**Lab Taks-6**

Submission Guidelines-

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

|  |
| --- |
| **Question-**  Develop an animation that will change the background color of the window after 20ms. Use at least two different colors. |
| **Graph-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void triangle();**  **void rectangle();**  **void display1(int a)**  **{**  **glutDisplayFunc(triangle);**  **glutPostRedisplay();**  **}**  **void rectangle()**  **{**  **glClearColor(1.0f, 1.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glEnd();**  **glFlush();**  **glutTimerFunc(500,display1,0);**  **}**  **void display2(int a)**  **{**  **glutDisplayFunc(rectangle);**  **glutPostRedisplay();**  **}**  **void triangle()**  **{**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glBegin(GL\_POLYGON);**  **glEnd();**  **glFlush();**  **glutTimerFunc(500,display2,0);**  **}**  **void display() {**  **rectangle();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("BackGround Change");**  **glutInitWindowSize(320, 320);**  **glutDisplayFunc(display);**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  A computer screen shot of a black screen  Description automatically generated |

|  |
| --- |
| **Question-**  Develop an animation that will call four objects separately, each after 20 ms. |
| **Graph-** |
| **Code-#include <windows.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **void circle(float radius, float xc, float yc, float r, float g, float b)**  **{**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(r,g,b);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=radius;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+xc,y+yc);**  **}**  **glEnd();**  **}**  **void triangle();**  **void rectangle();**  **void circle1();**  **void display4(int a){**  **glutDisplayFunc(rectangle);**  **glutPostRedisplay();**  **}**  **void circle2(){**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **circle(3,0,0,1,0,0);**  **glFlush();**  **glutTimerFunc(500,display4,0);**  **}**  **void display3(int a){**  **glutDisplayFunc(circle2);**  **glutPostRedisplay();**  **}**  **void circle1(){**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **circle(3,0,0,1,0,0);**  **glFlush();**  **glutTimerFunc(500,display3,0);**  **}**  **void display1(int a)**  **{**  **glutDisplayFunc(triangle);**  **glutPostRedisplay();**  **}**  **void rectangle()**  **{**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(2.0f, 0.0f);**  **glVertex2f(2.0f, 3.0f);**  **glVertex2f(0.0f, 3.0f);**  **glEnd();**  **glFlush();**  **glutTimerFunc(500,display1,0);**  **}**  **void display2(int a)**  **{**  **glutDisplayFunc(circle1);**  **glutPostRedisplay();**  **}**  **void triangle()**  **{**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5f, 0.0f, 1.0f);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(2.0f, 0.0f);**  **glVertex2f(2.0f, 3.0f);**  **glEnd();**  **glFlush();**  **glutTimerFunc(500,display2,0);**  **}**  **void display() {**  **rectangle();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("4 object");**  **glutInitWindowSize(320, 320);**  **glutDisplayFunc(display);**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  A computer screen with a black screen  Description automatically generated |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). The objects will move to the left, right, up and down in a loop. |
| **Graph-A graph with circles and circles with blue dots  Description automatically generated** |
| **Code-#include <windows.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **// Position variables for each object**  **float circleY = 0.0f;**  **float rectX = 0.0f;**  **float triangleX = 0.0f;**  **float yellowCircleY = 0.0f;**  **bool circleDirection = true;**  **bool rectDirection = true;**  **bool triangleDirection = true;**  **bool yellowCircleDirection = true;**  **void circle(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 200; i++) {**  **float pi = 3.1416;**  **float angle = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(angle);**  **float y = radius \* sin(angle);**  **glColor3f(r, g, b);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **void rectangle(float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **glColor3f(r, g, b);**  **glVertex2f(xc - 1.0f, yc - 0.5f);**  **glVertex2f(xc + 1.0f, yc - 0.5f);**  **glVertex2f(xc + 1.0f, yc + 0.5f);**  **glVertex2f(xc - 1.0f, yc + 0.5f);**  **glEnd();**  **}**  **void triangle(float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **glColor3f(r, g, b);**  **glVertex2f(xc, yc + 1.5f);**  **glVertex2f(xc - 1.0f, yc - 1.0f);**  **glVertex2f(xc + 1.0f, yc - 1.0f);**  **glEnd();**  **}**  **void updatePositions(int value) {**  **if (circleDirection)**  **circleY += 0.1f;**  **else**  **circleY -= 0.1f;**  **if (circleY >= 5.0f || circleY <= -5.0f)**  **circleDirection = !circleDirection;**  **if (rectDirection)**  **rectX -= 0.1f;**  **else**  **rectX += 0.1f;**  **if (rectX >= 5.0f || rectX <= -5.0f)**  **rectDirection = !rectDirection;**  **if (triangleDirection)**  **triangleX += 0.1f;**  **else**  **triangleX -= 0.1f;**  **if (triangleX >= 5.0f || triangleX <= -5.0f)**  **triangleDirection = !triangleDirection;**  **if (yellowCircleDirection)**  **yellowCircleY -= 0.1f;**  **else**  **yellowCircleY += 0.1f;**  **if (yellowCircleY >= 5.0f || yellowCircleY <= -5.0f)**  **yellowCircleDirection = !yellowCircleDirection;**  **glutPostRedisplay();**  **glutTimerFunc(50, updatePositions, 0);**  **}**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **circle(0.5f, 0.0f, circleY, 1.0f, 0.0f, 0.0f);**  **rectangle(rectX, 0.0f, 0.0f, 1.0f, 0.0f);**  **triangle(triangleX, 0.0f, 0.0f, 0.0f, 1.0f);**  **circle(0.5f, 0.0f, yellowCircleY, 1.0f, 1.0f, 0.0f);**  **glFlush();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**  **glutInitWindowSize(640, 480);**  **glutInitWindowPosition(100, 100);**  **glutCreateWindow("Simplified 4 Object Animation");**  **gluOrtho2D(-10, 10, -10, 10);**  **glutDisplayFunc(display);**  **glutTimerFunc(50, updatePositions, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). Four different keys will be dedicated each objects. The objects will move to the left, right, up and down in a loop as the keys are pressed individually. |
| **Graph-A graph with circles and circles  Description automatically generated** |
| **Code-#include <windows.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **float positions[4][2] = {{0.0f, 0.0f}, {0.0f, 0.0f}, {0.0f, 0.0f}, {0.0f, 0.0f}};**  **float \_circle = 0.0f;**  **float \_circle2 = 0.0f;**  **float \_rectangle = 0.0f;**  **float \_triangle = 0.0f;**  **void circle(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 200; i++) {**  **float pi = 3.1416;**  **float angle = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(angle);**  **float y = radius \* sin(angle);**  **glColor3f(r, g, b);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **void rectangle(float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **glColor3f(r, g, b);**  **glVertex2f(xc - 1.0f, yc - 0.5f);**  **glVertex2f(xc + 1.0f, yc - 0.5f);**  **glVertex2f(xc + 1.0f, yc + 0.5f);**  **glVertex2f(xc - 1.0f, yc + 0.5f);**  **glEnd();**  **}**  **void triangle(float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **glColor3f(r, g, b);**  **glVertex2f(xc, yc + 1.5f);**  **glVertex2f(xc - 1.0f, yc - 1.0f);**  **glVertex2f(xc + 1.0f, yc - 1.0f);**  **glEnd();**  **}**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **circle(0.5f, 0.0, \_circle, 1.0f, 0.0f, 0.0f);**  **rectangle(0.0f, \_rectangle, 0.0f, 1.0f, 0.0f);**  **triangle(\_triangle, 0.0f, 0.0f, 0.0f, 1.0f);**  **circle(0.5f, \_circle2,0.0f, 1.0f, 1.0f, 0.0f);**  **glFlush();**  **}**  **void keyboard(unsigned char key, int x, int y) {**  **float step = 0.5f;**  **switch (key) {**  **case 'w':**  **\_circle += step;**  **if (\_circle > 5.0f) \_circle = -5.0f;**  **break;**  **case 's':**  **\_circle -= step;**  **if (\_circle > 5.0f) \_circle = 5.0f;**  **break;**  **case 'a':**  **\_rectangle -= step;**  **if (\_rectangle < -5.0f) \_rectangle = 5.0f;**  **break;**  **case 'd':**  **\_rectangle += step;**  **if (\_rectangle > +5.0f) \_rectangle = -5.0f;**  **break;**  **case 'i':**  **\_triangle += step;**  **if (\_triangle > 5.0f) \_triangle = -5.0f;**  **break;**  **case 'k':**  **\_triangle -= step;**  **if (\_triangle < -5.0f) \_triangle = 5.0f;**  **break;**  **case 'j':**  **\_circle2 += step;**  **if (\_circle2 > 5.0f) \_circle2 = -5.0f;**  **break;**  **case 'l':**  **\_circle2 -= step;**  **if (\_circle2 > 5.0f) \_circle2 = +5.0f;**  **break;**  **default:**  **break;**  **}**  **glutPostRedisplay();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**  **glutInitWindowSize(640, 480);**  **glutInitWindowPosition(100, 100);**  **glutCreateWindow("Object Movement with Keyboard");**  **gluOrtho2D(-10, 10, -10, 10);**  **glutDisplayFunc(display);**  **glutKeyboardFunc(keyboard);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A computer screen with a black square with colorful dots and lines  Description automatically generated** |

