EXPERIMENT 6 **DATE : 21/10/2021**

AIM

To write a program to animate a swinging pendulum.

ALGORITHM

Step 1: Start

Step 2: Initialize the glut library toolkit

Step 3: Initialize window size and position

Step 4: Set pixel(0, 0)

Step 5:

```
points = []
theta = -3.14
r = 15.0
while theta <= 0:
    x = r * math.cos(theta)
    y = r * math.sin(theta)
    Append (x, y) to points
    theta += 0.1

For two iterations:
    swing(points)
    swing(reversed(points))
```

Step 6: Define function swing

```
Function swing(points):
  For each point in points:
    Set pixel(point)
    Wait for 0.1 seconds
    Unset pixel(point)
```

Step 7: Stop

PROGRAM

```
import time
import math
from OpenGL.GL import *
from OpenGL.GLU import *
```

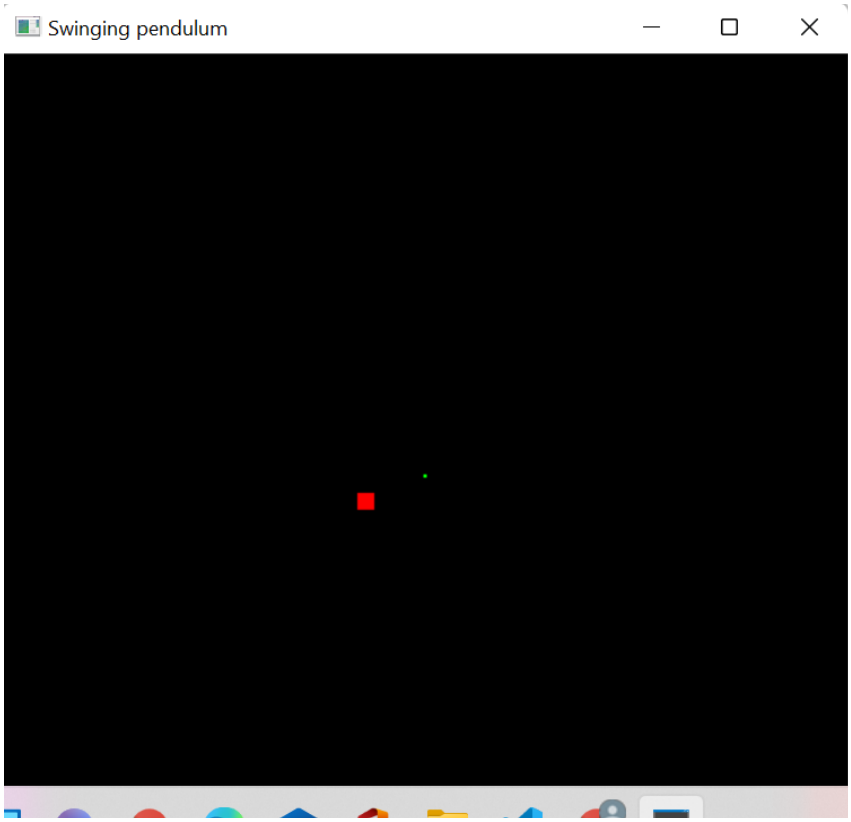
```
from OpenGL.GLUT import *

WINDOW_SIZE = 500
DEFAULT_SCALE = 100

def pendulum():
    glClear(GL_COLOR_BUFFER_BIT)
    glPointSize(2)

    glBegin(GL_POINTS)
    glColor3f(0, 1, 0)
    glVertex2f(0, 0)
    glEnd()
    glFlush()

    glPointSize(10)
    points = []
    theta = -3.14
    r = 15.0
    while theta <= 0:
        x = float(r) * math.cos(theta)
        y = float(r) * math.sin(theta)
        points.append((x / DEFAULT_SCALE, y / DEFAULT_SCALE))
        theta += 0.1

    for i in range(2):
        swing(points)
        swing(reversed(points))



def swing(points):
    for point in points:
        glPointSize(10)
```

```
        glColor3f(1, 0, 0)
        glBegin(GL_POINTS)
        glVertex2f(point[0], point[1])
        glEnd()
        glFlush()

        time.sleep(0.1)
        glBegin(GL_POINTS)
        glColor3f(0, 0, 0)
        glVertex2f(point[0], point[1])
        glEnd()
        glFlush()


def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(WINDOW_SIZE, WINDOW_SIZE)
    glutInitWindowPosition(450, 200)

    glutCreateWindow("Swinging pendulum")
    glutDisplayFunc(pendulum)
    glutMainLoop()


main()
```

RESULT

The required program has been created.

INPUT/OUTPUT


Swinging pendulum