DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER GRAPHICS AND MULTIMEDIA

A MINI PROJECT REPORT

On

“Paint Software using OpenGL”

Submitted by

Name	USN
Adithya S N	4NM15CS008
Ankith B	4NM15CS020

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified)

NITTE -574 110, Udupi District, KARNATAKA

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved,
NBA Accredited, ISO 9001:2008 Certified)
NITTE -574 110, Udupi District, KARNATAKA

CERTIFICATE

Certified that the project work entitled " Paint Softwre using OpenGL " is a bonafide work carried out by Adithya S N **(USN)**4NM15CS008**,Anktih B (USN)**4NM15CS020 in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belgaum during the year 2017-2018.

Signature of Guide
Pawan Hegde
Assistant Professor
Dept of CSE
NMAMIT Nitte

Abstract:

The “Paint Software using OpenGL” is an OpenGL project which aims to simulate the working of a paint brush to draw something on a screen. The drawing can be done with various colors. The brush size can be increased or decreased depending on the users needs. The brush also comes in the shape of square, triangle and line. These can be done by interaction through keyboard or mouse.

Introduction:

The “Paint Software using OpenGL” is an OpenGL project which aims to simulate the working of a paint brush to draw something on a screen. The drawing can be done with various colors. The brush size can be increased or decreased depending on the users needs. The brush also comes in the shape of square, triangle and line.

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.

The “Paint Software using OpenGL” is an OpenGL project which aims to simulate the working of a brush. The drawing with brush can be done with various colors. The brush size can be increased or decreased depending on the users needs. The brush also comes in the shape of square, triangle and line.

The “Paint Software using OpenGL” also comes with menu buttons to select colors using mouse to draw.Menu buttons also has an option to increase or decrease the size of the brush and also has the eraser option.

Implementation:

i. Special functions:

a) int SubMenu=glutCreateMenu(menu);

Syntax:

return_type specifier variable=glutCreateMenu(functon_name);

uses:

creates a id with a return type specifier to implement sub menus under main menu.

b) glutPostRedisplay()

marks the current window as needing to be distroyed

c) push_back()

pushes the values to vectors which in turn is used to represent the drawn object in screen.

d) glutMotionFunc()

sets the motion callback for the current window.
Used to store the mouse co-ordinates,color and size of the brush using push_back() function to a vector.

ii. Source Code:

```
#include<iostream>
#include<GL/gl.h>
#include<GL/glu.h>
#include<GL/glx.h>
#include <GL/glut.h>
#include<vector>
#define WINDOW_WIDTH 800
#define WINDOW_HEIGHT 600

using namespace std;

struct Color{ float r,g,b;};

struct Point{
   int mouseX, mouseY, brush;
   Color color;
   int size;
};
vector<Point> points;
Color color={1,1,1};
int size=1;
int num = 0;
bool leftClick = false;
void drawShape(int num)
{
   for (auto &x:points)
   {
      glColor3f(x.color.r, x.color.g, x.color.b);
      switch(x.brush){
      case 0:
         glBegin(GL_POLYGON);
         glVertex2f(x.mouseX - x.size, x.mouseY - x.size);
         glVertex2f(x.mouseX + x.size, x.mouseY - x.size);
```

```
            glVertex2f(x.mouseX + x.size, x.mouseY + x.size);
            glVertex2f(x.mouseX - x.size, x.mouseY + x.size);
            glEnd();
            break;
        case 1:
            glBegin(GL_TRIANGLES);
            glVertex2f(x.mouseX - x.size, x.mouseY - x.size);
            glVertex2f(x.mouseX + x.size, x.mouseY - x.size);
            glVertex2f(x.mouseX, x.mouseY + x.size);
            glEnd();
            break;
        case 2:
            glBegin(GL_LINES);
            glVertex2f(x.mouseX, x.mouseY - x.size);
            glVertex2f(x.mouseX, x.mouseY + x.size);
            glEnd();

        }
    }
    glFlush();
}

void display(void)
{
    glClearColor(0, 0, 0, 0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    drawShape(num);
    glFlush();
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
    case '1':
        color={1,1,1};
        break;
    case '2':
```

```
    color={1,0,0};
    break;
case '3':
    color={0,1,0};
    break;
case '4':
    color={1, 1, 0};
    break;
case '5':
    color={1, 0, 1};
    break;
case '6':
    color={0, 1, 1};
    break;
case '7':
    color={0, 0, 1};
    break;
case '+':
    if (size < 65)
        size = size * 2;
    break;
case '-':
    if (size > 1 )
        size = size / 2;
    break;
case 's':
    num++;
    if (num > 2)
        num = 0;
    break;
case 'c':
      points.resize(0);
     break;
case 'e':
    color={0,0,0};
    break;

}
```

```
    glutPostRedisplay();
}
void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, WINDOW_WIDTH - 1, WINDOW_HEIGHT - 1, 0, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
}
void mouse(int button, int action, int x, int y)
{
    if (button == GLUT_LEFT_BUTTON)
    {
        if (action == GLUT_DOWN)
            leftClick = true;
        else
            leftClick = false;
    }
}


void mouseMove(int x, int y)
{
    if (leftClick)
    {
        points.push_back({x,y,num,color, size});
        glutPostRedisplay();
    }
}
void menufun1(int value)
{
    if(value==1)
    {
        //display();
        glutDisplayFunc(display);
        glutMotionFunc(mouseMove);
        glutMouseFunc(mouse);
        glutKeyboardFunc(keyboard);
    }
    if(value==9)
    {
        if (size < 65)
            size = size * 2;
        glutPostRedisplay();
    }
    if(value==10)
    {
        if (size > 1 )
            size = size / 2;
        glutPostRedisplay();
    }
    if(value==14)
    {
        color={0,0,0};
        glutPostRedisplay();
    }
}

void menufun2(int value)
{
    if(value==2)
    {
        color={1,1,1};
        glutPostRedisplay();
    }
    if(value==3)
    {
        color={1,0,0};
        glutPostRedisplay();

    }
    if(value==4)
    {
        color={0,1,0};
        glutPostRedisplay();
    }
    if(value==5)
    {
        color={1, 1, 0};
        glutPostRedisplay();
    }
    if(value==6)
    {
        color={0, 0, 1};
        glutPostRedisplay();
    }
    if(value==7)
    {
        color={1,0,1};
        glutPostRedisplay();
    }
    if(value==8)
    {
        color={0, 1, 1};
        glutPostRedisplay();
    }

}
void menufun3(int value)
{
    if(value==11)
    {
        num=0;
        glutDisplayFunc(display);
        glutMotionFunc(mouseMove);
        glutMouseFunc(mouse);
        glutKeyboardFunc(keyboard);
    }
    if(value==12)
    {
        num=1;
        glutDisplayFunc(display);
        glutMotionFunc(mouseMove);
        glutMouseFunc(mouse);
        glutKeyboardFunc(keyboard);
    }
    if(value==13)
    {
        num=2;
        glutDisplayFunc(display);
        glutMotionFunc(mouseMove);
        glutMouseFunc(mouse);
        glutKeyboardFunc(keyboard);
    }

}
int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT);
    glutInitWindowPosition(500, 300);
    glutCreateWindow("Paint Software");
    int SubMenu=glutCreateMenu(menufun2);
    glutAddMenuEntry("White",2);
    glutAddMenuEntry("Red",3);
    glutAddMenuEntry("Green",4);
    glutAddMenuEntry("Yellow",5);
    glutAddMenuEntry("Blue",6);
    glutAddMenuEntry("Magenta",7);
    glutAddMenuEntry("Cyan",8);
    int SubMenu2=glutCreateMenu(menufun3);
    glutAddMenuEntry("Quad",11);
    glutAddMenuEntry("Triangle",12);
    glutAddMenuEntry("Line",13);
    int MainMenu=glutCreateMenu(menufun1);
    glutAddSubMenu("Draw",SubMenu2);
    glutAddMenuEntry("+",9);
    glutAddMenuEntry("-",10);
    glutAddMenuEntry("Eraser",14);
    glutAddSubMenu("Color",SubMenu);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

Explanation:

- Vectors are used to store mouse co-ordinates.
- drawShape funtion is used to draw using various brush shapes like square,triangle or line.
- display function is used to draw the object which in turn calls drawShape function.
- keyboard function is used to change the color of brush, increase or decrease brush size,change the size of brush etc.
- init function is used to initialize the drawing window.
- mouse function is used to check if left mouse button is pressed or not.
- mouseMove function is used to store mouse co-ordinates as mouse moves and also call glutPostRedisplay.
- menfun1 function is used to select draw option,increase or decrease brush size and eraser function in menu.
- menufun2 function is used select colors from menu.
- menufun3 function is used select brush shape such as sqaure,triangle or line.
- main function is used to call various other function, initialise drawing window,create menus and submenus etc.

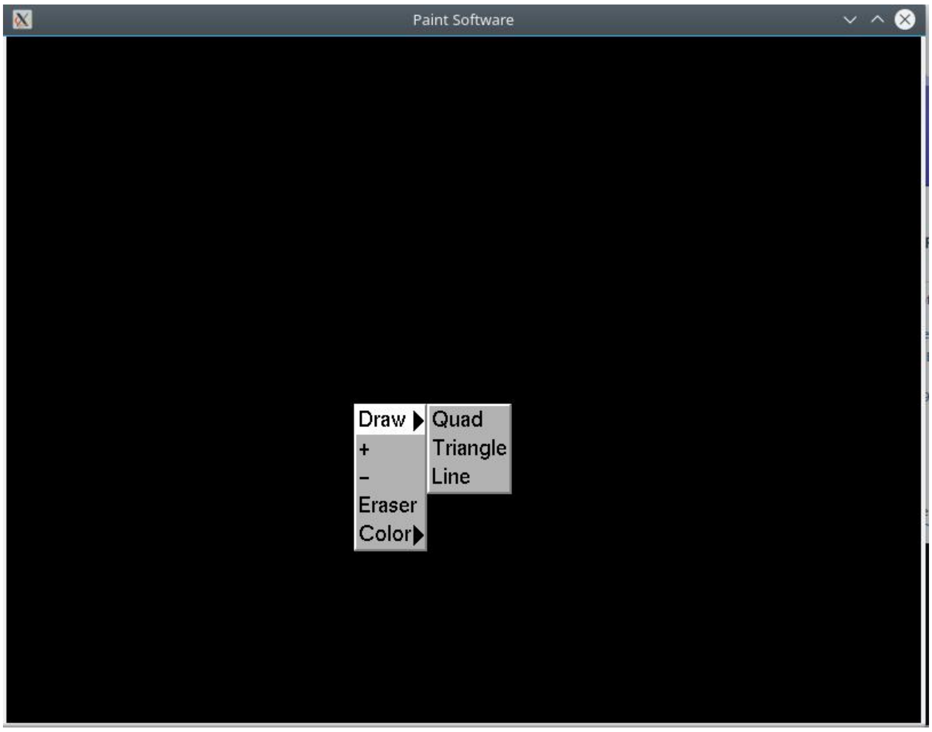

Illustration 1: Menu functions

Illustration 2: Menu functions

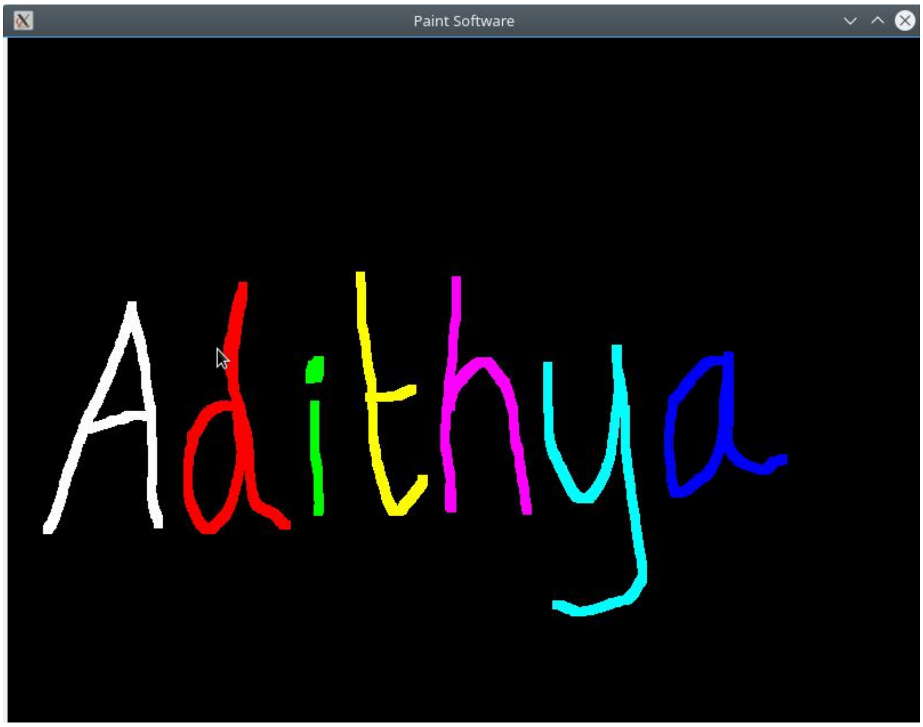

Illustration 3: Drawing with different colors

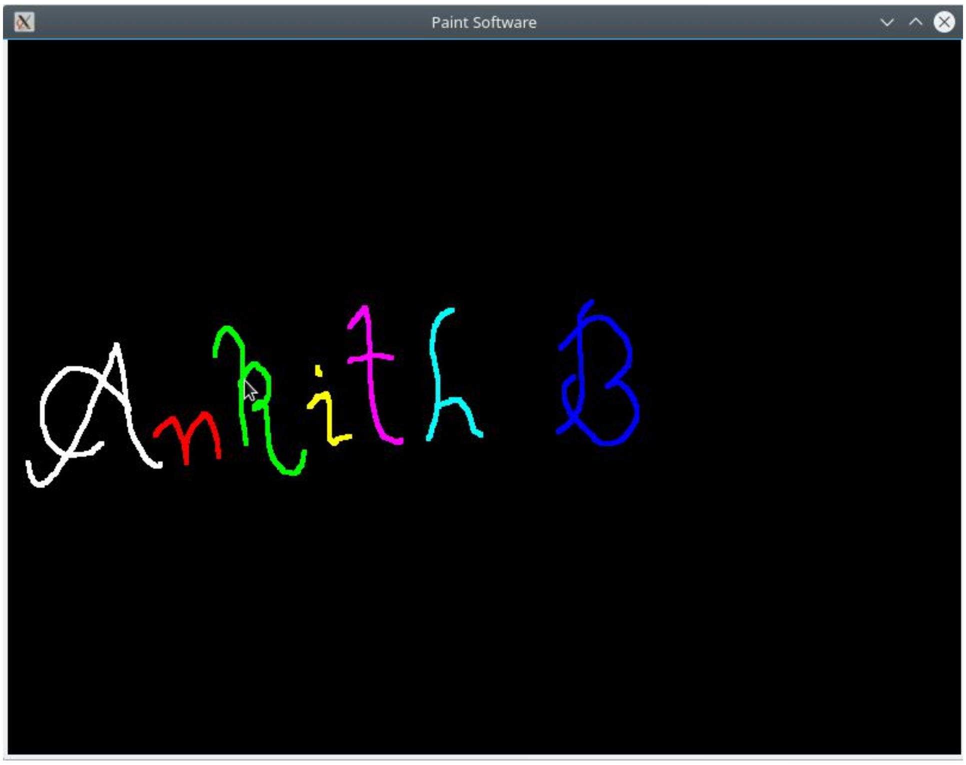

Illustration 4: Drawing with different colors

Conclusion:

The very purpose of developing this project is to exploit the srength of OpenGL graphics capabilities and to implement various concepts . The OpenGL project library consists of various functions to implement keyboard and mouse functions to take inputs, to create and interact using menus etc. These functions are used regularly to create any projects using OpenGL as well as to provide an understanding of the use of OpenGL library in real world usage.

Reference:

https://www.opengl.org/

Amardhruva N Prabhu