Practical 8: -

Code:

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

float personX = 0.0f, personY = -0.5f;

float rainSpeed = 0.02f;

float dropX[100], dropY[100];

int walkingDirection = 1;

void drawCircle(float x, float y, float radius) {

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(x, y);

for (int i = 0; i <= 360; i++) {

float theta = 2.0f \* 3.14159265358979323846f \* float(i) / 360.0f;

float dx = radius \* cosf(theta);

float dy = radius \* sinf(theta);

glVertex2f(x + dx, y + dy);

}

glEnd();

}

void drawPerson(float x, float y) {

glColor3f(1.0f, 0.8f, 0.6f);

drawCircle(x, y + 0.1f, 0.1f);

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

glBegin(GL\_QUADS);

glVertex2f(x - 0.05f, y - 0.2f);

glVertex2f(x + 0.05f, y - 0.2f);

glVertex2f(x + 0.05f, y - 0.6f);

glVertex2f(x - 0.05f, y - 0.6f);

glEnd();

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

glBegin(GL\_LINES);

glVertex2f(x - 0.05f, y - 0.6f);

glVertex2f(x - 0.1f, y - 0.9f);

glVertex2f(x + 0.05f, y - 0.6f);

glVertex2f(x + 0.1f, y - 0.9f);

glEnd();

glBegin(GL\_LINES);

glVertex2f(x - 0.05f, y - 0.3f);

glVertex2f(x - 0.1f, y - 0.4f);

glVertex2f(x + 0.05f, y - 0.3f);

glVertex2f(x + 0.1f, y - 0.4f);

glEnd();

}

void drawRaindrops() {

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

for (int i = 0; i < 100; i++) {

drawCircle(dropX[i], dropY[i], 0.02f);

}

}

void update(int value) {

personX += walkingDirection \* 0.05f;

if (personX > 1.0f) walkingDirection = -1;

if (personX < -1.0f) walkingDirection = 1;

for (int i = 0; i < 100; i++) {

dropY[i] -= rainSpeed;

if (dropY[i] < -1.0f) {

dropY[i] = 1.0f;

dropX[i] = (rand() % 200 - 100) / 100.0f;

}

}

glutPostRedisplay();

glutTimerFunc(16, update, 0);

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

drawRaindrops();

drawPerson(personX, personY);

glutSwapBuffers();

}

void initOpenGL() {

glClearColor(0.8f, 0.8f, 0.8f, 1.0f);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutCreateWindow("Walking Person in Rain");

initOpenGL();

for (int i = 0; i < 100; i++) {

dropX[i] = (rand() % 200 - 100) / 100.0f;

dropY[i] = (rand() % 200) / 100.0f + 1.0f;

}

glutDisplayFunc(display);

glutTimerFunc(25, update, 0);

glutMainLoop();

return 0;

}

Output:



