**Lab Practice-7**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.

|  |
| --- |
| **Question-**  Rain effect |
| **Graph-** |
| **Code-**  **#include <iostream>**  **#include<windows.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **//water**  **void rainy\_river();**  **int water = 0;**  **float waterX = 0;**  **float waterY = 0;**  **float boatx=0;**  **float boaty=0;**  **//rain**  **int rains=0;**  **int x=0;**  **int y=0;**  **float wx=0;**  **float wy=0;**  **void draw\_line(float cx, float cy, float r, int num)**  **{**  **glBegin(GL\_TRIANGLE\_FAN);**  **for (int i = 0; i < num; i++)**  **{**  **float angle = 2.0f \* 3.1416f \* float(i) / float(num);//get current angle**  **float x = r \* cosf(angle);//calculate x**  **float y = r \* sinf(angle);//calculate y**  **glVertex2f(x + cx, y + cy);//output vertex**  **}**  **glEnd();**  **glFlush();**  **}**  **void river()**  **{**  **glColor3ub (33, 232, 229);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-6, 0);**  **glVertex2f(-6, 3);**  **glVertex2f(6, 3);**  **glVertex2f(6, 0);**  **glEnd();**  **}**  **void land()**  **{**  **glColor3ub (194, 148, 23);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-6, -3);**  **glVertex2f(-6, 0);**  **glVertex2f(6, 0);**  **glVertex2f(6, -3);**  **glEnd();**  **}**  **void water1()**  **{**  **if(rains==0)**  **{**  **//glColor3ub (169, 242, 242);**  **glColor4f(1.0f, 1.0f, 1.0f, 0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(-5,2);**  **glVertex2f(-4,2);**  **glVertex2f(-4,1);**  **glVertex2f(-3,1);**  **glVertex2f(-3,2);**  **glVertex2f(-2,2);**  **glVertex2f(-2,1);**  **glVertex2f(-1,1);**  **glVertex2f(-1,2);**  **glVertex2f(0,2);**  **glVertex2f(0,1);**  **glVertex2f(1,1);**  **glVertex2f(1,2);**  **glVertex2f(2,2);**  **glVertex2f(2,1);**  **glVertex2f(3,1);**  **glVertex2f(3,2);**  **glVertex2f(4,2);**  **glVertex2f(4,1);**  **glVertex2f(5,1);**  **glEnd();**  **}**  **}**  **void move\_water(int x){**  **if (water ==1)**  **{**  **waterX += 0.01;//left or right side move water**  **}**  **if (waterX>2)**  **{**  **waterX = -5;**  **}**  **glPushMatrix();**  **glTranslatef(waterX, waterY, 0);**  **if(x==1)**  **{**  **rainy\_river();**  **water1();**  **}**  **else{**  **rainy\_river();**  **}**  **glPopMatrix();**  **glFlush();**  **}**  **void rain()**  **{**  **glColor3d(0,1,0);**  **glBegin(GL\_POINTS);**  **for(int i=1;i<=10000;i++)**  **{**  **x=rand(),y=rand();**  **x%=1000; y%=30;**  **glBegin(GL\_LINES);**  **glColor3b(1,1,1);**  **glVertex2d(x,y);**  **glVertex2d(x+0.3,y+0.3);**  **glEnd();**  **}**  **for(int i=1;i<=10000;i+=1)**  **{**  **x=rand(),y=rand();**  **x%=10000; y%=2000;**  **glBegin(GL\_LINES);**  **glColor3b(1,1,1);**  **glVertex2d(x,y);**  **glVertex2d(x+0.3,y+0.3);**  **glEnd();**  **}**  **}**  **void move\_rain(int x)**  **{**  **if(rains ==1)**  **{**  **wx -= 0.01;**  **wy -= 0.01;**  **glPushMatrix();**  **glTranslatef(wx, wy, 0);**  **if (x==1)**  **{**  **rain();**  **}**  **glPopMatrix();**  **glFlush();**  **}**  **}**  **void rainy\_river()**  **{**  **if(rains==1)**  **{**  **int xaxis=20;**  **int yaxis=10;**  **glColor3ub (160, 220, 220);**  **glBegin(GL\_LINES);**  **for(int i=0; i<6; i++)**  **{**  **for(int j=0; j<15; j++){**  **glVertex2f( xaxis-0.5, yaxis-0.1);**  **glVertex2f( xaxis-1.5, yaxis);**  **glVertex2f( xaxis, yaxis);**  **glVertex2f( xaxis-0.5, yaxis-0.1);**  **xaxis-=1.5;**  **}**  **yaxis-=0.5;**  **xaxis=20;**  **}**  **glEnd();**  **}**  **}**  **void display(void)**  **{**  **river();**  **land();**  **move\_water(1);**  **move\_rain(1);**  **glFlush();**  **glutPostRedisplay();**  **glutSwapBuffers();**  **}**  **void keyboard(unsigned char key, int x, int y)**  **{**  **if (key == 'C' || key == 'c')**  **{ //water string**  **water = 1;**  **}**  **if (key == 'D' || key == 'd')**  **{ //water stop**  **water = 0;**  **}**  **else if (key == '5' )**  **{ //rain starting**  **rains = 1;**  **}**  **else if (key == '3' )**  **{ //rain stop**  **rains = 0;**  **}**  **}**  **void drawscene(void)**  **{**  **glClearColor(1.0f,1.0f,1.0f,1.0f);**  **glColor3f(.0f,.0f,.0f);**  **glLoadIdentity();**  **gluOrtho2D(-6,6,-3,3);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(320,320);**  **glutCreateWindow("Rain Scenery");**  **glutPostRedisplay();**  **glutDisplayFunc(display);**  **glutKeyboardFunc(keyboard);**  **drawscene();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Create a scenerio where there will be ground, river and it will rain. The rain will ooverflow the  river thay will cause flood. |
| **Graph** |
| **Code-**  **#include<iostream>**  **#include<windows.h>**  **#include<GL/glut.h>**  **#include<math.h>**  **//water**  **void rainy\_river();**  **int water = 0;**  **float waterX = 0;**  **float waterY = 0;**  **float boatx=0;**  **float boaty=0;**  **float \_move=0;**  **//rain**  **int rains=0;**  **int x=0;**  **int y=0;**  **float wx=0;**  **float wy=0;**  **void draw\_line(float cx, float cy, float r, int num)**  **{**  **glBegin(GL\_TRIANGLE\_FAN);**  **for (int i = 0; i < num; i++)**  **{**  **float angle = 2.0f \* 3.1416f \* float(i) / float(num);**  **float x = r \* cosf(angle);**  **float y = r \* sinf(angle);**  **glVertex2f(x + cx, y + cy);**  **}**  **glEnd();**  **glFlush();**  **}**  **void river()**  **{**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3ub (33, 232, 229);**  **glVertex2f(-6, 0);**  **glVertex2f(-6, 3);**  **glVertex2f(6, 3);**  **glVertex2f(6, 0);**  **glEnd();**  **if(rains==1)**  **{**  **glTranslatef(0,\_move,0);**  **glBegin(GL\_POLYGON);**  **glColor3ub (33, 232, 229);**  **glVertex2f(-6, 0);**  **glVertex2f(-6, 3);**  **glVertex2f(6, 3);**  **glVertex2f(6, 0);**  **glEnd();**  **}**  **glPopMatrix();**  **}**  **void land()**  **{**  **glColor3ub (194, 148, 23);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-6, -3);**  **glVertex2f(-6, 0);**  **glVertex2f(6, 0);**  **glVertex2f(6, -3);**  **glEnd();**  **}**  **void water1()**  **{**  **if(rains==0)**  **{**  **glColor4f(1.0f, 1.0f, 1.0f, 0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(-5,2);**  **glVertex2f(-4,2);**  **glVertex2f(-4,1);**  **glVertex2f(-3,1);**  **glVertex2f(-3,2);**  **glVertex2f(-2,2);**  **glVertex2f(-2,1);**  **glVertex2f(-1,1);**  **glVertex2f(-1,2);**  **glVertex2f(0,2);**  **glVertex2f(0,1);**  **glVertex2f(1,1);**  **glVertex2f(1,2);**  **glVertex2f(2,2);**  **glVertex2f(2,1);**  **glVertex2f(3,1);**  **glVertex2f(3,2);**  **glVertex2f(4,2);**  **glVertex2f(4,1);**  **glVertex2f(5,1);**  **glEnd();**  **}**  **}**  **void move\_water(int x)**  **{**  **if (water ==1)**  **{**  **waterX += 0.01;**  **}**  **if (waterX>2)**  **{**  **waterX = -5;**  **}**  **glPushMatrix();**  **glTranslatef(waterX, waterY, 0);**  **if(x==1)**  **{**  **rainy\_river();**  **water1();**  **}**  **else**  **{**  **rainy\_river();**  **}**  **glPopMatrix();**  **glFlush();**  **}**  **void rain()**  **{**  **glColor3d(0,1,0);**  **glBegin(GL\_POINTS);**  **for(int i=1;i<=50000;i++)**  **{**  **x=rand(),y=rand();**  **x%=5000; y%=50;**  **glBegin(GL\_LINES);**  **glColor3b(1,1,1);**  **glVertex2d(x,y);**  **glVertex2d(x+0.3,y+0.3);**  **glEnd();**  **}**  **for(int i=1;i<=50000;i+=1)**  **{**  **x=rand(),y=rand();**  **x%=5000; y%=20;**  **glBegin(GL\_LINES);**  **glColor3b(1,1,1);**  **glVertex2d(x,y);**  **glVertex2d(x+0.3,y+0.3);**  **glEnd();**  **}**  **}**  **void move\_rain(int x)**  **{**  **if(rains ==1)**  **{**  **wx -= 0.02;**  **wy -= 0.02;**  **glPushMatrix();**  **glTranslatef(wx, wy, 0);**  **if (x==1)**  **{**  **rain();**  **}**  **glPopMatrix();**  **glFlush();**  **}**  **}**  **void rainy\_river()**  **{**  **if(rains==1)**  **{**  **int xaxis=20;**  **int yaxis=10;**  **glColor3ub (160, 220, 220);**  **glBegin(GL\_LINES);**  **for(int i=0; i<6; i++)**  **{**  **for(int j=0; j<15; j++)**  **{**  **glVertex2f( xaxis-0.5, yaxis-0.1);**  **glVertex2f( xaxis-1.5, yaxis);**  **glVertex2f( xaxis, yaxis);**  **glVertex2f( xaxis-0.5, yaxis-0.1);**  **xaxis-=1.5;**  **}**  **yaxis-=0.5;**  **xaxis=20;**  **}**  **glEnd();**  **}**  **}**  **void update(int value)**  **{**  **\_move += .02;**  **if(\_move > 6)**  **{**  **\_move = -5;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **void display(void)**  **{**  **land();**  **river();**  **move\_water(1);**  **move\_rain(1);**  **glFlush();**  **glutPostRedisplay();**  **glutSwapBuffers();**  **}**  **void keyboard(unsigned char key, int x, int y)**  **{**  **if (key == 'R' || key == 'r')**  **{**  **water = 1;//water start**  **}**  **if (key == 'S' || key == 's')**  **{**  **water = 0;//water flow**  **}**  **else if (key == '5' )**  **{**  **rains = 1;//rain start**  **}**  **else if (key == '3' )**  **{**  **rains = 0;//rain stop**  **}**  **}**  **void drawscene(void)**  **{**  **glClearColor(1.0f,1.0f,1.0f,1.0f);**  **glColor3f(.0f,.0f,.0f);**  **glLoadIdentity();**  **gluOrtho2D(-6,6,-3,3);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(320,320);**  **glutCreateWindow("Flood Scenery");**  **glutPostRedisplay();**  **glutDisplayFunc(display);**  **glutKeyboardFunc(keyboard);**  **drawscene();**  **glutTimerFunc(20, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
|  |
|  |
|  |
|  |

|  |
| --- |
| Create a zoom in and out animation using scaling. |
| Graph- |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **float \_move=1;**  **int zoom=2;**  **void StreetLamp()**  **{**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glScalef(\_move,\_move,0);**  **//3**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142);**  **glVertex2f(164,45.9);**  **glVertex2f(164,43.9);**  **glVertex2f(177.8,43.9);**  **glVertex2f(177.8,45.9);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(1,1,0);**  **glVertex2f(166,32);**  **glVertex2f(164,43.9);**  **glVertex2f(177.8,43.9);**  **glVertex2f(176,32);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(166,30);**  **glVertex2f(166,32);**  **glVertex2f(176,32);**  **glVertex2f(176,30);**  **glEnd();**  **//1**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142);**  **glVertex2f(121.9,45.9);**  **glVertex2f(121.9,43.9);**  **glVertex2f(136.4,43.9);**  **glVertex2f(136.4,45.9);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(1,1,0);**  **glVertex2f(124,32);**  **glVertex2f(121.9,43.9);**  **glVertex2f(136.4,43.9);**  **glVertex2f(134,32);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(124,30);**  **glVertex2f(124,32);**  **glVertex2f(134,32);**  **glVertex2f(134,30);**  **glEnd();**  **//2**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142);**  **glVertex2f(142.2,54.2);**  **glVertex2f(142.2,56.2);**  **glVertex2f(157.8,56.2);**  **glVertex2f(157.8,54.2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(1,1,0);**  **glVertex2f(144.3,42.3);**  **glVertex2f(142.2,54.2);**  **glVertex2f(157.8,54.4);**  **glVertex2f(155.9,42.2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142);**  **glVertex2f(144.3,40.2);**  **glVertex2f(144.3,42.3);**  **glVertex2f(155.9,42.3);**  **glVertex2f(155.9,40.2);**  **glEnd();**  **//m**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(148.4,-11.8);**  **glVertex2f(148.5,40.2);**  **glVertex2f(151.4,40.2);**  **glVertex2f(151.3,-11.8);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(145,-15);**  **glVertex2f(144.9,-11.9);**  **glVertex2f(154.9,-11.9);**  **glVertex2f(155,-15);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(142.8,-18.8);**  **glVertex2f(142.8,-15);**  **glVertex2f(157.1,-15);**  **glVertex2f(157.1,-18.8);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(170,26);**  **glVertex2f(170,30);**  **glVertex2f(172,30);**  **glVertex2f(172,23.01);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(128,23.01);**  **glVertex2f(130,26);**  **glVertex2f(170,26);**  **glVertex2f(172,23.01);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(124,30);**  **glVertex2f(134,32);**  **glVertex2f(134,32);**  **glVertex2f(124,30);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(128,23.01);**  **glVertex2f(128,30);**  **glVertex2f(130,30);**  **glVertex2f(130,26);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(170,45);**  **glVertex2f(170,48);**  **glVertex2f(172,48);**  **glVertex2f(172,45);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(128,45);**  **glVertex2f(128,48);**  **glVertex2f(130,48);**  **glVertex2f(130,45);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(144, 143, 142 );**  **glVertex2f(149,56);**  **glVertex2f(149,59);**  **glVertex2f(151,59);**  **glVertex2f(151,56);**  **glEnd();**  **glPopMatrix();**  **}**  **void update(int value)**  **{**  **if(zoom==1)**  **{**  **\_move += .002;**  **if(\_move > 2)**  **{**  **\_move=1.5;**  **}**  **}**  **if(zoom==0)**  **{**  **\_move -= .002;**  **if(\_move < 1)**  **{**  **\_move=0.5;**  **}**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **void keyboard(unsigned char key, int x, int y)**  **{**  **if (key == 'Z' || key == 'z'){ //zoom in**  **zoom = 1;**  **}**  **if (key == 'O' || key == 'o'){ //zoom out**  **zoom = 0;**  **}**  **}**  **void display()**  **{**  **glClearColor(0,0,0,0);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **StreetLamp();**  **glFlush();**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutCreateWindow("OpenGL Scenery");**  **glutInitWindowSize(320,320);**  **glutDisplayFunc(display);**  **glutKeyboardFunc(keyboard);**  **gluOrtho2D(10,400,-100,200);**  **glutTimerFunc(20, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| Output- |

|  |
| --- |
| Create an animation that will show a story in a texual form based on a still image. Your animation will start with a still image and texts will animated on the screen to describe the image. |
| Graph- |
| Code- |
| Output- |