**Lab Practice-7**

Submission Guidelines-

* Rename the file with your serial number only
* Must submit within time that will be discussed in class VUES
* Must include resources for all the section in the table

|  |
| --- |
| **Question-**  Create a simple scenario to show weather change. It will start with sunny weather. After sometimes the sky will become overcast. Again, after some time it will start to rain. The weather will change automatically. |
| **Graph** |
| **Code-**  **#include <GL/glut.h>**  **#include <bits/stdc++.h>**  **#include <thread>**  **#include <chrono>**  **int scenario = 0;**  **void display();**  **void updateWeather(int value);**  **void drawHouse() {**    **glColor3f(0.8f, 0.5f, 0.2f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.6f, -0.5f);**  **glVertex2f(-0.2f, -0.5f);**  **glVertex2f(-0.2f, -0.2f);**  **glVertex2f(-0.6f, -0.2f);**  **glEnd();**    **glColor3f(0.9f, 0.1f, 0.1f);**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(-0.65f, -0.2f);**  **glVertex2f(-0.15f, -0.2f);**  **glVertex2f(-0.4f, 0.1f);**  **glEnd();**  **}**  **void drawSunnyBackground() {**  **glColor3f(0.0f, 0.8f, 1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**    **glColor3f(1.0f, 1.0f, 0.0f);**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 360; i++) {**  **float angle = i \* 3.14159f / 180.0f;**  **glVertex2f(0.7f + 0.1f \* cos(angle), 0.7f + 0.1f \* sin(angle));**  **}**  **glEnd();**  **}**  **void drawOvercastBackground() {**  **glColor3f(0.5f, 0.5f, 0.5f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **}**  **void drawRainyBackground() {**  **glColor3f(0.3f, 0.3f, 0.6f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**    **glColor3f(0.8f, 0.8f, 1.0f);**  **for (float x = -1.0f; x < 1.0f; x += 0.1f) {**  **for (float y = -1.0f; y < 1.0f; y += 0.1f) {**  **glBegin(GL\_LINES);**  **glVertex2f(x, y);**  **glVertex2f(x + 0.02f, y - 0.1f);**  **glEnd();**  **}**  **}**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**    **if (scenario == 0) {**  **drawSunnyBackground();**  **} else if (scenario == 1) {**  **drawOvercastBackground();**  **} else if (scenario == 2) {**  **drawRainyBackground();**  **}**  **drawHouse();**  **glFlush();**  **glutSwapBuffers();**  **}**  **void updateWeather(int value) {**  **scenario = (scenario + 1) % 3;**  **glutPostRedisplay();**  **glutTimerFunc(3000, updateWeather, 0);**  **}**  **void init() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glMatrixMode(GL\_PROJECTION);**  **glLoadIdentity();**  **gluOrtho2D(-1.0f, 1.0f, -1.0f, 1.0f);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 600);**  **glutCreateWindow("Weather Change Scenario");**  **init();**  **glutDisplayFunc(display);**  **glutTimerFunc(3000, updateWeather, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Create a simple scenario to show weather change. It will start with sunny weather. After sometimes the sky will become overcast. Again, after some time it will start to rain. Change the weather using keyboard. |
| **Graph** |
| **Code-**  **#include <GL/glut.h>**  **#include <bits/stdc++.h>**  **int scenario = 0;**  **void display();**  **void keyboard(unsigned char key, int x, int y);**  **void drawHouse() {**    **glColor3f(0.8f, 0.5f, 0.2f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.6f, -0.5f);**  **glVertex2f(-0.2f, -0.5f);**  **glVertex2f(-0.2f, -0.2f);**  **glVertex2f(-0.6f, -0.2f);**  **glEnd();**  **// Roof**  **glColor3f(0.9f, 0.1f, 0.1f);**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(-0.65f, -0.2f);**  **glVertex2f(-0.15f, -0.2f);**  **glVertex2f(-0.4f, 0.1f);**  **glEnd();**  **}**  **void drawSunnyBackground() {**  **glColor3f(0.0f, 0.8f, 1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**    **glColor3f(1.0f, 1.0f, 0.0f);**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 360; i++) {**  **float angle = i \* 3.14159f / 180.0f;**  **glVertex2f(0.7f + 0.1f \* cos(angle), 0.7f + 0.1f \* sin(angle));**  **}**  **glEnd();**  **}**  **void drawOvercastBackground() {**  **glColor3f(0.5f, 0.5f, 0.5f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **}**  **void drawRainyBackground() {**  **glColor3f(0.3f, 0.3f, 0.6f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **// Rain drops**  **glColor3f(0.8f, 0.8f, 1.0f);**  **for (float x = -1.0f; x < 1.0f; x += 0.1f) {**  **for (float y = -1.0f; y < 1.0f; y += 0.1f) {**  **glBegin(GL\_LINES);**  **glVertex2f(x, y);**  **glVertex2f(x + 0.02f, y - 0.1f);**  **glEnd();**  **}**  **}**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**    **if (scenario == 0) {**  **drawSunnyBackground();**  **} else if (scenario == 1) {**  **drawOvercastBackground();**  **} else if (scenario == 2) {**  **drawRainyBackground();**  **}**  **drawHouse();**  **glFlush();**  **glutSwapBuffers();**  **}**  **void keyboard(unsigned char key, int x, int y) {**  **if (key == 's' || key == 'S') {**  **scenario = 0;**  **} else if (key == 'o' || key == 'O') {**  **scenario = 1;**  **} else if (key == 'r' || key == 'R') {**  **scenario = 2;**  **}**  **glutPostRedisplay();**  **}**  **void init() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glMatrixMode(GL\_PROJECTION);**  **glLoadIdentity();**  **gluOrtho2D(-1.0f, 1.0f, -1.0f, 1.0f);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 600);**  **glutCreateWindow("Weather Change Scenario");**  **init();**  **glutDisplayFunc(display);**  **glutKeyboardFunc(keyboard);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Write codes to show a first scenario that will have a car moving in a sunny day. As the car goes out of the axis, another scenario will appear where the sky will be overcast. The car will move again in this scenario. When the car goes out of the axis another scenario will be called where it will rain. The car will move through this scenario re-appearing each time it goes out of axis. |
| **Graph** |
| **Code-**  **#include <GL/glut.h>**  **#include <cmath>**  **// Global variables**  **float carX = -1.0f;**  **int scenario = 0;**  **// Function prototypes**  **void display();**  **void timer(int);**  **void drawCar();**  **void drawSunnyBackground();**  **void drawOvercastBackground();**  **void drawRainyBackground();**  **void init();**  **void drawCar() {**  **// Car body (main rectangle)**  **glColor3f(0.0f, 0.0f, 1.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(carX, -0.5f);**  **glVertex2f(carX + 0.4f, -0.5f);**  **glVertex2f(carX + 0.4f, -0.4f);**  **glVertex2f(carX, -0.4f);**  **glEnd();**  **// Car roof**  **glColor3f(0.0f, 0.0f, 0.7f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(carX + 0.1f, -0.4f);**  **glVertex2f(carX + 0.3f, -0.4f);**  **glVertex2f(carX + 0.25f, -0.35f);**  **glVertex2f(carX + 0.15f, -0.35f);**  **glEnd();**  **// Car windows**  **glColor3f(0.5f, 0.8f, 1.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(carX + 0.15f, -0.4f);**  **glVertex2f(carX + 0.25f, -0.4f);**  **glVertex2f(carX + 0.22f, -0.36f);**  **glVertex2f(carX + 0.18f, -0.36f);**  **glEnd();**  **// Wheels**  **glColor3f(0.0f, 0.0f, 0.0f);**  **float radius = 0.05f;**  **int num\_segments = 100;**  **// Front wheel**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < num\_segments; i++) {**  **float theta = 2.0f \* 3.14159f \* float(i) / float(num\_segments);**  **float x = radius \* cosf(theta);**  **float y = radius \* sinf(theta);**  **glVertex2f(carX + 0.1f + x, -0.55f + y);**  **}**  **glEnd();**  **// Rear wheel**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < num\_segments; i++) {**  **float theta = 2.0f \* 3.14159f \* float(i) / float(num\_segments);**  **float x = radius \* cosf(theta);**  **float y = radius \* sinf(theta);**  **glVertex2f(carX + 0.3f + x, -0.55f + y);**  **}**  **glEnd();**  **// Wheel rims**  **glColor3f(0.7f, 0.7f, 0.7f);**  **radius = 0.03f;**  **// Front rim**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < num\_segments; i++) {**  **float theta = 2.0f \* 3.14159f \* float(i) / float(num\_segments);**  **float x = radius \* cosf(theta);**  **float y = radius \* sinf(theta);**  **glVertex2f(carX + 0.1f + x, -0.55f + y);**  **}**  **glEnd();**  **// Rear rim**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < num\_segments; i++) {**  **float theta = 2.0f \* 3.14159f \* float(i) / float(num\_segments);**  **float x = radius \* cosf(theta);**  **float y = radius \* sinf(theta);**  **glVertex2f(carX + 0.3f + x, -0.55f + y);**  **}**  **glEnd();**  **}**  **void drawSunnyBackground() {**  **// Sky**  **glColor3f(0.0f, 0.8f, 1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, 0.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **// Ground**  **glColor3f(0.0f, 0.6f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(-1.0f, 0.0f);**  **glEnd();**  **// Sun**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 360; i++) {**  **float angle = i \* 3.14159f / 180.0f;**  **glVertex2f(0.7f + 0.1f \* cos(angle), 0.7f + 0.1f \* sin(angle));**  **}**  **glEnd();**  **}**  **void drawOvercastBackground() {**  **// Sky with gradient**  **glBegin(GL\_QUADS);**  **glColor3f(0.6f, 0.6f, 0.6f);**  **glVertex2f(-1.0f, 0.0f);**  **glVertex2f(1.0f, 0.0f);**  **glColor3f(0.8f, 0.8f, 0.8f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **// Ground**  **glColor3f(0.4f, 0.4f, 0.4f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(-1.0f, 0.0f);**  **glEnd();**  **}**  **void drawRainyBackground() {**  **// Sky**  **glColor3f(0.2f, 0.2f, 0.4f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, 0.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(1.0f, 1.0f);**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **// Ground**  **glColor3f(0.1f, 0.1f, 0.3f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.0f, -1.0f);**  **glVertex2f(1.0f, -1.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(-1.0f, 0.0f);**  **glEnd();**  **// Rain drops**  **glColor3f(0.8f, 0.8f, 1.0f);**  **for (float x = -1.0f; x < 1.0f; x += 0.1f) {**  **for (float y = 0.5f; y < 1.0f; y += 0.1f) {**  **glBegin(GL\_LINES);**  **glVertex2f(x, y);**  **glVertex2f(x + 0.02f, y - 0.1f);**  **glEnd();**  **}**  **}**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **if (scenario == 0) {**  **drawSunnyBackground();**  **} else if (scenario == 1) {**  **drawOvercastBackground();**  **} else if (scenario == 2) {**  **drawRainyBackground();**  **}**  **drawCar();**  **glFlush();**  **glutSwapBuffers();**  **}**  **void timer(int) {**  **carX += 0.01f;**  **if (carX > 1.0f) {**  **carX = -1.0f;**  **scenario = (scenario + 1) % 3;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(16, timer, 0);**  **}**  **void init() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black background**  **glMatrixMode(GL\_PROJECTION);**  **glLoadIdentity();**  **gluOrtho2D(-1.0f, 1.0f, -1.0f, 1.0f);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 600);**  **glutCreateWindow("Car Moving Through Different Scenarios");**  **init();**  **glutDisplayFunc(display);**  **glutTimerFunc(0, timer, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |