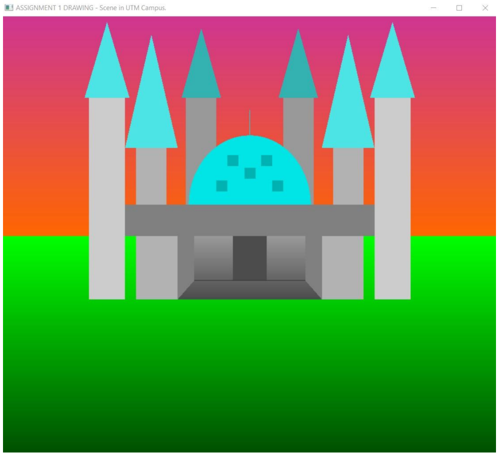

Figure 2: example of final output

2.4 PROPOSED ENHANCEMENT/ IMPROVEMENT

Based on the output generated, the program can be improved by adding some decorations at the surrounding of the building such as trees, grasses, etc. The tree can be formed by drawing a few triangles and rectangles. A simple drawing of grasses can be created by drawing a series of connected lines using GL_LINE or GL_LINE_STRIP. Besides, after getting to know more about the OpenGL functions, the advanced enhancement proposed is to improve the 2D scene into a 3D scene. Some of the simple OpenGL 3D drawing primitives can be used such as glutSolidCylinder, glutSolidCube and glutSolidCone to draw the structure of the building instead of the GL_POLYGON or GL_TRIANGLES which only able to draw the 2D shapes. These functions are based on the GLUT, which is a library of utilities for OpenGL programs. These functions enable us to create geometric objects more easily and faster compared to GL_POLYGON or GL_TRIANGLES.

2.5 CRITIQUE ON NEW SKILLS ACQUIRED

There are some new skills and knowledge that can be acquired through this assignment.First, the assignment exercise helps me to develop my OpenGL foundation. I am able to explore more OpenGL functions such as the techniques of drawing simple shapes such as rectangles, circles, triangles, semicircles, lines and so on. I also acquired the skills of creating objects with multiple colours by using the function of GL_QUADS by changing the parameter values inside the glVertex. By using the OpenGL techniques and skills learnt, I was able to make the output of the 2D Scene in UTM Campus more attractive and interesting with the implementation of a variety of OpenGL functions for creating shapes and color. After understanding and equipped myself with the basic skills and knowledge of the OpenGL functions, I am able to implement those techniques into my assignment and generate a meaningful 2D or 3D scene.

3 . ASSIGNMENT 2

3.1 INTRODUCTION FOR ASSIGNMENT 2

Assignment 2 is done as a group assignment. Assignment 2 contains 2 tasks, where a program is produced to generate output primitives of a line and a circle. By applying the techniques learnt from the previous lab exercise, the program is produced which is able to display an output of a line by using Bresenham's Algorithm learnt in Chapter 3 and the second task output is able to draw a circle by applying the Midpoint Circle Algorithm. In this assignment, the formula of these algorithms has been applied in the code segments, where the program will generate the output based on the input values entered. Assignment 2 involves the implementation of algorithms for graphics primitives and attributes, which will also provide an understanding on how these functions actually work and we can learn to implement the techniques learnt for other applications.

3.2 SOLUTIONS AND TECHNIQUES USED

Assignment 2 is focusing on output primitives. As the requirement stated, both of the tasks include a function named as setPixel that is used to set the pixel/point position. The arguments of the setPixel function include x coordinate and y coordinate. The glBegin() function will invoke the GL_POINTS to draw the point according to every pixel value inside the argument.

There are two different algorithm functions created for drawing line and circle respectively. For task one, we have generated a program that will display the output of a line, using Bresenham's algorithm. Therefore, a function named lineBresenham() is created. The arguments are x and y coordinates of the first endpoint and last endpoint. In the function, some of the variables are initialized to find out the value of constants dx,dy, and the decision parameter p. And we have implemented the algorithm formula, where in the while loop, if p is greater or equal 0, y=y+1; p=p+2*dy-2*dy, which is used to calculate the next coordinate of x,y to be plotted and the next p. Else, p=p+2*dy.

