

## **Project Code:**

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

import time

from inputs import *

from drawHere import drawHere

def draw_point(x, y):
    glPointSize(5)
    glEnable(GL_POINT_SMOOTH)
    glBegin(GL_POINTS)
    glVertex2f(x,y)
    glEnd()

def iterate():
    glViewport(0, 0, 1265, 645) # keep it same as window size
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(0.0, 1000, 0.0, 500, 0.0, 1.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

def getZone(x1, y1, x2, y2):
    dx=x2-x1
    dy=y2-y1
    if abs(dx)>=abs(dy) and dx>=0 and dy>=0: return 0
    if abs(dy)>=abs(dx) and dx>=0 and dy>=0: return 1
    if abs(dy)>=abs(dx) and dx<=0 and dy>=0: return 2
```

```
if abs(dx)>=abs(dy) and dx<=0 and dy>=0: return 3
if abs(dx)>=abs(dy) and dx<=0 and dy<=0: return 4
if abs(dy)>=abs(dx) and dx<=0 and dy<=0: return 5
if abs(dy)>=abs(dx) and dx>=0 and dy<=0: return 6
if abs(dx)>=abs(dy) and dx>=0 and dy<=0: return 7
```

```
def changingZone0toX(zone, x, y):
```

```
    if zone==0:
        x_=x
        y_=y
    if zone==1:
        x_=y
        y_=x
    if zone==2:
        x_=-y
        y_=x
    if zone==3:
        x_=-x
        y_=y
    if zone==4:
        x_=-x
        y_=-y
    if zone==5:
        x_=-y
        y_=-x
    if zone==6:
        x_=y
        y_=-x
    if zone==7:
        x_=x
        y_=-y
    return (x_, y_)
```

```
def changingZoneXto0(zone, x, y):
```

```
    if zone==0:
        x_=x
        y_=y
```

```

if zone==1:
    x_=y
    y_=x
if zone==2:
    x_=y
    y_=-x
if zone==3:
    x_=-x
    y_=y
if zone==4:
    x_=-x
    y_=-y
if zone==5:
    x_=-y
    y_=-x
if zone==6:
    x_=-y
    y_=x
if zone==7:
    x_=x
    y_=-y
return (x_, y_)

```

```

def getIntermediatePoints(x1, y1, x2, y2):
    dx=x2-x1
    dy=y2-y1
    d=2*dy-dx
    incE=2*dy
    incNE=2*(dy-dx)
    y=y1
    x=x1
    intPoints=[]
    while x<=x2:
        intPoints.append( (x, y) )

        if d>0:
            d=d+incNE

```

```

        y=y+1
    else:
        d=d+incE

    x=x+1

    return intPoints

def newDrawLine(x1, y1, x2, y2):
    zone=getZone(x1, y1, x2, y2)

    p1=changingZoneXto0(zone, x1, y1)
    x1_=p1[0]
    y1_=p1[1]
    p2=changingZoneXto0(zone, x2, y2)
    x2_=p2[0]
    y2_=p2[1]

    intPoints=getIntermediatePoints(x1_, y1_, x2_, y2_)

    for i in range(0, len(intPoints)):
        x=intPoints[i][0]
        y=intPoints[i][1]

        p=changingZone0toX(zone, x, y)
        x=p[0]
        y=p[1]

        draw_point(x, y)

def drawCircle(xc, yc, r):
    d = 1-r
    x = 0
    y = r

    while x<y:
        if d<0: #E

```

```
d = d+2*x+3
x = x+1
```

```
else: #SE
    d = d+2*x-2*y+5
    x = x+1
    y = y-1
```

```
zoneChange(x, y, xc, yc)
```

```
def zoneChange(x, y, xc, yc):
    new_x = x + xc
    new_y = y + yc
    draw_point(new_x, new_y)

    new_x = y + xc
    new_y = x + yc
    draw_point(new_x, new_y)

    new_x = -1 * y + xc
    new_y = x + yc
    draw_point(new_x, new_y)

    new_x = -1 * x + xc
    new_y = y + yc
    draw_point(new_x, new_y)

    new_x = -1 * x + xc
    new_y = -1 * y + yc
    draw_point(new_x, new_y)

    new_x = -1 * y + xc
    new_y = -1 * x + yc
    draw_point(new_x, new_y)

    new_x = y + xc
    new_y = -1 * x + yc
```

```

draw_point(new_x, new_y)

new_x = x+ xc
new_y = -1 * y + yc
draw_point(new_x, new_y)

def showScreen():
    glClearColor(1.0, 1.0, 1.0, 0.0)
    if weather=='night': glClearColor(0.0, 0.0, 0.0, 0.0)
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    drawHere(glColor3f, newDrawLine, drawCircle, time,
    glutSwapBuffers, glClear, GL_COLOR_BUFFER_BIT,
    GL_DEPTH_BUFFER_BIT)
    glutSwapBuffers()
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

glutInit()
glutInitDisplayMode(GLUT_RGBA)
glutInitWindowSize(1265, 645) #window size
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"CSE423 Project") #window name
glutDisplayFunc(showScreen)
glutMainLoop()

from projectRunner import *

#Enter distance to travel by cycle as an integer value
#[Note: if distance is more than 515 units, then hits tree]
distanceTravelled=200

#Enter a weather or mode: sunny, cloudy, rainy or night
weather='sunny'

import numpy as np
import math

```

```
import random
```

```
from inputs import *
```

```
def drawHere(glColor3f, newDrawLine, drawCircle, time,  
glutSwapBuffers, glClear, GL_COLOR_BUFFER_BIT,  
GL_DEPTH_BUFFER_BIT):
```

```
    for i in range(0, distanceTravelled):
```

```
        drawTheHorizon(newDrawLine, drawCircle)
```

```
        drawTheSun(newDrawLine, drawCircle)
```

```
        drawTheMoon(newDrawLine, drawCircle)
```

```
        drawTheCycle(newDrawLine, drawCircle, i)
```

```
        drawTheMan(newDrawLine, drawCircle, i)
```

```
        drawTheCloud(newDrawLine, drawCircle)
```

```
        drawTheTree(newDrawLine, drawCircle)
```

```
    if i==515:
```

```
        #ROTATING TREE
```

```
        #here 'mx' and 'my' variables are used to fix tree position
```

```
        mx=665
```

```
        my=50
```

```
        for j in range(0, 90+1):
```

```
            a = math.cos(math.radians(-j))
```

```
            b = math.sin(math.radians(-j))
```

```
            #distance of axis of rotation along x axis
```

```

r = np.array([[a, -b, mx],
              #distance of axis of rotation along y axis
              [b, a, my],
              [0, 0, 1]])

def putHere(x1, y1, x2, y2):
    #setting rotating tree color
    newDrawLine(
        np.matmul(r, np.array([[x1], [y1], [1]]))
    )[0][0],
        np.matmul(r, np.array([[x1], [y1], [1]]))
    )[1][0],
        np.matmul(r, np.array([[x2], [y2], [1]]))
    )[0][0],
        np.matmul(r, np.array([[x2], [y2], [1]]))
    )[1][0]
    )

```

```

glColor3f(165/255, 42/255, 42/255)
putHere(650-mx, 50-my, 650-mx, 200-my)
putHere(680-mx, 50-my, 680-mx, 200-my)
putHere(650-mx, 50-my, 680-mx, 50-my)
putHere(650-mx, 200-my, 630-mx, 250-my)
putHere(680-mx, 200-my, 700-mx, 250-my)
putHere(680-mx, 200-my, 700-mx, 250-my)
putHere(660-mx, 220-my, 650-mx, 250-my)
putHere(660-mx, 220-my, 680-mx, 250-my)

```

```

glColor3f(0.3,1,0)
putHere(650-mx, 180-my, 620-mx, 180-my)
putHere(620-mx, 180-my, 600-mx, 200-my)
putHere(600-mx, 200-my, 600-mx, 220-my)
putHere(600-mx, 220-my, 580-mx, 230-my)
putHere(580-mx, 230-my, 585-mx, 250-my)
putHere(585-mx, 250-my, 590-mx, 250-my)
putHere(590-mx, 250-my, 595-mx, 270-my)
putHere(595-mx, 270-my, 605-mx, 270-my)

```



```
putHere(605-mx, 270-my, 610-mx, 300-my)
putHere(610-mx, 300-my, 620-mx, 310-my)
putHere(620-mx, 310-my, 640-mx, 310-my)
putHere(640-mx, 310-my, 660-mx, 330-my)
putHere(660-mx, 330-my, 680-mx, 330-my)
putHere(680-mx, 330-my, 700-mx, 310-my)
putHere(700-mx, 310-my, 720-mx, 310-my)
putHere(720-mx, 310-my, 740-mx, 290-my)
putHere(740-mx, 290-my, 750-mx, 270-my)
putHere(750-mx, 270-my, 760-mx, 250-my)
putHere(760-mx, 250-my, 760-mx, 230-my)
putHere(760-mx, 230-my, 740-mx, 210-my)
putHere(740-mx, 210-my, 740-mx, 200-my)
putHere(740-mx, 200-my, 710-mx, 180-my)
putHere(710-mx, 180-my, 680-mx, 180-my)
```

```
drawTheHorizon(newDrawLine, drawCircle)
```

```
drawTheSun(newDrawLine, drawCircle)
```

```
drawTheMoon(newDrawLine, drawCircle)
```

```
drawTheCycle(newDrawLine, drawCircle, i)
```

```
drawTheMan(newDrawLine, drawCircle, i)
```

```
drawTheCloud(newDrawLine, drawCircle)
```

```
# time.sleep(0.01)
glutSwapBuffers()
glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT)
```

```
break
```

```
# time.sleep(0.01)
glutSwapBuffers()
```

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
```

```
def drawTheSun(newDrawLine, drawCircle):  
    if weather!='sunny': return  
  
    #SUN  
    glColor3f(1,0.9,0)  
    sunX=150  
    for sunFiller in range(20):  
        drawCircle(80+sunX, 400, sunFiller)  
        newDrawLine(105+sunX, 400,130+sunX,400)  
        newDrawLine(30+sunX, 400, 55+sunX, 400)  
        newDrawLine(80+sunX, 425,80+sunX,445)  
        newDrawLine(80+sunX, 375,80+sunX,355)  
        newDrawLine(100+sunX, 420,120+sunX,430)  
        newDrawLine(60+sunX, 420,40+sunX,430)  
        newDrawLine(60+sunX, 380,40+sunX,365)  
        newDrawLine(100+sunX, 380,120+sunX,365)
```

```
def drawTheHorizon(newDrawLine, drawCircle):  
    #HORIZON  
    glColor3f(0.6,0.6,0.6)  
    newDrawLine(0, 200, 1000, 200)
```

```
def drawTheMan(newDrawLine, drawCircle, i):  
    #MAN  
    glColor3f(0,0,0.5)  
    drawCircle(50+i, 160, 10)  
    newDrawLine(46+i, 150, 46+i, 145)  
    newDrawLine(54+i, 150, 54+i, 145)  
    newDrawLine(42+i, 145, 58+i, 145)  
    newDrawLine(42+i, 145, 38+i, 113)  
    newDrawLine(58+i, 145, 54+i, 113)  
    newDrawLine(38+i, 113, 54+i, 113)  
    newDrawLine(38+i, 113, 55+i, 90)  
    newDrawLine(55+i, 113, 50+i, 70)
```

```
newDrawLine(50+i, 113,60+i,90)
newDrawLine(60+i, 90,55+i, 70)
newDrawLine(50+i, 70,55+i, 70)
newDrawLine(54+i, 135,105+i, 130)
newDrawLine(56+i, 125,105+i, 130)
```

```
def drawTheCycle(newDrawLine, drawCircle, i):
    #CYCLE
    glColor3f(1,0,0)
    drawCircle(20+i, 70, 20)
    newDrawLine(20+i,70,50+i,100)
    newDrawLine(20+i,70,50+i,70)
    newDrawLine(50+i,100,100+i,110)
    newDrawLine(50+i,70,100+i,110)
    drawCircle(50+i, 70, 4)
    newDrawLine(50+i,70,50+i,100)
    newDrawLine(50+i,100,45+i,110)
    newDrawLine(40+i,110,55+i,110)
    newDrawLine(40+i,110,40+i,113)
    newDrawLine(55+i,110,55+i,113)
    newDrawLine(40+i,113,55+i,113)
    drawCircle(115+i, 70, 20)
    newDrawLine(115+i,70,100+i,115)
    newDrawLine(100+i,115,100+i,130)
    newDrawLine(95+i,130,105+i,130)
```

```
def drawTheTree(newDrawLine, drawCircle):
    #TREE
    glColor3f(165/255, 42/255, 42/255)
    mx=0
    my=0
    newDrawLine(650-mx, 50-my, 650-mx, 200-my)
    newDrawLine(680-mx, 50-my, 680-mx, 200-my)
    newDrawLine(650-mx, 50-my, 680-mx, 50-my)
    newDrawLine(650-mx, 200-my, 630-mx, 250-my)
    newDrawLine(680-mx, 200-my, 700-mx, 250-my)
    newDrawLine(680-mx, 200-my, 700-mx, 250-my)
```

```
newDrawLine(660-mx, 220-my, 650-mx, 250-my)
newDrawLine(660-mx, 220-my, 680-mx, 250-my)
```

```
glColor3f(0.3,1,0)
```

```
newDrawLine(650-mx, 180-my, 620-mx, 180-my)
newDrawLine(620-mx, 180-my, 600-mx, 200-my)
newDrawLine(600-mx, 200-my, 600-mx, 220-my)
newDrawLine(600-mx, 220-my, 580-mx, 230-my)
newDrawLine(580-mx, 230-my, 585-mx, 250-my)
newDrawLine(585-mx, 250-my, 590-mx, 250-my)
newDrawLine(590-mx, 250-my, 595-mx, 270-my)
newDrawLine(595-mx, 270-my, 605-mx, 270-my)
newDrawLine(605-mx, 270-my, 610-mx, 300-my)
newDrawLine(610-mx, 300-my, 620-mx, 310-my)
newDrawLine(620-mx, 310-my, 640-mx, 310-my)
newDrawLine(640-mx, 310-my, 660-mx, 330-my)
newDrawLine(660-mx, 330-my, 680-mx, 330-my)
newDrawLine(680-mx, 330-my, 700-mx, 310-my)
newDrawLine(700-mx, 310-my, 720-mx, 310-my)
newDrawLine(720-mx, 310-my, 740-mx, 290-my)
newDrawLine(740-mx, 290-my, 750-mx, 270-my)
newDrawLine(750-mx, 270-my, 760-mx, 250-my)
newDrawLine(760-mx, 250-my, 760-mx, 230-my)
newDrawLine(760-mx, 230-my, 740-mx, 210-my)
newDrawLine(740-mx, 210-my, 740-mx, 200-my)
newDrawLine(740-mx, 200-my, 710-mx, 180-my)
newDrawLine(710-mx, 180-my, 680-mx, 180-my)
```

```
def drawTheCloud(newDrawLine, drawCircle):
```

```
    if weather!='cloudy' and weather!='rainy': return
```

```
    #CLOUDS
```

```
    glColor3f(0.2,0.8,1)
```

```
    def drawCloudPoints1(x1, y1, x2, y2):
```

```
        newDrawLine(x1*0.9, y1*0.5+290, x2*0.9, y2*0.5+290)
```

```
    def drawCloudPoints2(x1, y1, x2, y2):
```

```
newDrawLine(x1*0.9+640, y1*0.5+290, x2*0.9+640,  
y2*0.5+290)
```

```
#CLOUD-1
```

```
drawCloudPoints1(100, 200, 350, 200)  
drawCloudPoints1(350, 200, 366.60269, 216.82032)  
drawCloudPoints1(366.60269, 216.82032, 368.52207, 235.08997)  
drawCloudPoints1(368.52207, 235.08997, 350, 250)  
drawCloudPoints1(350, 250, 330.13436, 252.52919)  
drawCloudPoints1(330.13436, 252.52919, 309.02111, 255.85095)  
drawCloudPoints1(309.02111, 255.85095, 316.69866, 269.96841)  
drawCloudPoints1(316.69866, 269.96841, 313.81958, 289.06851)  
drawCloudPoints1(313.81958, 289.06851, 300, 300)  
drawCloudPoints1(300, 300, 275.43186, 306.50772)  
drawCloudPoints1(250, 300, 275.43186, 306.50772)  
drawCloudPoints1(250, 300, 262.95585, 324.77738)  
drawCloudPoints1(262.95585, 324.77738, 266.79463, 348.86011)  
drawCloudPoints1(266.79463, 348.86011, 265.83493, 369.62108)  
drawCloudPoints1(265.83493, 369.62108, 256.238, 385.39942)  
drawCloudPoints1(256.238, 385.39942, 235.12476, 400.34733)  
drawCloudPoints1(235.12476, 400.34733, 208.25336, 407.82128)  
drawCloudPoints1(185.22073, 401.17776, 208.25336, 407.82128)  
drawCloudPoints1(166.02687, 384.56899, 185.22073, 401.17776)  
drawCloudPoints1(166.02687, 384.56899, 157.38964, 364.63845)  
drawCloudPoints1(157.38964, 364.63845, 157.38964, 364.63845)  
drawCloudPoints1(157.38964, 364.63845, 150.67179, 338.89484)  
drawCloudPoints1(150.67179, 338.89484, 147.79271, 318.13387)  
drawCloudPoints1(147.79271, 318.13387, 150, 300)  
drawCloudPoints1(150, 300, 163.14779, 277.44236)  
drawCloudPoints1(163.14779, 277.44236, 147.79271, 284.91631)  
drawCloudPoints1(147.79271, 284.91631, 131.47793, 294.05114)  
drawCloudPoints1(105.56622, 293.2207, 131.47793, 294.05114,)  
drawCloudPoints1(85.41267, 285.74675, 105.56622, 293.2207)  
drawCloudPoints1(66.21881, 270.79885, 85.41267, 285.74675)  
drawCloudPoints1(53.7428, 260.83358, 66.21881, 270.79885)  
drawCloudPoints1(53.7428, 260.83358, 46.06526, 240.90305)  
drawCloudPoints1(46.06526, 240.90305, 50.86372, 225.12471)
```

drawCloudPoints1(50.86372, 225.12471,67.1785, 211.00725)  
drawCloudPoints1(67.1785, 211.00725,100, 200)

## #CLOUD-2

drawCloudPoints2(100, 200 ,350, 200)  
drawCloudPoints2(350, 200, 366.60269, 216.82032)  
drawCloudPoints2(366.60269, 216.82032, 368.52207, 235.08997)  
drawCloudPoints2(368.52207, 235.08997, 350, 250)  
drawCloudPoints2(350, 250, 330.13436, 252.52919)  
drawCloudPoints2(330.13436, 252.52919, 309.02111, 255.85095)  
drawCloudPoints2(309.02111, 255.85095,316.69866, 269.96841)  
drawCloudPoints2(316.69866, 269.96841,313.81958, 289.06851)  
drawCloudPoints2(313.81958, 289.06851,300, 300)  
drawCloudPoints2(300, 300,275.43186, 306.50772)  
drawCloudPoints2(250, 300,275.43186, 306.50772)  
drawCloudPoints2(250, 300,262.95585, 324.77738)  
drawCloudPoints2(262.95585, 324.77738,266.79463, 348.86011)  
drawCloudPoints2(266.79463, 348.86011,265.83493, 369.62108)  
drawCloudPoints2(265.83493, 369.62108,256.238, 385.39942)  
drawCloudPoints2(256.238, 385.39942,235.12476, 400.34733)  
drawCloudPoints2(235.12476, 400.34733,208.25336, 407.82128)  
drawCloudPoints2(185.22073, 401.17776,208.25336, 407.82128)  
drawCloudPoints2(166.02687, 384.56899,185.22073, 401.17776)  
drawCloudPoints2(166.02687, 384.56899,157.38964, 364.63845)  
drawCloudPoints2(157.38964, 364.63845,157.38964, 364.63845)  
drawCloudPoints2(157.38964, 364.63845,150.67179, 338.89484)  
drawCloudPoints2(150.67179, 338.89484,147.79271, 318.13387)  
drawCloudPoints2(147.79271, 318.13387,150, 300)  
drawCloudPoints2(150, 300,163.14779, 277.44236)  
drawCloudPoints2(163.14779, 277.44236,147.79271, 284.91631)  
drawCloudPoints2(147.79271, 284.91631,131.47793, 294.05114)  
drawCloudPoints2(105.56622, 293.2207,131.47793, 294.05114,)  
drawCloudPoints2(85.41267, 285.74675,105.56622, 293.2207)  
drawCloudPoints2(66.21881, 270.79885,85.41267, 285.74675)  
drawCloudPoints2(53.7428, 260.83358,66.21881, 270.79885)  
drawCloudPoints2(53.7428, 260.83358,46.06526, 240.90305)  
drawCloudPoints2(46.06526, 240.90305,50.86372, 225.12471)

```

drawCloudPoints2(50.86372, 225.12471,67.1785, 211.00725)
drawCloudPoints2(67.1785, 211.00725,100, 200)

if weather=='rainy':
    for iii in range(0, 30):
        x=random.randint(0, 1265)
        y=random.randint(0, 645)
        glColor3f(0,0,1)
        newDrawLine(x,y,x,y+10)

def drawTheMoon(newDrawLine, drawCircle):
    if weather!='night': return

    #MOON
    glColor3f(1,1,1)
    moonX=150
    for moonFiller in range(30):
        drawCircle(80+moonX, 400, moonFiller)

    glColor3f(0,0,0)
    moonX=165
    moonY=10
    for moonFiller in range(30):
        drawCircle(80+moonX, 400+moonY, moonFiller)

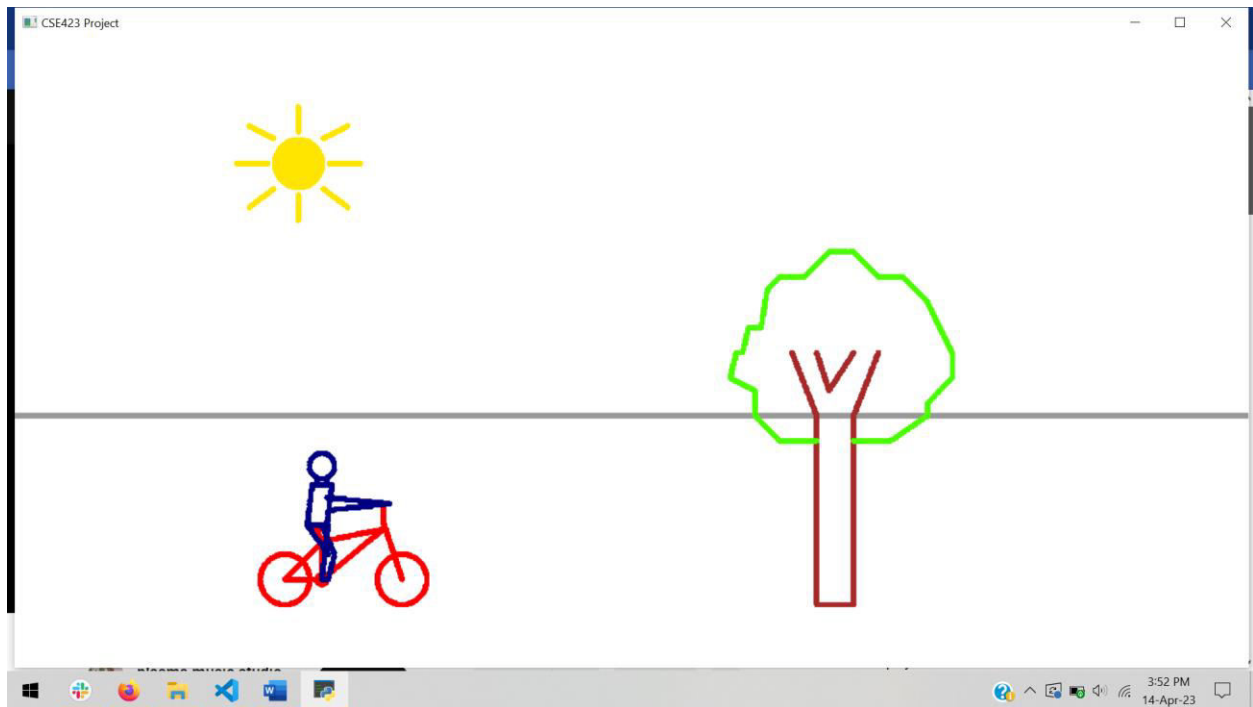
```

### **Outputs:**

Input:

distanceTravelled: 200

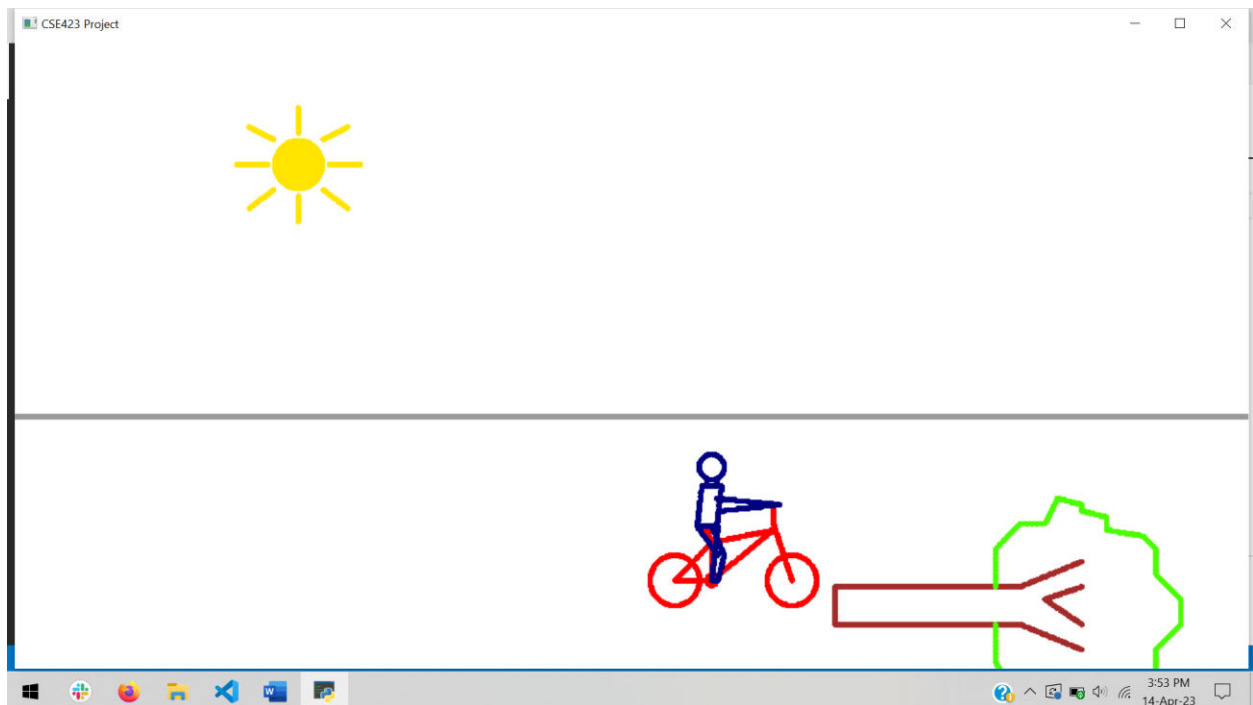
weather: 'sunny'



Input:

distanceTravelled: 600

weather: 'sunny'

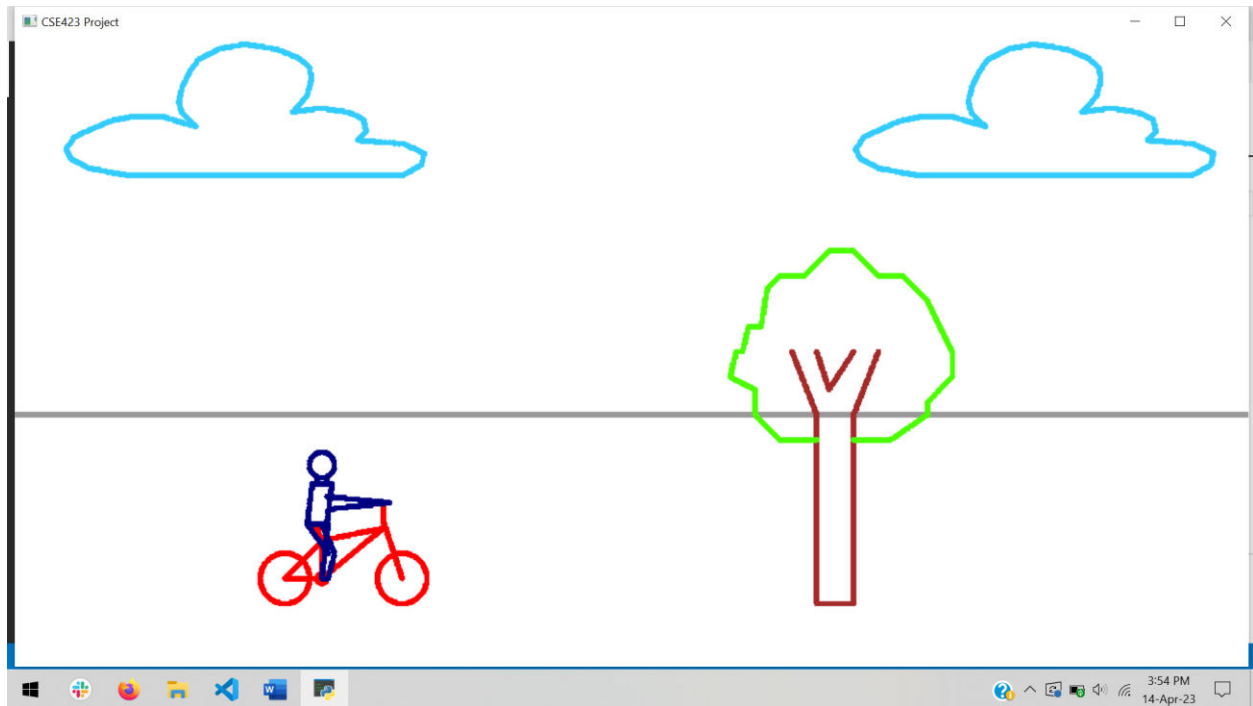




Input:

distanceTravelled: 200

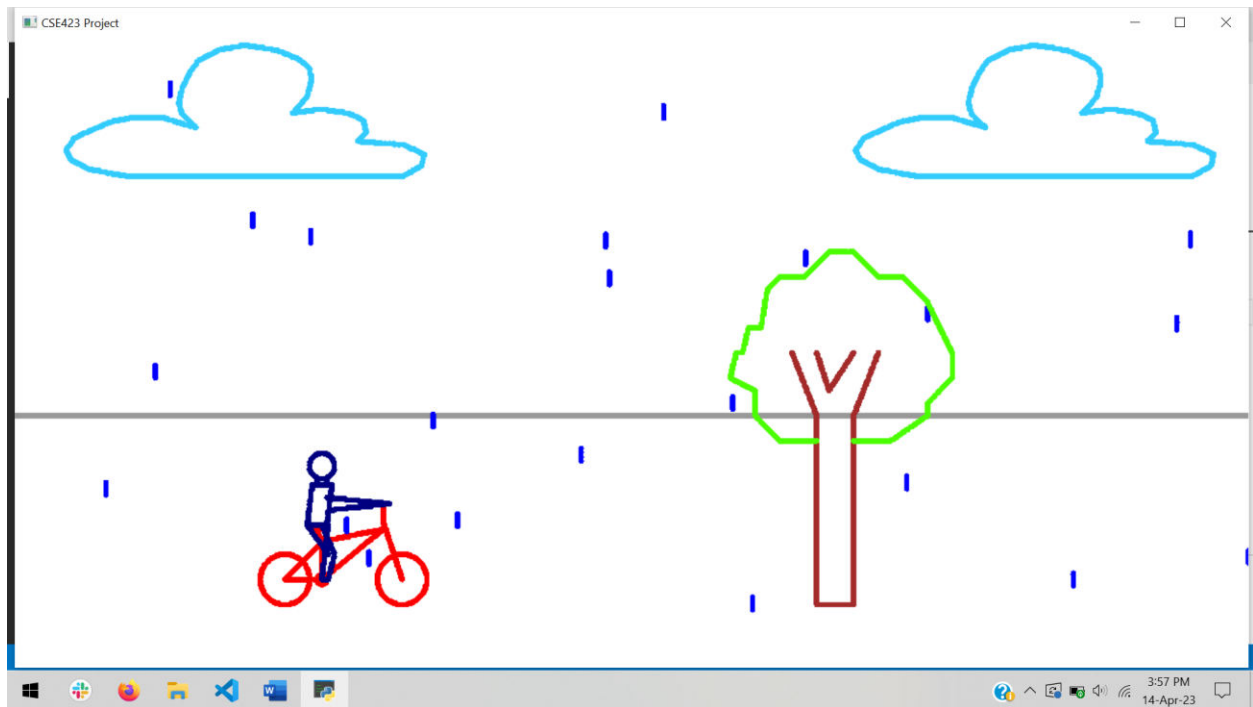
weather: 'cloudy'



Input:

distanceTravelled: 200

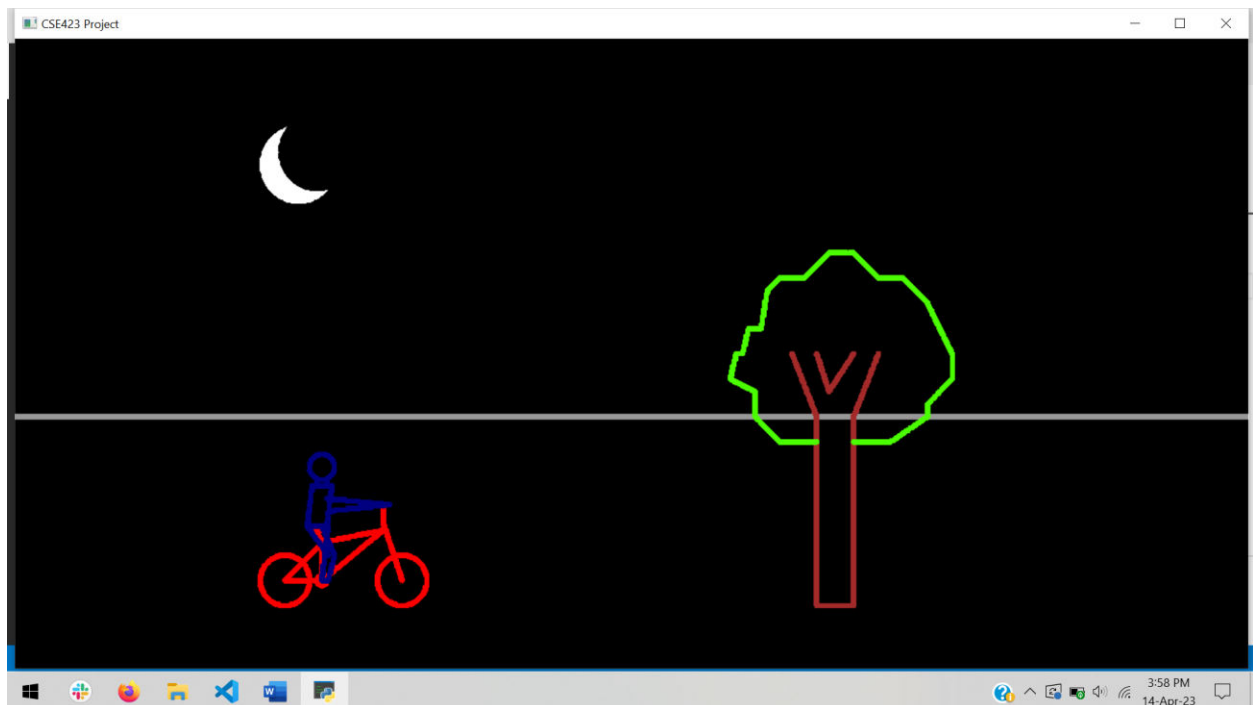
weather: 'rainy'



Input:

distanceTravelled: 200

weather: 'night'



### **Short Description:**

A cyclist is riding a cycle. The distance travelled by the cycle depends on user input. If user inputs distance 515 units or less, then the cycle stops after travelling that specified distance. However, if the user inputs a distance greater than 515 units, then the cycle hits the tree and the tree falls by rotating 90 degrees clockwise. The user can also change the weather or mode and background by providing inputs: sunny, cloudy, rainy and night. Midpoint line and midpoint circle drawing algorithms have been used to draw everything using lines and circles.

Translation: Translation has been applied in the project to place the different objects like man, cycle, tree, sun, moon, clouds etc. in the appropriate positions.

Scaling: Scaling has been used to scale different objects including the clouds in the picture by stretching along x-axis and y-axis by multiplying appropriate numbers with the co-ordinates.

Animation: The man, cycle, tree etc. have been animated. They are made to move by slowly increasing/decreasing the coordinates of the objects inside a for-loop. At the end of every frame, the window is cleared using `glClear()` function. The animation speed is controlled using `time.sleep()` function.

Rotation Transformation: The tree has been made to fall by rotating it by 90 degrees in clockwise direction. Animation has been added to show the motion gradually.