VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELGAVI, KARNATAKA -590 018

A Mini-Project Report on

"DOREMON"

Submitted in partial fulfillment for the Computer Graphics Laboratory with Mini-Project (18CSL67) course of Sixth Semester of Bachelor of Engineering in Computer Science & Engineering during the academic year 2021-22.

By

Team Code: **CGP2022-C05**

Dhanush M S	**4MH19CS026**
Kushi P Kogilvadi	**4MH19CS042**

: Under the Guidance of :

Prof. Harish H K

Assistant Professor

Department of CS&E

MIT Mysore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna Taluk, Mandya Dist-571477.

Accredited By:

2021-22

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

~~ CERTIFICATE ~~

*Certified that the mini-project work entitled "**Doremon**" is a bonafide work carried out by **Dhanush M S** (4MH19CS026) & **Kushi P Kogilvadi** (4MH19CS042) for the Computer Graphics Laboratory with Mini-Project (18CSL67) of Sixth Semester in Computer Science & Engineering under Visvesvaraya Technological University, Belgavi during academic year 2021-22.*

It is certified that all corrections/suggestions indicated for Internal Assignment have been incorporated in the report. The report has been approved as it satisfies the course requirements.

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Signature of Guide	Signature of the HOD
Prof. Harish H K	**Dr. Shivamurthy R C**
Assistant Professor	Professor & HOD
Dept. of CS&E	Dept. of CS&E
MIT Mysore	MIT Mysore

External viva

Name of the Examiners **Signature with date**

1)………………………………………………………………………………

2) ………………………………………………………………………………

~~~~ ACKNOWLEDGEMENT ~~~~

It is the time to acknowledge all those who have extended their guidance, inspiration, and their wholehearted co-operation all along our project work.

We are also grateful to **Dr. B G Naresh Kumar**, principal, MIT Mysore, **Dr. Shivamurthy R C**, HOD, CS&E, MIT Mysore, Guide **Harish H K,** Assistant professor, CS&E, MIT Mysore for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

We wish to place a deep sense of gratitude to all Teaching and Non-Teaching staffs of Computer Science and Engineering Department for whole-hearted guidance and constant support without which Endeavour would not have been possible.

Our gratitude will not be complete without thanking our parents and our friends, who have been a constant source of support and aspirations

Kushi P Kogilvadi

Dhanush M S

~~~ ABSTRACT ~~~

In this project the main aim is to illustrate the concepts and usage of pre-built functions in OpenGL. Creating a cartoon character by the name "DOREMON" where the idea behind this project is to display cartoon character doremon with computer graphics. This graphics package is based on the OpenGL library functions. The programming language used here is C using OpenGL libraries. In this project, we are demonstrating a cartoon character DOREMON in 3D

To make it look almost exactly the same as the cartoon took as a reference, we have used shapes like spheres and cubes and we have used basic red, blue, black, white colors

As it is a 3D display of the cartoon character, we will be able to rotate it along the x axis, even we are demonstrating translation, scaling functions, zoom in, zoom out function with keyboard interaction using D and K keys

~~~~~ CONTENTS ~~~~~

CHAPTER - 1

INTRODUCTION

1.1 Aim

The aim of this project is to display a cartoon character (DOREMON) by using the pre-built functionalities in Open GL

1.2 Overview

Doremon is an 3d cartoon character which is simple, good looking and entertaining. We mainly have created the character in this project such as displaying the cartoon character holding a Dora cake, we have made the doremon to rotate around the y axis and we have used translation, scaling functionalities for the cartoon to change its size.

To make it decorative and colorful, keeping the character as reference, we have used basic colors such as blue, white, black, and red where white color is used for the hands, feet, mouth, stomach area of the character. Black for the eye ball color of the character, blue for the arms, legs, face, and the body area. Red color is used to represent the opened mouth of doremon and the nose

We have used camera functionalities to move the character in x, y and z axes where it looks like the character is moving forward and backward

We have also used keyboard functionalities to move the character along the axes and to spin

1.3 Outcome

The outcome of the project is that, we will be able to see a 3D cartoon character which is almost same as the reference taken with the specified functionalities.

CHAPTER – 2

DESIGN AND IMPLEMENTATION

2.1 Algorithm

Step 1: Start

Step 2: Register Displaymode

Step 3: Register WindowSize and WindowPosition

Step 4: Create window providing "Doremon"

Step 5: invoke void Draw function

Step 6: Declare Global variables T,Cx,Cy,Cz,

Step 7: Construct respective spheres cuboids with all properties and their positions as specified with coordinates

Step 8: Invoke Init() to setup our perspective

Step 9: Utilize key() for keyboard interaction

Step 10: Stop

2.2 Flow Chart

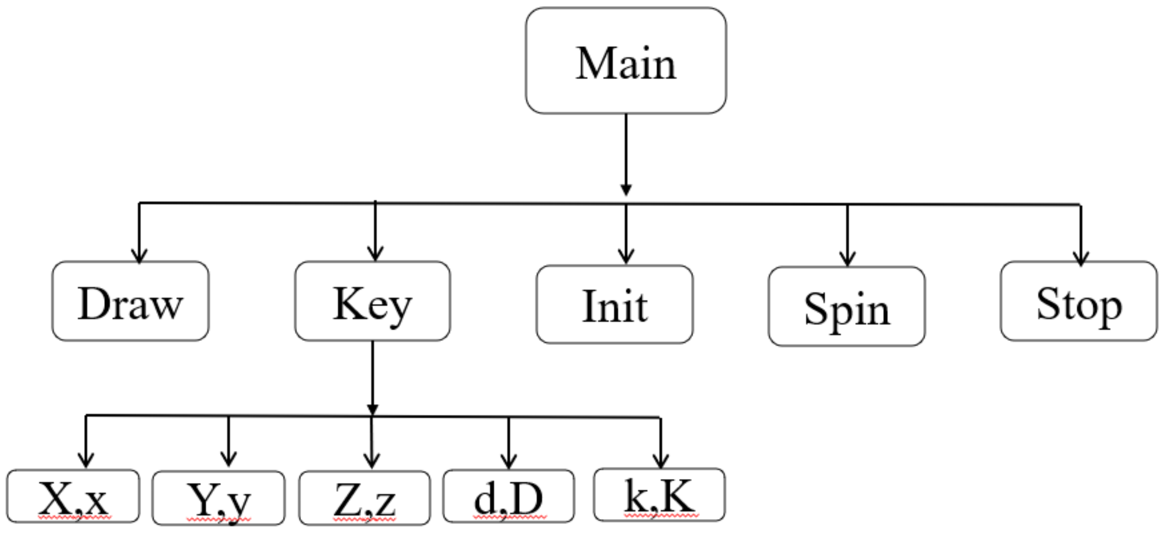


Fig 2.1 Flow chart

2.3 OpenGL API's Used with Description

This program is implemented using various OpenGL functions which are shownbelow.

Various functions used in this program.

➢ glutInit() : interaction between the windowing system and OPENGL is initiated.

➢ glutInitDisplayMode() : used when double buffering is required and depth information is required.

➢ glutCreateWindow() : this opens the OPENGL window and displays the title at topof the window.

➢ glutInitWindowSize() : specifies the size of the window.

➢ glutInitWindowPosition() : specifies the position of the window in screen coordinates.

➢ glutKeyboardFunc() : handles normal ascii symbols.

➢ glutIdleFunc() : this handles the processing of the background.

➢ glutDisplayFunc() : this handles redrawing of the window.

➢ glutMainLoop() : this starts the main loop, it never returns.

➢ glRotate() : used to rotate 3D objects

➢ glColor3fv() : used to render color to faces.

➢ glFlush() : used to flush the pipeline.

➢ glutPostRedisplay() : used to trigger an automatic redrawal of the object.

➢ glLoadIdentity() : used to load or initialize to the identity matrix.

➢ glTranslatef() : used to translate or move the rotation centre from one point to another.

➢ glutCreateMenu() creates a new pop-up menu and returns a unique small integer identifier.

➢ gluLookAt() : used to change the camera view point

➢ glutSolidSphere() : used to create a 3D sphere

➢ glTranslatef() : used fot translation

➢ glPushMatrix() : pushes the current matrix stack down by one, duplicating the current matrix.

➢ glPopMatrics() : pops the current matrix stack, replacing the current matrix with the one below it on the stack

➢ glScalef() :

➢ glSolidCube() : used to create a 3D cube

2.4 Source Code

```
#include<iostream>
#include<GL/glut.h>
#include<GL/glu.h>
#include<stdio.h>
#include<stdarg.h>
#include<windows.h>
#include<string.h>
GLfloat T = 0;
GLfloat Cx=0,Cy=0,Cz=3;
void spin()
{
   T = T + 0.1;
   if(T>360)
      T = 0;
   glutPostRedisplay();
}
void stop()
{
   T = T + 1000;
   if(T>360)
      T = 0;
   glutPostRedisplay();
}
void Draw()
 {
   glClearColor(0,1,0,0);
   glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();
  gluLookAt(Cx,Cy,Cz,0,0,0,0,1,0);
   glRotatef(T,0,1,0);
   /*headblue*/
   glPushMatrix();
      glColor3f(0,0.55,0.86);
       glTranslatef(0,0.9,-0.15);
      glutSolidSphere(0.6,10,10);
   glPopMatrix();
   /*head*/
   glPushMatrix();
      glColor3f(1,1,1);
       glTranslatef(0,0.9,0);
      glutSolidSphere(0.5,10,10);
   glPopMatrix();
   /*middle*/
   glPushMatrix();
      glColor3f(0,0.55,0.86);
      glScalef(0.8,1.0,0.8);
      glutSolidSphere(0.7,10,10);
   glPopMatrix();
    /*bell*/
    glPushMatrix();
      glColor3f(1,1,0);
      glTranslatef(0,0.5,0.32);
      glScalef(0.25,0.25,0.25);
      glutSolidSphere(0.4,10,10);
   glPopMatrix();
   /*pocket*/
   glPushMatrix();
```

```
glColor3f(1,1,1);
        glTranslatef(0,-0.2,0.5);
        glScalef(0.5,0.35,0.25);
        glutSolidSphere(0.4,10,10);
    glPopMatrix();
    /*legs*/
    glPushMatrix();
     glColor3f(0,0.55,0.86);
        glTranslatef(-0.35,-0.5,0.1);
        glScalef(0.3,0.6,0.3);
        glutSolidCube(1);
    glPopMatrix();
    /* doremon pada*/
    glPushMatrix();
     glColor3f(1,1,1);
        glTranslatef(-0.35,-0.8,0.25);
        glScalef(0.5,0.2,0.5);
        glutSolidSphere(0.7,10,10);
    glPopMatrix();
    /*legs*/
    glPushMatrix();
     glColor3f(0,0.55,0.86);
        glTranslatef(0.35,-0.5,0.1);
        glScalef(0.3,0.6,0.3);
        glutSolidCube(1);
    glPopMatrix();
    /*doremon pada*/
    glPushMatrix();
     glColor3f(1,1,1);
```

```
glTranslatef(0.35,-0.8,0.25);
        glScalef(0.5,0.2,0.5);
        glutSolidSphere(0.7,10,10);
    glPopMatrix();
    /*hands*/
     glPushMatrix();
     glColor3f(0,0.55,0.86);
        glTranslatef(-0.35,-0.06,0.5);
        glScalef(0.3,0.2,0.3);
        glutSolidCube(1);
    glPopMatrix();
    /*doremon hasta*/
    glPushMatrix();
    glColor3f(1,1,1);
         glTranslatef(-0.37,-0.06,0.825);
         glutSolidSphere(0.17,10,10);
        glPopMatrix();
    glPushMatrix();
     glColor3f(0,0.55,0.86);
        glTranslatef(0.35,-0.06,0.5);
        glScalef(0.3,0.2,0.3);
        glutSolidCube(1);
    glPopMatrix();
    /*doremon hasta*/
    glPushMatrix();
        glColor3f(1,1,1);
         glTranslatef(0.37,-0.06,0.825);
         glutSolidSphere(0.17,10,10);
        glPopMatrix();
       /*doracake*/
        glPushMatrix();
        glColor3f(0.8,0.75,0);
        glTranslatef(0.37,0.1,0.825);
         glutSolidSphere(0.17,10,10);
        glPopMatrix();
        glPushMatrix();
        glColor3f(0.8,0.75,0);
        glTranslatef(0.37,0.2,0.825);
         glutSolidSphere(0.17,10,10);
        glPopMatrix();
        glPushMatrix();
        glColor3f(1,1,1);
        glTranslatef(0.37,0.15,0.825);
         glutSolidSphere(0.17,10,10);
        glPopMatrix();
    /*tail*/
    glPushMatrix();
     glColor3f(1,1,1);
        glTranslatef(0,-0.3,-0.6);
        glScalef(0.4,0.4,0.4);
        glutSolidSphere(0.5,10,10);
    glPopMatrix();
    /*eyes*/
    /*first inner*/
    glPushMatrix();
         glColor3f(1,1,1);
        glTranslatef(-0.39,0.9,0.3);
        glScalef(0.2,0.2,0.2);
```

