**Practical 8**

Implement animation principles for any object.

**Program Code:-**

**Bouncing ball code:-**

#include <GL/glut.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include <math.h>

#include <stdio.h>

#include <string.h>

#define WIDTH 640

#define HEIGHT 480

void reshape(int width, int height){

glViewport(0,0,width,height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-WIDTH/2,WIDTH/2-1,-HEIGHT/2,HEIGHT/2-1,-1,1);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void init(void){

glClearColor(0.0,0.0,0.0,1.0);

glPointSize(2.0);

}

void Timer(int ex)

{

glutPostRedisplay();

glutTimerFunc(30,Timer,0);

}

int k=20;

void circle(int p,int m)

{

int x=0,y,d,r;

r=m;

y=r;

d=3-2\*r;

while(x<=y)

{

glVertex2i(x,y+p);

glVertex2i(y,x+p);

glVertex2i(-x,y+p);

glVertex2i(-y,x+p);

glVertex2i(-x,-y+p);

glVertex2i(-y,-x+p);

glVertex2i(y,-x+p);

glVertex2i(x,-y+p);

if(d<0)

d=d+4\*x+6;

else

{

d=d+4\*(x-y)+10;

y--;

}

x++;

}

}

int r=100,flag=0;

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glColor4f(0.0,1.0,1.0,1.0);

glBegin(GL\_POINTS);

circle(k,r);

if(flag==0)

{

if((k+r)<=240)

{

k=k+10;

}

if((k+r)>=240){

flag=1;

}

}

if(flag==1)

{

k=k-10;

if((k-r)<=-240)

{

flag=0;

if(r!=10)

r=r-10;

}

}

glEnd();

glutSwapBuffers();

}

void idle(void){

/\* do nothing \*/

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA | GLUT\_DEPTH);

glutInitWindowPosition(0,0);

glutInitWindowSize(WIDTH,HEIGHT);

glutCreateWindow(argv[0]);

init();

glutIdleFunc(idle);

glutReshapeFunc(reshape);

glutDisplayFunc(display);

glutTimerFunc(0,Timer,0);

glutMainLoop();

return(1);

}