**National Institute of Technology Silchar**

**Department of Computer Science and Engineering**

**Computer Graphics Project**

**(B. Tech 4th Sem)**

**under “Mr. Badal Soni”**

**Title: Clock in OpenGL**

**Submitted by:**

**Mrinmay Mukherjee (15-1-5-057)**

**Rohan Deb (15-1-5-046)**

**Ashwin Bansal (15-1-5-006)**

**Rishab Pincha (15-1-5-050)**

Acknowledgement

It would not have been possible for us to complete the project without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them.

Firstly we are highly indebted to Mr. Badal Soni for giving us this opportunity to work on the project. The experience of doing this project has surely enriched our technical skills to a great extent. Also we would like to extend our sincere gratitude to him for his guidance and constant supervision, as well as for providing necessary information regarding the project.

We also take this opportunity to thank all our friends who have directly or indirectly helped us in completing the project by offering us wonderful insights or solutions to the various problems that we encountered throughout the course of the project.

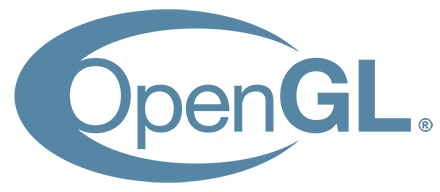
Project Overview

The objective of the following project is to display a working clock consisting of second, minute and hour hands along with date using OpenGL Library.

The main outlines of the project are as follows:

* To draw the basic structure of the clock, 60 points have been plotted along a circular path at intervals of pi/30 degree using the OpenGL function glBegin(GL\_POINTS).
* To draw the second, minute and hour hands three lines have been drawn of appropriate length and at appropriate positions using the openGL function glBegin(GL\_LINES). It treats each pair of consecutive vertices as an independent line segment. And draw the lines depending upon the coordinates provided.
* To make the clock dynamic, we refresh the screen and redraw the clock at a specific time interval (15 ms) using the OpenGL function glutTimerFunc() by recalculating the angles for each hand.
* Also we maintain the aspect ratio of the clock by using the OpenGL function glutRedrawFunc().
* To display the date we use the functions glRasterPos2f() to specify its position and glutBitmapCharacter() to display characters as Bitmap.

Overview of OpenGL



OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms.

OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

OpenGL Available Everywhere: Supported on all UNIX® workstations, and shipped standard with every Windows 95/98/2000/NT and MacOS PC, no other graphics API operates on a wider range of hardware platforms and software environments.

OpenGL runs on every major operating system including Mac OS, OS/2, UNIX, Windows 95/98, Windows 2000, Windows NT, Linux, OPENStep, and BeOS; it also works with every major windowing system, including Win32, MacOS, Presentation Manager, and X-Window System. OpenGL is callable from Ada, C, C++, Fortran, Python, Perl and Java and offers complete independence from network protocols and topologies.

The OpenGL interface: Our application will be designed to access OpenGL directly through functions in three libraries namely: gl,glu,glut.

Description of Functions

**drawPoint**

The function drawpoint takes a structure variable and prints the point at coordinates (x,y).

**drawLine**

The function drawLine takes three arguments x, y and angle. It draws a line from (x,x) to (y\*cos(angle), y\*sin(angle)).

**addDate**

This function uses the predefined time structure present in C under the header files time.h and sys/timeb.h to produce the present date and time. The date is then displayed at the top of the clock by using the function glutBitmapCharacter().

**drawMarks**

The function drawMarks does not take any argument. It draws 60 points along a circular path starting at 0 degree to 2 pi degree at intervals of pi/30 degrees. The points at 0 degree, pi/2 degree, pi degree and 3pi/2 degree are drawn at a fixed radius (CLOCK\_RADIUS) from the centre of the clock in the corresponding axes. All other points are plotted as (CLOCK\_RADIUS\*sin(angle), CLOCK\_RADIUS\*cos(angle)).

**drawClock**

The function drawClock does not take any arguments and draws the second, minute and hour hands of their corresponding length and from centre of the clock to positions decided upon by the secondAngle, minuteAngle and hourAngle respectively.

**reshape**

The function reshape takes the width and height of the display window as parameter and rescales the points and lines to maintain the aspect ratio in case the screen is resized.

**redraw**

The function redraw does not take any argument and redraws the lines for second, minute and hour hand at specified time intervals.

Source Code

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEADERS

#include <stdio.h>

#include <math.h>

#include <GL/glut.h>

#include <GL/gl.h>

#include <time.h>

#include <sys/timeb.h>

#include <string.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//CONSTANTS

#define HOUR\_HAND\_LENGTH 50.0

#define MINUTE\_HAND\_LENGTH 70.7106781187

#define SECOND\_HAND\_LENGTH 78.1024967591

#define CLOCK\_RADIUS 80.0

#define PI 3.14159

#define CLOCK\_VOL 100.00

#define ANGLE\_ONE\_MIN PI/30.0

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//STRUCTURE DEFINING POINT

typedef struct Point {

double x , y;

}point;

//ANGLES FOR EACH HAND

double secondAngle = 0 , minuteAngle = 0 , hourAngle = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//FUNCTION TO INITIALIZE STUFF

void init(void) {

gluOrtho2D(-100 , 100 , -100 , 100);

glClearColor(0.0 , 0.0 , 0.0 , 0.0);

glMatrixMode(GL\_PROJECTION);

glShadeModel (GL\_SMOOTH);

glLoadIdentity();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//FUNCTION TO DRAW A POINT

void drawPoint(point p) {

glBegin(GL\_POINTS);

glVertex2f(p.x , p.y);

glEnd();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//FUNCTION TO DRAW A LINE

void drawLine(double x , double y , double angle) {

glVertex2f(x , x);

glVertex2f(y\*cos(angle) , y\*sin(angle) );

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void addDate() {

time\_t rawtime;

struct tm \* timeinfo;

time ( &rawtime );

timeinfo = localtime ( &rawtime );

char \* s = asctime (timeinfo);

char y[20];

int count\_space = 0 , i;

for(i=0 ; i<strlen(s) ; ++i) {

if(s[i]==' ') {

count\_space++;

y[i] = ',';

}

else{

y[i] = s[i];

}

if(count\_space==3)

break;

}

y[i] = '\0';

glColor3f(1.0 , 1.0 , 0.0);

glRasterPos2f(0 , 90);

for(i=0 ; y[i]!='\0' ; ++i) {

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_10 , y[i]);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//FUNCTION TO DRAW THE BASIC CLOCK STRUCTURE

void drawMarks(void) {

point sp , ep , p;

int count = 0;

double i = PI/6.0;

sp.x = 0 , sp.y = 0;

ep.x = 0 , ep.y = CLOCK\_RADIUS;

glPointSize(10.0);

drawPoint(sp);

for(i=0 ; i<=2\*PI ; i+=PI/30.0) {

if(i==PI/2.0) {

p.x = CLOCK\_RADIUS;

}

else if(i==3\*PI/2.0) {

p.x = -CLOCK\_RADIUS;

}

else {

p.x = ep.y\*sin(i);

}

p.y = ep.y\*cos(i);

if(count%5==0) {

glPointSize(7.0);

drawPoint(p);

}

else {

glPointSize(3.0);

drawPoint(p);

}

count++;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//MAIN DRAW FUNCTION

void drawClock(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0 , 0.0 , 0.0);

glLineWidth(1.0);

drawMarks();

addDate();

glLineWidth(5.0);

glColor3f(1.0f, 0.0f, 1.0f);

glBegin(GL\_LINES);

drawLine(0.0 , HOUR\_HAND\_LENGTH , -hourAngle+PI/2);

glEnd();

glLineWidth(3.0);

glColor3f(0.0f, 1.0f, 1.0f);

glBegin(GL\_LINES);

drawLine(0.0 , MINUTE\_HAND\_LENGTH , -minuteAngle+PI/2);

glEnd();

glLineWidth(1.0);

glColor3f(0.0f, 0.0f, 1.0f);

glBegin(GL\_LINES);

drawLine(0.0 , SECOND\_HAND\_LENGTH , -secondAngle+PI/2);

glEnd();

glFlush();

glutSwapBuffers();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//FUNCTION TO MAINTAIN ASPECT RATIO WHEN SCREEN IS RESIZED

void reshape(int w , int h) {

double aspectRatio;

//to avoid division by 0

if(h==0) {

h=1 ;

}

glViewport(0 , 0 , (GLsizei)w , (GLsizei)h );

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

aspectRatio = (double)w/(double)h;

if (w <= h)

glOrtho (-CLOCK\_VOL , CLOCK\_VOL , -CLOCK\_VOL / aspectRatio , CLOCK\_VOL / aspectRatio , 1.0, -1.0);

else

glOrtho (-CLOCK\_VOL \* aspectRatio , CLOCK\_VOL \* aspectRatio, -CLOCK\_VOL, CLOCK\_VOL , 1.0, -1.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//TIMER FUNCTION

void redraw() {

struct timeb tb;

time\_t tim=time(0);

struct tm\* t;

t=localtime(&tim);

ftime(&tb);

secondAngle = (double)(t->tm\_sec+ (double)tb.millitm/1000.0)/30.0 \* PI;

minuteAngle = (double)(t->tm\_min)/30.0 \* PI + secondAngle/60.0;

hourAngle = (double)(t->tm\_hour > 12 ? t->tm\_hour-12 : t->tm\_hour)/6.0 \* PI+ minuteAngle/12.0;

glutPostRedisplay();

glutTimerFunc(15, redraw , 1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//DRIVER FUNCTION

int main(int argc , char \*\* argv) {

glutInit(&argc , argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500 , 500);

glutInitWindowPosition(100 , 100);

glutCreateWindow("Clock");

init();

glutDisplayFunc(drawClock);

glutReshapeFunc(reshape);

glutTimerFunc(15 , redraw , 1);

glutMainLoop();

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/