```
glutSolidSphere(0.35,10,10);
 glPopMatrix();
 /*first outer*/
  glPushMatrix();
     glColor3f(0,0,0);
     glTranslatef(-0.3,0.9,0.25);
     glScalef(0.2,0.2,0.2);
     glutSolidSphere(0.8,10,10);
 glPopMatrix();
 /*second inner*/
 glPushMatrix();
      glColor3f(1,1,1);
     glTranslatef(0.39,0.9,0.3);
     glScalef(0.2,0.2,0.2);
     glutSolidSphere(0.35,10,10);
 glPopMatrix();
 /*second outer*/
  glPushMatrix();
     glColor3f(0,0,0);
     glTranslatef(0.3,0.9,0.25);
     glScalef(0.2,0.2,0.2);
     glutSolidSphere(0.8,10,10);
 glPopMatrix();
 glPopMatrix();
 /*nose*/
 glPushMatrix();
     glColor3f(1,0,0);
     glTranslatef(0,0.82,0.43);
     glScalef(0.4,0.2,0.4);
```

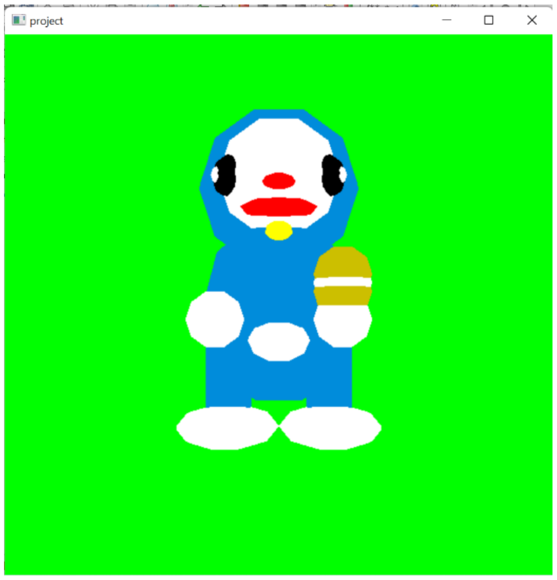
```
   glutSolidSphere(0.3,10,10);
    glPopMatrix();
    /*mouth*/
     glPushMatrix();
        glTranslatef(0,0.68,0.3);
        glScalef(0.6,0.2,0.4);
        glutSolidSphere(0.45,10,10);
    glPopMatrix();
    glutSwapBuffers();
}
void Key(unsigned char ch,int x,int y)
{
    switch(ch)
    {
        case 'x' : Cx = Cx - 0.5;   break;
        case 'X' : Cx = Cx + 0.5;   break;
        case 'y' : Cy = Cy - 0.5;   break;
        case 'Y' : Cy = Cy + 0.5;   break;
        case 'z' : Cz = Cz - 0.5;   break;
        case 'Z' : Cz = Cz + 0.5;   break;
        case 'd' : glutIdleFunc(spin); break;
        case 'D' : glutIdleFunc(spin); break;
        case 'k' : glutIdleFunc(stop); break;
        case 'K' : glutIdleFunc(stop); break;
    }
}
void MyInit()
{
    glEnable(GL_DEPTH_TEST);
```

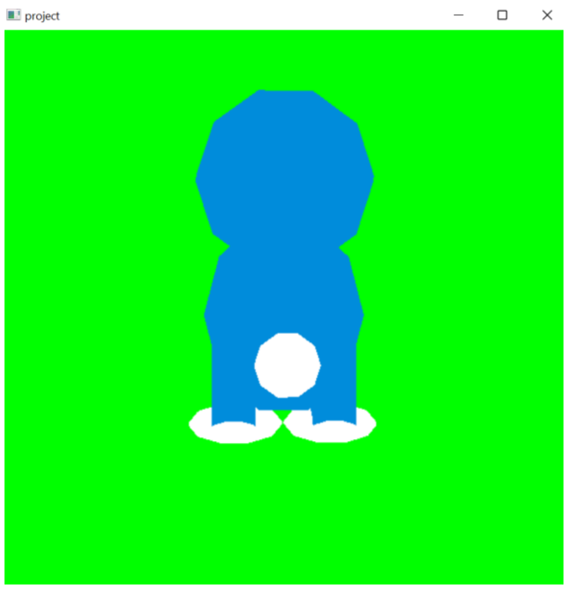
```
  glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glFrustum(-1,1,-1,1,2,10);
   glMatrixMode(GL_MODELVIEW);
}
int main(int argC ,char *argV[])
{
   glutInit(&argC,argV);
   glutInitWindowSize(600,600);
   glutInitWindowPosition(50,50);
   glutInitDisplayMode(GLUT_RGB|GLUT_DOUBLE|GLUT_DEPTH);
   glutCreateWindow("project");
   MyInit();
   glutDisplayFunc(Draw);
   glutIdleFunc(spin);
   glutKeyboardFunc(Key);
   glutMainLoop();
   return 0;
}
```

CHAPTER – 3

RESULT ANALYSIS

3.1 Snap shots





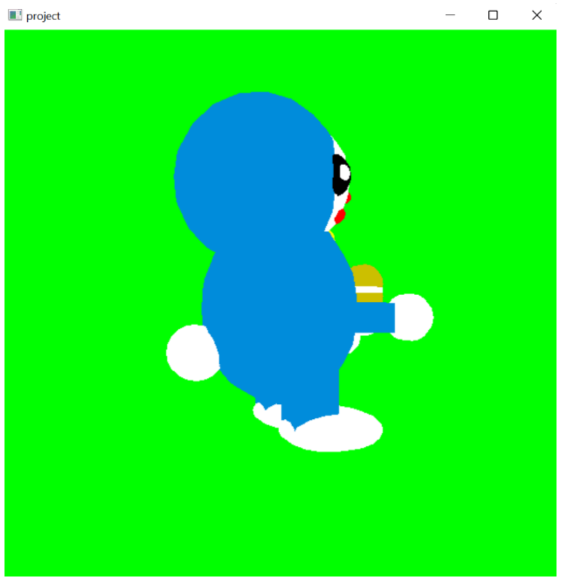


Fig 3.1 Front, back, and side view of 3D DOREMON

3.2 Discussion

We have successfully incorporated the Glut APIs to create spheres, cubes and then using scaling, rotation, translation functionalities we have aligned them in manner such that we were able to get our desired output. Further using the glcolor3fv() function we have added colors to our 3d model. To rotate our model we used rotate functions and also added camera functionalities using which user can zoom in or zoom out in x, y or z axes respectively, user can make use of keys x, y, z to use camera functionalities as seen in the above snap shots (Fig 3.1.1).

CHAPTER – 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

Our aim has been successfully achieved which was to design a 3D cartoon character with rotating, transition, and scaling functionalities with user interface through keyboard

4.2 Future Enhancement

Along with the present functionalities we would like to make the model smoother to view, add a helicopter upon doremon's head, more transitions with respect to the mouth.

CHAPTER – 5

REFERENCES

- Text Book- Interactive Computer Graphics Pearson Education 5th edition : Edward angel
- OpenGL Documentation
- You Tube channel learning bits