3.3 RESULTS

By implementing the functions we created for task 1 and task 2 in Assignment 2, we are able to display the output of a line and a circle successfully. Based on the results obtained, we can see that Bresenham's algorithm is an efficient and accurate algorithm to generate a line. Bresenham’s algorithm will help to define what is the next point to be plotted to form a line. The coordinate changes along the line occur in unit steps in either the x or y direction. Therefore, Bresenham’s algorithm can be used to obtain the values of successive decision parameters using incremental integer calculations.

For the second task, the functions of circlePoints and circleMidPoint are coding that illustrates how the Midpoint Circle algorithm works. The value of the circle radius and the center coordinates of x and y of the circle to be generated are passed to the function of circleMidpoint. A pixel position along the circular path in the first octant is then computed and passed to the function of circlePoints and finally the output is set to red color in the drawCircle function. In my opinion, this Midpoint Circle algorithm is a time-consuming algorithm since the coding is quite long for the different value of x coordinate of the center.

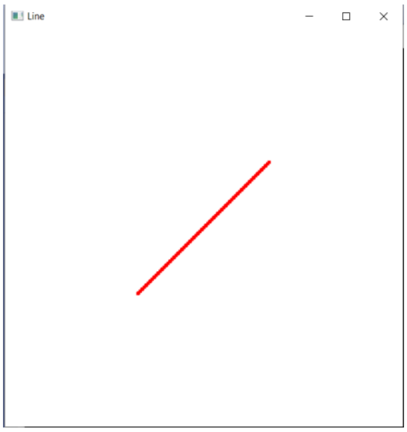

Figure 3: example output of Bresenham’s Algorithm

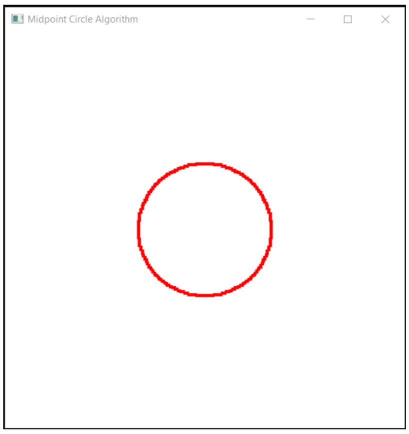

Figure 4: example output of Midpoint Circle Algorithm

3.4 PROPOSED ENHANCEMENT/ IMPROVEMENT

Based on the output generated, the proposed enhancement to improve the program of task 1 in Assignment 2 is to simplify the formula. The long formula written in the code segment can be simplified by initializing them into constant variables such as twody instead of 2*dy; y + + instead of y=y+1. This can help to shorten the code and improve the user’s understanding while reading the code and avoid errors.

For task 2 in Assignment 2, 8-way symmetry is implemented in the circlePoints function. It is 4 times faster compared with 4-way symmetry. The 8-way symmetry is adding symmetries across diagonal lines where x=y. In the functions, we can permute the coordinates of x and y by swapping the position, and simultaneously stepping along y in another part of the circle. However, both the outputs generated by the Bresenham’s Algorithm and Midpoint Circle Algorithm do not provide a smooth and perfect line or shape. Therefore, a suggestion for anti-aliasing or smoothing the straight line segments by modifying the intensity of the pixels is proposed.

3.5 CRITIQUE ON NEW SKILLS ACQUIRED

There are some new skills and knowledge that can be acquired through Assignment 2. In this assignment, we have to create a new function that is required to apply the Bresenham’s algorithm and Midpoint Circle Algorithm in order to get the output. Therefore, I can develop my programming skills which are associated with the idea of graphics drawing algorithms. I am able to create functions for different purposes such as setPixel for setting the pixel position, circleMidpoint to find the value of decision parameter p value and circlePoints to plot the points for the shape. And the function of drawCircle will draw the output of the circle formed by points. By using the theory of algorithms learnt, I was able to make the output line and circle with the implementation of a variety of OpenGL functions. These skills are useful for my future use or project. After understanding and equipped myself with the basic skills and knowledge of the Bresenham's, Midpoint Circle algorithms and OpenGL functions, I am able to implement those techniques into my assignment and generate the output.

4 . ASSIGNMENT 3

4.1 INTRODUCTION FOR ASSIGNMENT 3

Assignment 3 is a group assignment to design and code a 3D character of a cartoon that moves hierarchically. There are several requirements that need to be implemented in order to create a 3D character output. The cartoon chosen to be the 3D character in our Assignment iis the SpongBob. The reason for choosing SpongeBob is due to the simple structure of SpongeBob and it fulfills the requirement of hierarchical modelling. In this assignment, I have leant to implement the techniques of 3D object modeling, hierarchical modeling and suitable 3D transformations on the model (SpongeBob) that we created. Assignment 3 involves a lot of techniques that we learnt before, where we have to make use of the OpenGL functions properly to design and build our model successfully.

4.2 SOLUTIONS AND TECHNIQUES USED

In this assignment 3, the techniques used include 3D object modelling, hierarchical modelling and 3D transformation in order to create a 3D model of a cartoon. The cartoon chosen is SpongeBob as the shape of the model is much easier and fulfils our design requirements.Since the structure of the model is built with cube or cuboid shape, therefore we implemented two different methods which use 3D primitives or by creating 3D objects to build our SpongeBob model. The model created is a hierarchical model with 1 main character (upper yellow body of SpongeBob) and few sub characters (bottom part, hands and legs). The sub characters are joined to the main character. The transformation was also implemented in the assignment using openGL functions such as glRotatef and glTranslatef to perform rotation of angle degrees around the vector x,y,z which adjusted according to our output requirement. glSolidCube is implemented to create the legs that are cuboid in shape. While the body part uses GL_POLYGON which requires the use of glVertex3f to set the position of coordinates to enable us to adjust the color of a particular face of the subpart.

4.3 RESULTS

The program is able to display the output of a simple SpongeBob 3D model successfully by implementing all the functions created. The hand of the SpongeBob model is moveable and requires interaction when displaying the output. The user can see the movement of SpongeBob's hand when pressing different buttons to adjust the movement hand. The body and legs are in cube shape and cuboid shape respectively. For the legs part, glSolidCube is implemented instead of GL_POLYGON. This is an efficient way as it helps us to create our desired 3D Model easily and faster compared to other methods.

The initial design of the program for Assignment 2 is too simple which only displays the model of SpongeBob. Therefore, the background design is added to improve the visual effect of the output. For example the background color added is based on the real cartoon scene, and the ground is created using a few semicircle shapes to imitate the sandhills.

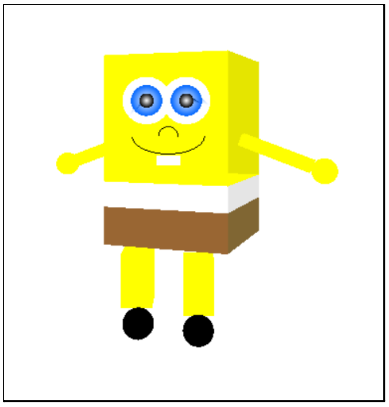

Figure 5: example of initial output

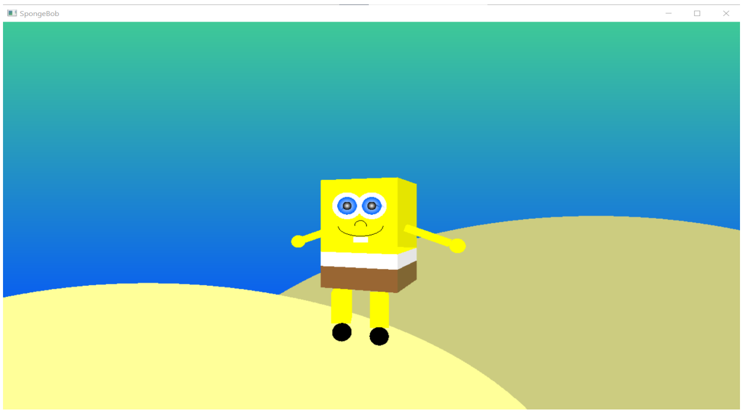

Figure 6: example of final output

4.4 PROPOSED ENHANCEMENT/ IMPROVEMENT

Based on the output generated, it can be observed that there are some improvements that can be made to generate a more attractive output. First, the proposed enhancement to improve the program by adding some decorations at the surrounding of the SpongeBob. The output seems very empty since only the 3D model of SpongeBob is displayed at the middle of the window.

Besides, since the size of the SpongeBob model is too small when displayed with the window size set, I think the window size can be adjusted smaller or the model size of SpongeBob can be adjusted to a bigger size, which will improve the visual experience.Next, some simple GLUT functions can be used instead of the long code segments to create a particular shape. For example the eyes are created using GL_TRIANGLE_FAN with a very long code. By using glutSolidSphere and adjusting the z coordinate in glVertex3f(), the same effect can be obtained easier with less code and effort.

4.5 CRITIQUE ON NEW SKILLS ACQUIRED

In this Assignment 3, there are a few new skills that can be acquired. In the program generated, we have created a few functions for different subparts. Some of the parts such as the body are created using GL_POLYGON while others are glutSolidCube. This is because we found that the method of glutSolidCube is much easier to create the shape. In the process of improving the program, I have more understanding of the OpenGL functions and acquired the skills on selecting the most efficient and appropriate function for different purposes. It also required my thinking skills associated with the OpenGL technique to compute the idea of design into an interesting graphics drawing.

Besides, we are also required to refresh and reuse the techniques that were learnt before such as how to implement the user interaction inside our program. Problem solving skills are really important for me to debug the error that happened when executing the program. After understanding and equipped myself with the strong foundation of skills and knowledge about graphics drawing and OpenGL functions, I am able to implement those techniques into my assignment and generate the 3D Model of SpongeBob.

5 . PROJECT

5.1 INTRODUCTION FOR PROJECT

The final project is an extension of Assignment 3 with a 3D character of a cartoon that moves hierarchically. However, there are some additional requirements that need to be implemented such as the use of lighting and shading and also the implementation of projection and camera models. Therefore the overall project, I have leant to implement the techniques of 3D object modeling, hierarchical modeling and suitable 3D transformations on the model (SpongeBob) that we created in Assignment 3, with the addition of lighting and shading effects, projection and camera model. Functions are created to generate the output that fulfil the requirements. In this final project, we have gathered most of the skills and techniques learnt as much as possible to generate the most satisfying and interesting output.

5.2 SOLUTIONS AND TECHNIQUES USED

In this project program, we applied a variety of GLUT functions and techniques that we had learnt in previous exercises in order to create an attractive output of the project. The basic transformations such as rotate, translate, and scale are implemented to transform the coordinate position of the objects. Besides, to fulfil the requirements, lighting and shading techniques are implemented in this project. Function such as GL_AMBIENT is specially for ambient colours. The ambient light intensity can be changed by adjusting the ambient surface color in the function. While GL_DIFFUSE is for diffuse colours, where diffuse light intensity is adjusted. In the project, GLfloat lightColor0 and lightColor1 are the variables of GL_POSITION that are needed to control the light position.

The other technique used is lighting. The shading mode implemented in this project is Flat Shading. The glShadeModel(GL_FLAT) function is used for shading. This function enables the flat shading selects the computed color as we entered, into the function assigning to the pixel fragments generated. The lighting effect will be displayed if the function is set to be enabled, else the colour of the SpongeBob model will back to the original colour when disabled.

The next technique is the implementation of projection. In the program, we have declared some variables that are required to enable the projection effect. Which includes the angle of the rotation in the y-axis to rotate the camera (angle), vector representing the camera position in X-Z plane (lx,lz) or X-Y plane (ly,lz) and the position of the camera. The viewing transformation of the camera model is defined using the gluLookAt function.

5.3 RESULTS

With the lighting effect implemented, the user is able to view different effects of the SpongeBob model that are displayed when they press different inputs from the keyboard. The output displayed becomes more funny, attractive and realistic with the implementation of lighting and shading. These techniques successfully make the output of the 3D SpongeBob model become more realistic. With the aid of implementation of projection, the user will be able to see a different perspective of view by changing the angle of the camera (screen) using the mouse control to scroll the screen in four directions (left/right/up/down). A function PlaySoundA() is created which enables the user to display an output of the closing theme song of Spong Bob when pressing the keyboard button ‘O’.

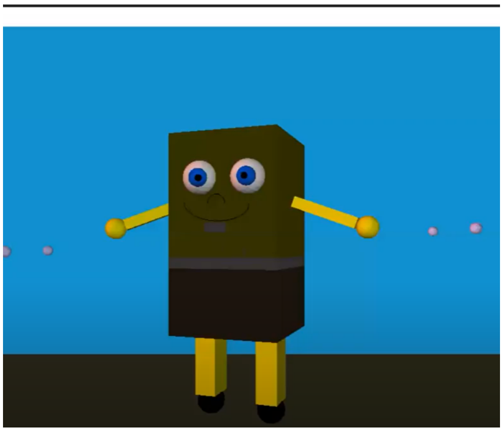

Figure 7: example output of glEnable(GL_LIGHTING)

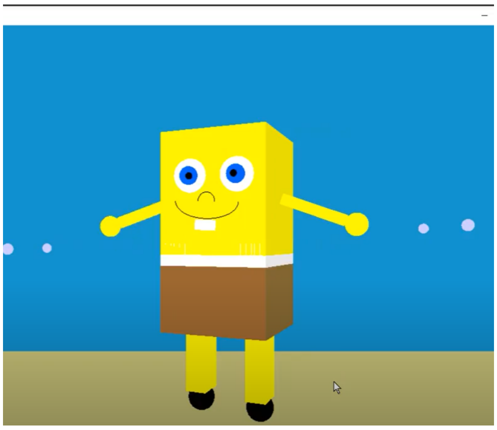

Figure 8: example output of glDisable(GL_LIGHTING)

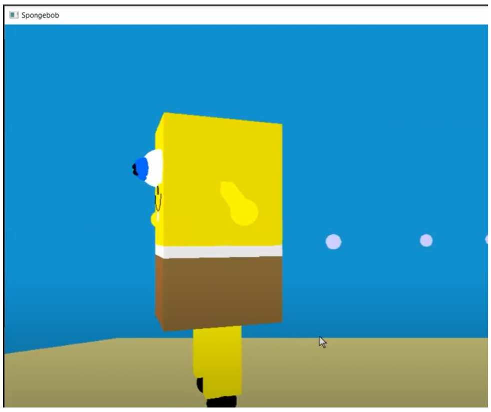

Figure 9: example output from left side view

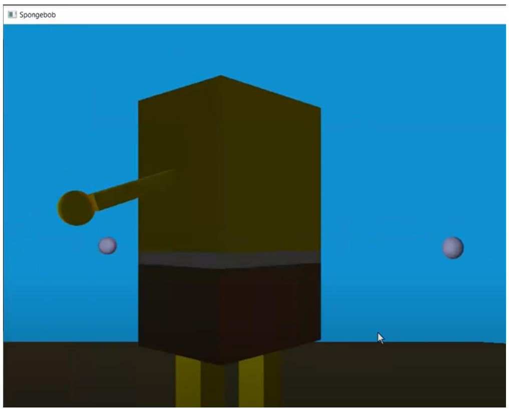

Figure 10: example output from left back view

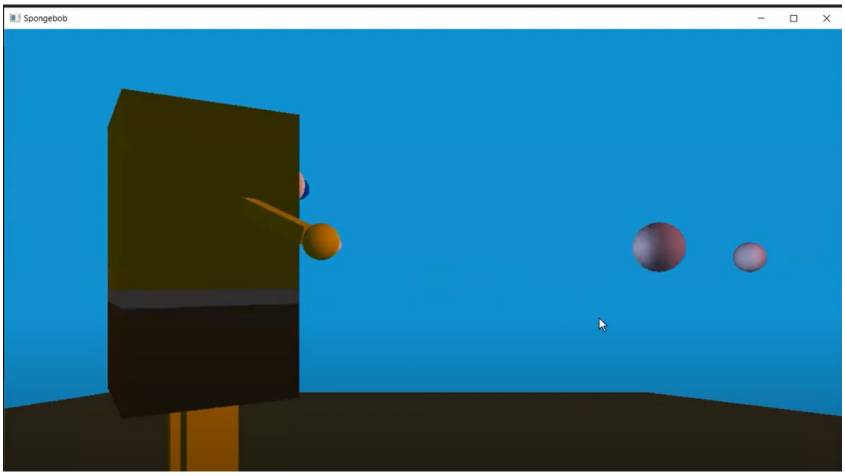

Figure 11: example output from right side view

5.4 PROPOSED ENHANCEMENT/ IMPROVEMENT

The SpongeBob model design is made up of different parts, including body, face,bottom, hands and legs. Based on the program generated in previous Assignment 3, we have made some improvements. Since there are some subparts of the model that are created directly in the display() function, it causes a lot of confusion when we add other parts. Therefore, all the subparts are drawn based on the methods created according to their function name to avoid errors.

As shown in the output, there are some improvements that can be made. First, the proposed enhancement to improve the program is by enhancing the details of the SpongeBob model. The nose and the mouth part of the SpongeBob is created using 2D shape and line. To make the face of the SpongeBob more realistic, the 3D graphic functions can be applied to create 3D shapes instead of 2D shapes. The sphere shape of SpongeBob hands can also be replaced by a more detailed fingers structure. Besides, the output is a bit empty and lacks decorations. By adding some decorations to the surroundings such as the house of SpongeBob, the output will be more attractive and interesting.

5.5 CRITIQUE ON NEW SKILLS ACQUIRED

In this Project, we have to gather all the skills acquired from the previous assignment exercises. Besides, the new skills and techniques in implementing the lighting function and shading function are acquired. In the program generated, we have created some special function for these lighting and shading effects. We have adjusted the ambient color and diffuse color that suits our output.

The next skill acquired is the technique of applying projection into our program. After understanding how the projection works, we managed to apply the projection and camera model. A 3D environment is created and the user can change the view point using the mouse click and scroll to left, right, up and down. Since the code of this project is quite long, we have to develop our problem solving skills to identify and debug the errors in our program. The team working is very important for completing the project. Good communication is required among the members when carrying a group discussion.

6 . SUMMARY OF ASSIGNMENT/PROJECT COVERAGE

	INTRO -DUCTION	GRAPHICS SYSTEM OVERVIEW	OUTPUT PRIMITIVE	COVERAGE
ASSIGNMENT 1	/	/	/	- Current computer graphics techniques and computer graphics application - Understand special effects of computer graphics and Computer graphics term - Indirectly requires us to study and understand graphics system - Require us to explore the OpenGL functions (simple OpenGL primitives, colours and syntaxes) - Require us to generate simple output of 2D/3D scene using OpenGL
ASSIGNMENT 2		/	/	- The algorithm relates to graphics drawing of points, line, ellipse and circle. - Implemented the algorithm of DDA Line Algorithm, Bresenham's Line Algorithm and Midpoint Circle Algorithm - Identify and reproduce graphics concepts through graphics programming using standard graphics libraries.
ASSIGNMENT 3	/	/	/	- The OpenGL functions relate to 3D primitives or 3D objects. - Implemented 3D object modeling, hierarchical modeling and suitable 3D Transformations - Require us to generate a simple 3D model that is able to move hierarchically using OpenGL.
PROJECT	/	/	/	- Implemented 3D object modeling, hierarchical modeling with proper camera, lighting and shading model using OpenGL. - Require us to generate a simple computer graphics project with a 3D model that moves hierarchically and properly uses projection, lighting and shading. - Require us to explore the OpenGL functions to display sound effects, interactivity using mouse /menu

7 . CONCLUSION

In conclusion, all the assignments and projects given provide a good opportunity to me about the fundamentals of computer graphics and its applications. I have a better understanding of how a computer graphics system works and manage to develop a simple graphics program using standard graphics libraries. The process of completing Assignment and project provided me a very good opportunity to explore and discover the variety of functions of OpenGL,graphics drawing algorithms, . After going through all the exercises and lessons, we are able to generate a simple program that will display a 2D scene and even a 3D scene with projection.

8 . REFLECTION

By implementing all the skills, knowledge and techniques learnt before, we are able to generate a desired output of SpongeBob in a 3D environment. Throughout the Assignments and project, we knew that all the functions of OpenGL can be linked together and applied in one program such as 3D object modelling, hierarchical modelling and 3D transformation to create our main character, while the functions of lightning and shading are able to give a brightness changing effect on object created and 3D effect. Besides, projection and camera models can be used to set up and view the surroundings of the object created. Adding some user interaction will make the program more interesting. The program is able to execute successfully without any errors, but still some improvement is needed to enhance the visual effect of the output. Since all the group discussions are carried out using Whatsapp, it is challenging to complete all the tasks within the limited time. In my opinion, a strong foundation in OpenGL and graphics drawing algorithms is really important. All the functions in the code segments are related to each other and we must have fully understood in order to complete the project.