**PROGRAM:**

#14. PRATHMESH BHOSALE

#include <math.h>

#include <GL/glut.h>

using namespace std;

float angle = 135;

float inc = 1.0;

void drawCircle(float segments, float radius, float sx, float sy) {

glBegin(GL\_LINE\_LOOP);

for(int ii = 0; ii < segments; ii++) {

float theta = 2.0f \* 3.1415926f \* float(ii) / float(segments);//get the current angle

float x = radius \* cosf(theta);//calculate the x component

float y = radius \* sinf(theta);//calculate the y component

glVertex2f(x + sx, y + sy);// vertex of circles

}

glEnd();

}

void drawCircles(float x1,float y1,float angle) {

float segments = 25;

float radius = 3.0;

// Draw the original circle

glLineWidth(4);

glColor3f(0,1,1);

drawCircle(segments,radius,x1,y1);

glColor3f(1,0,1);

glLineWidth(3);

glBegin(GL\_LINES); //hour line

glVertex2f(x1,y1);

glVertex2f(1,1);

glEnd();

glColor3f(1,1,0);

glLineWidth(1); //time line

glBegin(GL\_LINES);

glVertex2f(x1,y1);

glVertex2f(x1,2);

glEnd();

double radian = angle\*3.14/180;

float y2 = 10 \* cos(radian);

float x2 = 10 \* sin(radian);

radius = 1.0;

glColor3f(1,1,1);

drawCircle(segments,radius,x2,y2);

glBegin(GL\_LINES); // vertex of lines

glVertex2f(x1,-2.8);

glVertex2f(x2,y2);

glEnd();

}

void display(void) {

glClearColor (0.0,0.0,0.0,1.0);

glClear (GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

glTranslatef(-10,0,-30);

glColor3f(1,1,1);

if (angle>225) {

angle = 225;

inc = -inc;

}

if (angle<135) {

angle = 135;

inc = -inc;

}

angle += inc;

drawCircles(0,0,angle);

glutSwapBuffers();

}

void reshape (int w, int h) {

glMatrixMode (GL\_PROJECTION); //maps camera to screen

glLoadIdentity ();

gluPerspective (100, (GLfloat)w / (GLfloat)h, 0.5, 100.0);

glMatrixMode (GL\_MODELVIEW); // maps objects to local coordinates

}

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

glutInit (&argc, argv);

glutInitDisplayMode (GLUT\_DOUBLE); //double buffered bit-mask

glutInitWindowSize (800, 600);

glutInitWindowPosition (0, 0);

glutCreateWindow ("A Pendulum");

glutDisplayFunc (display);

glutIdleFunc (display);

glutReshapeFunc (reshape);

glutMainLoop ();

return 0;

}

**OUTPUT:**

Graphical user interface, application

Description automatically generated