|  |
| --- |
| **Question-**  Develop a scenario where it will rain and gradually create flood |
| **Graph-** |
| **Code-**  **#include <GL/glut.h>**  **#include <cstdlib>**  **#include <ctime>**  **float raindropY = 10.0f;**  **float waterLevel = -10.0f;**  **void drawRaindrop(float x, float y) {**  **glPointSize(10.0f);**  **glBegin(GL\_POINTS);**  **glColor3f(0.0f, 0.0f, 1.0f);**  **glVertex2f(x, y);**  **glEnd();**  **}**  **void drawWater(float level) {**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 1.0f);**  **glVertex2f(-10.0f, level);**  **glVertex2f(10.0f, level);**  **glVertex2f(10.0f, -10.0f);**  **glVertex2f(-10.0f, -10.0f);**  **glEnd();**  **}**  **void update(int value) {**  **raindropY -= 0.2f;**  **if (raindropY < waterLevel + 0.2f) {**  **raindropY = 10.0f;**  **}**  **waterLevel += 0.01f;**  **if (waterLevel >= 10.0f) {**  **waterLevel = 10.0f;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(50, update, 0);**  **}**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **drawWater(waterLevel);**  **drawRaindrop(0.0f, raindropY);**  **drawRaindrop(1.0f, raindropY); drawRaindrop(2.0f, raindropY);**  **drawRaindrop(-1.0f, raindropY); drawRaindrop(-2.0f, raindropY);**  **drawRaindrop(3.0f, raindropY); drawRaindrop(4.0f, raindropY);**  **drawRaindrop(-3.0f, raindropY); drawRaindrop(-4.0f, raindropY);**  **drawRaindrop(5.0f, raindropY); drawRaindrop(6.0f, raindropY);**  **drawRaindrop(-5.0f, raindropY); drawRaindrop(-5.0f, raindropY);**  **drawRaindrop(7.0f, raindropY); drawRaindrop(8.0f, raindropY);**  **drawRaindrop(-7.0f, raindropY); drawRaindrop(-6.0f, raindropY);**  **glFlush();**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**  **glutInitWindowSize(800, 600);**  **glutInitWindowPosition(100, 100);**  **glutCreateWindow("Rain and Flood Simulation");**  **gluOrtho2D(-10.0, 10.0, -10.0, 10.0);**  **glutDisplayFunc(display);**  **glutTimerFunc(50, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |