
# SESSION 2018/2019 SEMESTER 2
# ASSIGNMENT 3:
## HIERARCHICAL MODELLING

**NAME**:

1) Tan See Jou (A17CS0218)

2) Ong Le Foo (A17CS0189)

**LECTURER NAME:** NORHAIDA BTE MOHD SUAIB

**SUBJECT NAME:** FUNDAMENTAL OF COMPUTER GRAPHICS

**SUBJECT CODE:** SCSV2213

**SECTION:** 01

# Contents

# 3D Character Hero Modelling – “Earth Baby”

**Computer Graphics Design**

# INTRODUCTION

This is the third assignment given for Fundamental of Computer Graphics class. This assignment is about designing a hierarchical model with at least three levels of hierarchy, and the theme for this year assignment is about designing a Malaysian superhero.

First, we are required to explore an available software, for example, Maya, Blender and so on to model the hierarchical model with complete rigging. After that, present the hierarchical model using simple primitives with the help of OpenGL functions. The presentation needs to include proper 3D transformation for the related parts or segments.

After the discussion, we had decided to choose Maya as the required software to explore. The reasons we chose Maya is because it has the functionality of handling the whole pipeline, from modelling through rigging, animation and rendering. Besides, Maya is also one of the software that is widely used by a lot of studios around the globe. Therefore, we think that investing time in learning and master Maya will definitely pay back in the future.

# SUPERHERO

The Malaysian superhero for our group is “Earth Baby”.

“Earth Baby” is a green environmental friendly hero. He is a superhero who has the superpower to recycle everything and give the unused things a new life. Besides, the flying ability of “Earth Baby” made him easier to reduce most of the trash especially the waste in the ocean.

The reasons we choose a green environmental friendly hero as our Malaysia superhero is due to the low environmental awareness among Malaysian. As we know, our planet has been filled by a lot of waste, and recycle is one of the effective solutions to it. It not only converting unused material into something useable but also decrease the universal carbon emissions. Therefore, recycling is important if we want to protect and provide a clean world for our prospect generations. This is also the reason we design our hero as a baby rather than a strong muscular man or women, as we want emphasis on the importance of recycling and environmental awareness for our future generation.

# Maya Modelling

## Modelling

"Earth Baby" basically is made up of several primitives. First, we model the head, upper body and lower body separately. In modelling the head part, three polygons had been used for the head and ears as well. After that, the whole upper body is made up of a polygon. Both of the hands are extended from the faces of the polygon. For the lower body, few spheres had been used to model the hip, thigh, shank and the shoes. After finished modelling all of the parts, combined all into an object so that easier for the next step. Lastly, we had use 3 polygons to model the recycle symbol and combine it into an object.

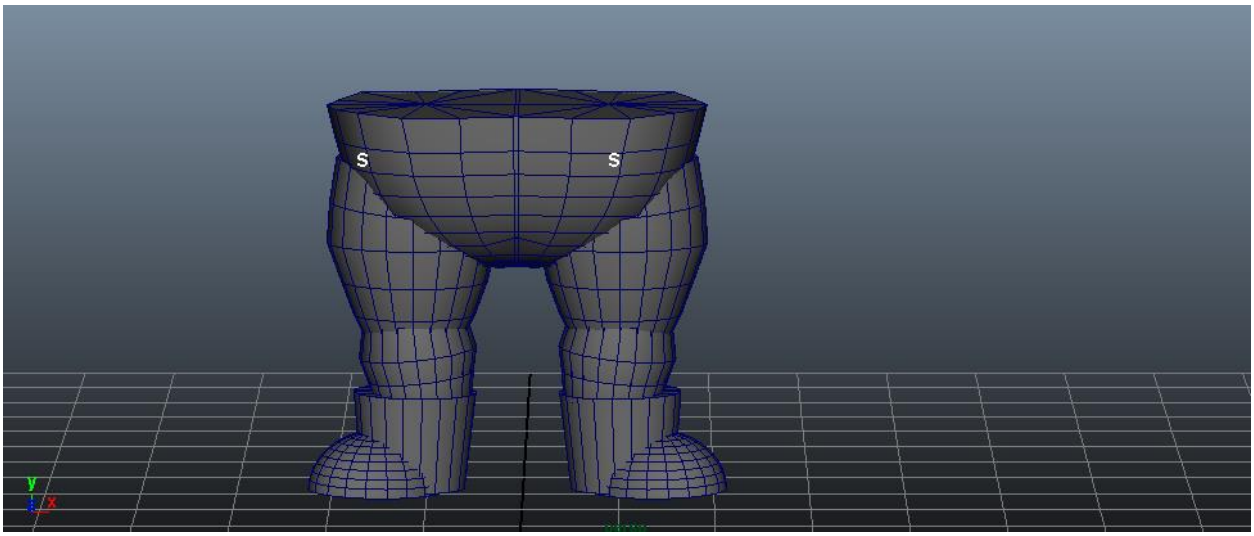

Diagram 1 shows the modelling process of the lower body.

## Rigging

3D rigging is the process of creating a skeleton for a 3D model so that enable it to move. Most commonly, a model need to rig before it can be animated. Joints are like bones in human body, it is the points of articulation created to control the model. For example, if the character need to move it arm, we need to place a joint for the upper arm, another joint for the elbow and another joint for the wrist, in order to allow the character to rotate the arm. We had set some of the parts of the joint in inverse kinematic mode, so that the child

joint can influence the movement of its parent. For example, we had applied IK for the Earth Baby's arm, when we move the Earth Baby's hand, the rest of the arm chain will be influence as well.

## Skinning

Skinning is the process of taking the joints of the rig and binding them to the actual 3D model. When the joints are bound to the model, it allows the model to move the with the joints. Without skinning process, the joint will have no influence on the model. Thus, when we complete the rigging process, we start skinning process.

## Weight painting

After binding, we try to move the character arm, but we found out that some part of the character faces is get influence by the movement of the arm. Therefore, weight painting is needed. Weight painting is a process that allow us to manually set the influence weight on a particular area of the model and correct the deformation on the model.

## Painting Texture- hypershade

We had use hypershade to add on color material on the model.

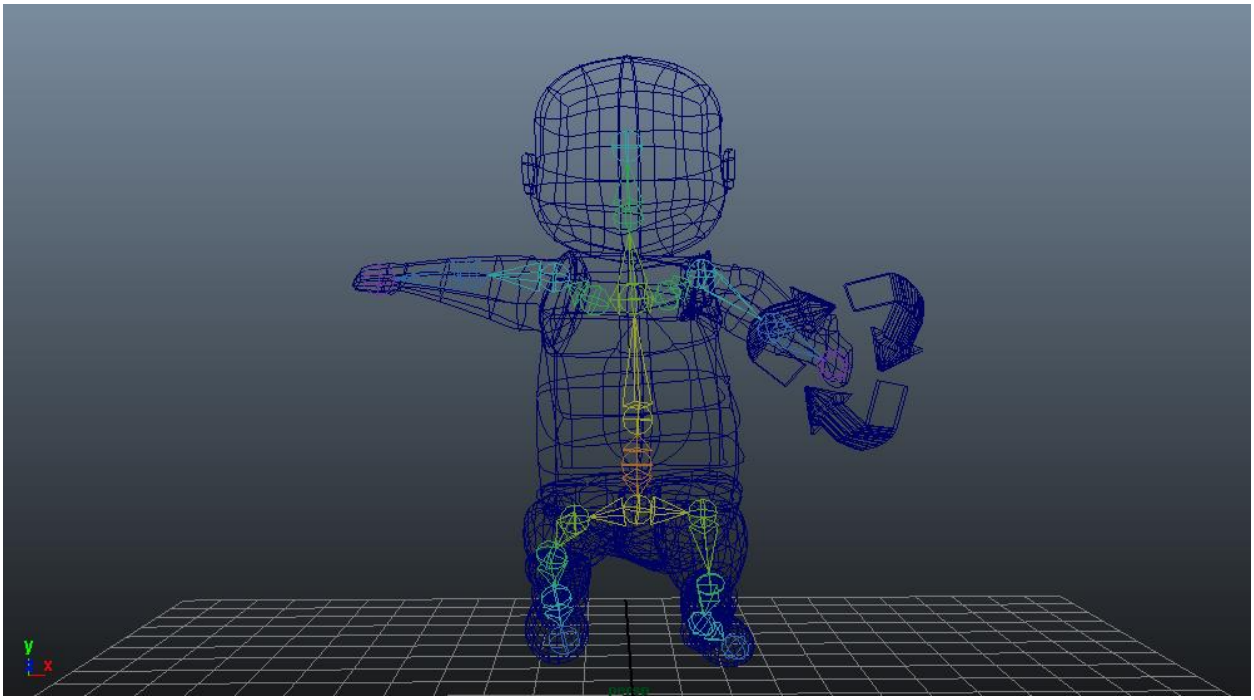

Diagram 2 shows the output that after process rigging and painting skin weight.

## Animation

The last step is to animate the model. We had set a few translations on different frame for the “Earth baby” to move up and down so that it give us a flying effect. After that, we set a few rotations on different frame for the recycle logo so that it can rotate.

## Final Output

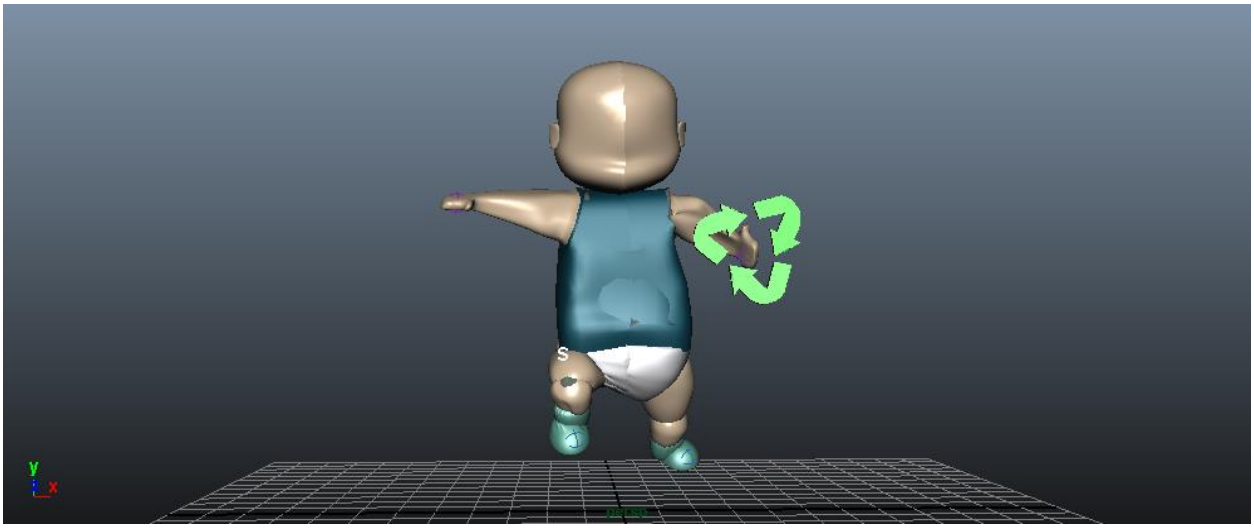

Diagram 3 shows the front views of “Earth Baby”

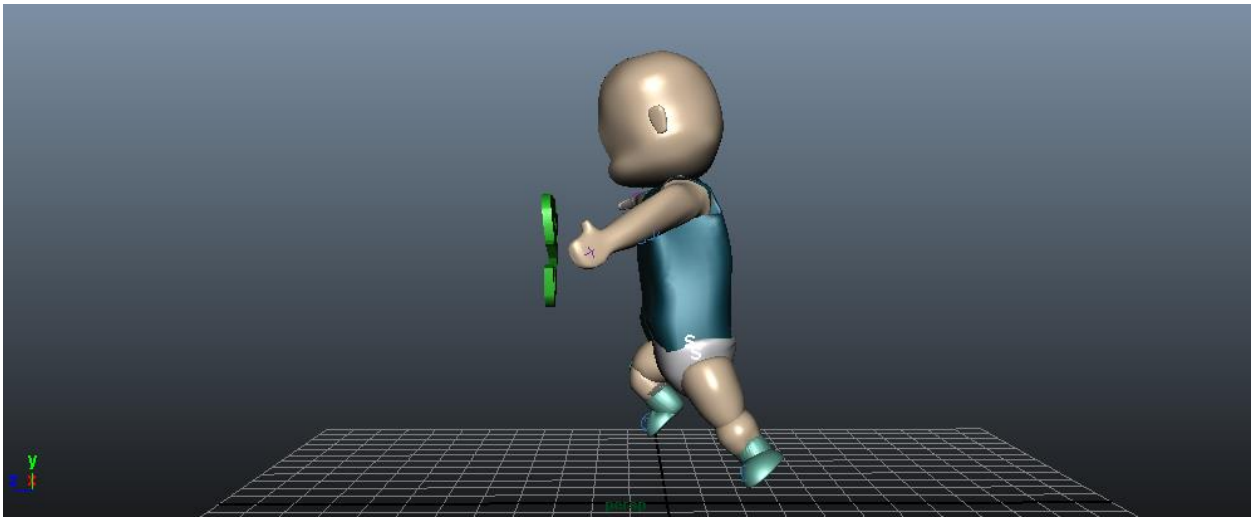

Diagram 4 shows the side views of “Earth Baby”

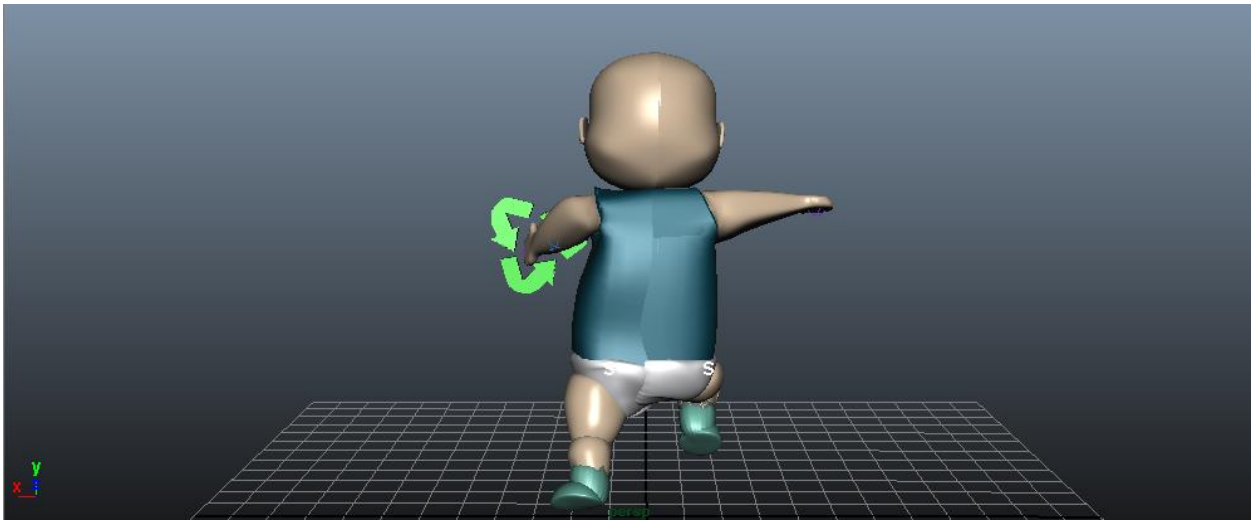

Diagram 5 shows the back views of “Earth Baby”

# OpenGL Modelling

## Character Design

The process start with using multiple 3D output primitives to draw our character. Before we make our model, we design our character with a pen and paper first. During the process of drawing the character in OpenGL, we realize there is a lot of difficulty due to the limitation of the 3D primitive shape. Hence, we decide to make a simple model that look like our character design.

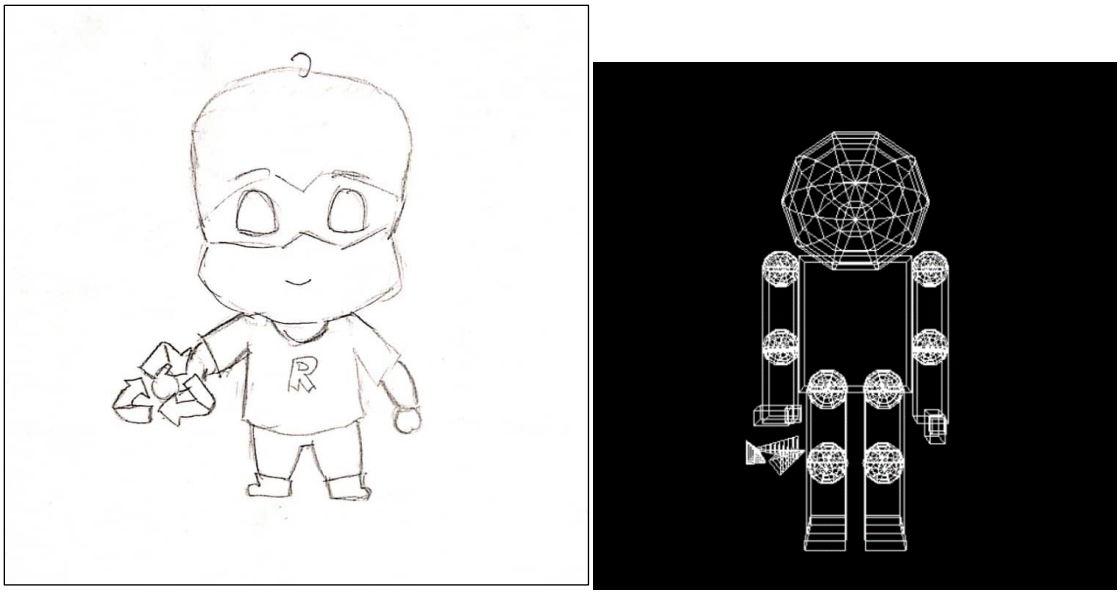

Diagram 6 shows our the draft design and Character Design aka Recycle Baby in OpenGL

The shapes that we had used included glutWireSphere, glutWireCube, and glutWireCone. The reason that Wire is used instead of Solid is because that this assignment doesn't need to include shading and lightning yet. If solids are being used, the detail of the character will be hard to be identified without the shading and lightning. Spheres are being used to draw the head and the joint of every limb. Cube are being used to draw the body and limb structure. Cones are being used to draw the Recycle Logo particle on his hand, we were unable to make it look like the actual recycle logo. Therefore, a triangle looking the logo is made.

In the beginning, every shape is separated individually by using glPushMatrix and glPopMatrix. The glPushMatrix and glPopMatrix can help to separate the transformation of each shape, hence the transformation is independent to the transformation of other shapes. Translation and scaling are being used to position and resize the 3D shape primitive.

```
//left arm
glPushMatrix();
    glTranslatef(-0.8, 0.15, 0);
    glScalef(0.65, 1.75, 0.8);
    glutWireCube(0.5);
glPopMatrix();
```

```
//left forearm
glPushMatrix();
    glTranslatef(-0.8, -0.65, 0);
    glScalef(0.65, 1.5, 0.8);
    glutWireCube(0.5);
glPopMatrix();
```

Diagram 7 shows the few examples of drawing some 3D Primitives

The hierarchical level are being done also using glPushMatrix and glPopMatrix. There at least 3 level of hierarchical level in this character design. One of the example of our hierarchical level are Body > Left Arm > Left Forearm > Left Palm > Recycle Logo.

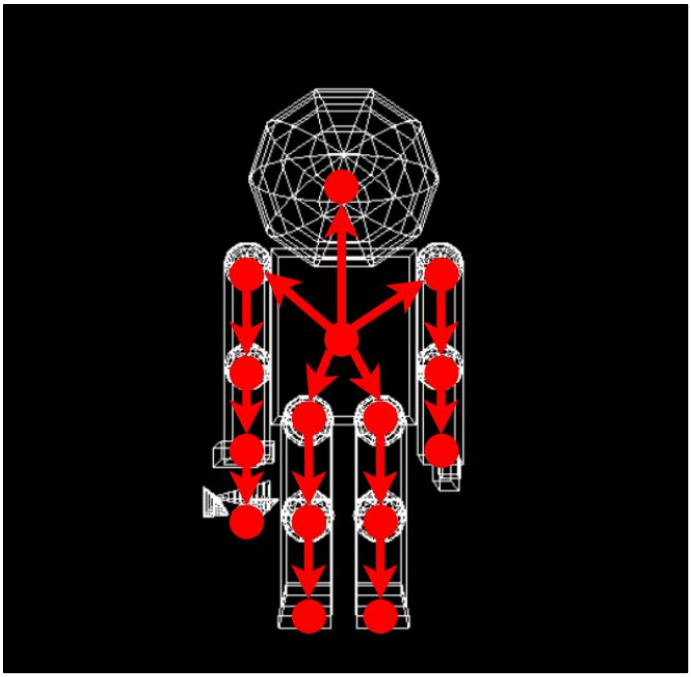

Diagram 8 shows the hierarchical level of our character design

```
    //LEFT HAND
        glPushMatrix(); //1
...
            //left shoulder joint
            glPushMatrix();
                glTranslatef(-0.80, 0.60, 0);
                //glutWireSphere(0.20, 10, 10);
                glutWireSphere(0.20, 10, 10);
            glPopMatrix();

            //left arm
            glPushMatrix();
                glTranslatef(-0.8, 0.15, 0);
                glScalef(0.65, 1.75, 0.8);
                //glutWireCube(0.5);
                glutWireCube(0.5);
            glPopMatrix();

            glPushMatrix();//2

                glTranslatef(-0.8, -0.25, 0);
                glRotatef(rLeftFore, 1, 0, 0);
                glTranslatef(0.8, 0.25, 0);

                //left elbow joint
                glPushMatrix();//3
                    glTranslatef(-0.8, -0.25, 0);
                    //glutWireSphere(0.20, 10, 10);
                    glutWireSphere(0.20, 10, 10);
                glPopMatrix();
```

Diagram 9 shows the coding of making the hierarchical level

## 3D Transformation

The transformation are being used to move the 3D output to appropriate position and size. Our output are made possible by using all three kinds of transformation; glTranslatef, glRotatef, and glScalef. Translation can be done by using glTranslatef, this will help to move the object to the desired position; Rotation can be done by using glRotatef, this function will rotate an object to an angle about any axis (x,y,z); Scaling can be done by using glScalef, this function will enlarge or shrink the object.

In order to make our character to have a pose or movement, a lot of rotation will are implemented. The rotation are a little bit complicated as the rotation function glRotatef will rotate the object

about the origin. Since our 3D object have an hierarchical level, we also need to make sure our transformation code are placed in the correct place. By implementing our transformation code in between a glPushMatrix and glPopMatrix can make sure that we are transforming the correct object without affecting the other.

Rotation are being used in a lot of places for our character posing. We rotated the leg and hand position of our character to make it seems like floating in air. The rotation are done in level, especially the leg. First the leg below the knee were facing front, we rotate it about the y-axis to the facing sideway first then only we rotate it inwards by using the z-axis. Next, rotation about the z-axis on the whole leg are done to make our character spread the leg a little, this will move the whole leg including the part below the knee.

```
//Right leg
    glPushMatrix();

        glTranslatef(0.3, -0.7, 0);
        glRotatef(rRightLeg,0,0,1);
        glTranslatef(-0.3,0.7,0);
```

```
glPushMatrix(); //Knee {

    glTranslatef(0.3, -1.5, 0);

    glRotatef(-30 + rRightKnee, 0, 0, 1);
    glRotatef(90, 0, 1, 0);
    glTranslatef(-0.3, 1.5, 0);
```

Diagram 10 shows the coding transformation of the leg

The diagram above shows the coding rotation transformation of the leg. The transformation of the knee are below the hierarchy level of the whole right leg. Which means that if the code on the left side of the diagram above are implemented, it will affect the object transformation on the right side of diagram. Translation are being done before and after the rotation to make sure our object are in the correct place when they're rotating about the origin.

After using multiple transformation, we finally had made our baby to have a pose that will fits our theme.

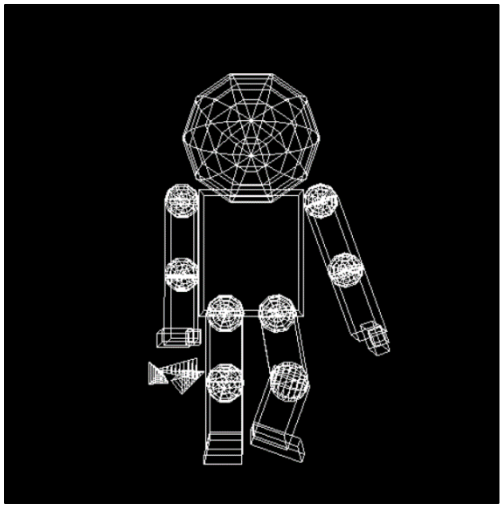

Diagram 11 shows the initial pose of our Character

# Animation / Movement

To make the character comes to life, animation needs to be done. Animation can be implemented easily by adding a glutPostRedisplay() function at the end of the display() function. This will let the OpenGL to run the display function in a loop with a changes in every frame. The apply the movement on our character we will set a variable as a delta distance or rotation. For each loop, certain transformation will change according to the delta value.

Firstly, we make our character to raise his hand first. We initialize the rotation of the arm as 0. Then slowly increase the rotation according to our delta rotation variable 0.1.

```
double rLeftArm = 0;
double rLeftFore = 0;
double drotate = 0.1; //delta rotation
```

Diagram 12 shows the initialization of the variable for rotation animation on Left Hand

In every loop, the rotation will have a changes of 0.1. A condition is being implemented to stop the rotation. In here, we make our character to raise his arm until a 45 degree then it stop, then proceed to move the part below the elbow to another 45 degree then it stop.

```
glTranslatef(-0.8, 0.6, 0);
glRotatef(rLeftArm, 1, 0, 0);
glTranslatef(0.8, -0.6, 0);

if (rLeftArm > -45) {
    rLeftArm -= drotate;
}
else if (rLeftFore > -45) {
    rLeftFore -= drotate;
}
```

Diagram 13 shows the implementation of animation in Left Arm

The same concept are being applied to other parts of the body of our character. For example, the recycle bin logo will rotate and it will always be in a loop. After our character had raised their hand. The recycle bin will act as a magical blast and it will translate in and out.

```
//Recycle triangle
glPushMatrix();
    glTranslatef(posxRecycle, posyRecycle, 0);
    glTranslatef(-0.875, 0, 0);
    glRotatef(rRecycle, 0, 1, 0);
    glTranslatef(0.875, 0, 0);

    rRecycle += spin;
    if (rLeftFore < -45) {
        posyRecycle -= dyRecycle;
        posxRecycle += dxRecycle;
        if (posyRecycle <= -2.5 || posyRecycle > 0) {
            dyRecycle = -dyRecycle;
            dxRecycle = -dxRecycle;
        }
    }
```

Diagram 14 shows the code of movement of Recycle Logo of our character

Other parts of animation such as making the baby translating up and down to make it look floating and making small movement to the leg in a loop to make it look natural are also being applied using the same concept.

```
//Right leg
    glPushMatrix();

        glTranslatef(0.3, -0.7, 0);
        glRotatef(rRightLeg,0,0,1);
        glTranslatef(-0.3,0.7,0);

        rRightLeg += drRightLeg;
        if (rRightLeg >= 15 || rRightLeg <= 10) {
            drRightLeg = -drRightLeg;
        }
```

```
glPushMatrix(); //Knee {

    glTranslatef(0.3, -1.5, 0);

    glRotatef(-30 + rRightKnee, 0, 0, 1);
    glRotatef(90, 0, 1, 0);
    glTranslatef(-0.3, 1.5, 0);

    rRightKnee -= drRightKnee;
    if (rRightKnee >= 20 || rRightKnee < 0) {
        drRightKnee = -drRightKnee;
    }
```

Diagram 15 shows the code of making small motion on the leg.

```
glPushMatrix(); //WHOLE BODY{

glTranslatef(0,posyWhole + dyWhole,0);
posyWhole += dyWhole;
if (posyWhole >= 0.2 || posyWhole <= -0.2) {
    dyWhole = -dyWhole;
}
```

Diagram 16 shows the code of making the character floating up and down

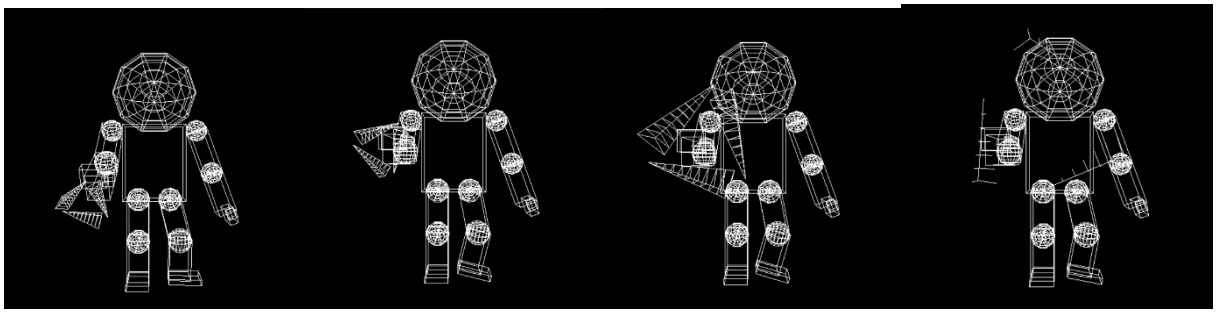

Diagram 17 shows the animation of our character, from raising his hands to shooting the blast and floating in a loop

# Achievement

In the process of doing this assignment, we had learnt a lot of things. First, we had discovered and explored the software Maya. Even though we had used this software before, but we were only scratching it on the surface. For now, we actually went deeper and explore how the modelling can be made. There are so much more stuff that we hadn't explore before regarding the modelling in Maya, such as the mirror duplicate, combining object, ridging the body and etc. We had also learnt how to perform color weight and making animation using Maya.

In OpenGL, we had applied what we had learnt in the class into this assignment. The hierarchical level of this assignment is definitely more advanced than what we had did before. Through this assignment, we had gained more knowledge regarding OpenGL. We had also discovered a new way to implement the animation to the object in OpenGL.

We had also learnt the difference of Maya and OpenGL. We believed that Maya is an enhanced version of OpenGL. OpenGL is still the traditional way of making 3D Modelling as it still needs coding to implement. While for Maya, everything are already built in and we only need to use the installed tools to create our model.

Besides the achievement of technical skill, there are many other things that we had achieved too such as the soft skills. We had gained more communication skills and teamworking skills. Our interpersonal skill had also improved.

# Conclusion

There is a significant difference between Maya and OpenGL. Before we start the assignment, we had made a lot of discussion on what our hero will be in the theme of Malaysia. A lot of changes has been made throughout the process of doing this assignment. Eventually, we finally make our decision and complete the design of our character. We distributed our work with each other to make sure our work can be done smoothly.

In Maya, we tried to model our character as beautiful as possible, since the tools are easier than using openGL. However in openGL, we only managed to get the output shapes of the character. In the final result, we are still happy with our outcome.

Even though we had faced a lot of problems during the process of our assignment, we still managed to complete it thanks to the cooperation of team members, and also with the guidance of our lecturer.

# Acknowledgement

In preparation for this assignment, we have to appreciate a few person who deserved our deepest gratitude. First, we would like to show our gratitude to our dearest lecturer, Madam Norhaida bte Mohd Suaib, for giving us a change to explore the 3D modelling in OpenGL and other software. Through this assignment, we had enhanced our understanding and knowledge in applying multiple primitives to a hierarchical level. Besides, we would also like to thanks to those who helped us directly and indirectly along with this assignment.

Finally, thanks to the teammates for spending so much time and full commitment to this assignment. The teammates give full responsibility and cooperation for everything they do.