**A scenery with moving particles**

**Members**

A. N. M. JuBaer (162-15-7850)

Md. Ashikur Rahman (162-15-7723)

Abu Sayem (162-15-7682)

Samiul Islam (162-15-7867)

* 1. **Snippets**

1. GL\_POLYGON
2. GL\_TRIANGLE\_FAN
3. Circle
   1. X = Radius \* cos(a)
   2. Y = Radius \* sin(a)
   3. a = 2 \* PI \* Points
4. GL\_LINES
5. GL\_QUDAS
6. glTranslate
7. glLoadIdentity
8. glScaled
9. glutTimerFunc
10. glutSwapBuffers

**Code**

#include <Gl/glu.h>

#include <Gl/glut.h>

#include <math.h>

float cloud1\_pos = 0.0, cloud2\_pos = 0.0, birdsPos = -1.8, birds2Pos = -1.6;

float boatPos = 1.8, boat2Pos = -3.1;

int cloud1State = 1, cloud2State = 1;

void sea()

{

//Sea

glBegin(GL\_POLYGON);

glColor3f(0.0096,0.8174,0.96);

glVertex2f(-1.0,-0.33);

glVertex2f(-1.0,0.33);

glVertex2f(1.0,0.33);

glVertex2f(1.0,-0.33);

glEnd();

}

void land()

{

glBegin(GL\_POLYGON);

glColor3f(0.92,0.8019,0.598);

glVertex2f(-1.0,-0.33);

glVertex2f(1.0,-0.33);

glVertex2f(1.0,-1.0);

glVertex2f(-1.0,-1.0);

glEnd();

}

void sun()

{

int vertices = 50;

GLfloat radius = 0.1;

GLfloat pi = 3.1418;

GLfloat xx = -0.5, xy = 0.5;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(0.99,0.9816,0.4851);

glVertex2f(xx,xy);

for(int i = 0; i<= vertices; i++)

{

glVertex2f(xx+(radius\*cos(i\*2\*pi/vertices)),

xy+(radius\*sin(i\*2\*pi/vertices)) );

}

glEnd();

}

void grass()

{

glLineWidth(4);

glBegin(GL\_LINES);

glColor3f(0.1204,0.75,0.0525);

glVertex2f(-.05f, -0.65f);

glVertex2f(-0.0f, -0.7f);

glVertex2f(0.05f, -0.65f);

glVertex2f(0.0f, -0.7f);

glVertex2f(0.027f, -0.62f);

glVertex2f(0.0f, -0.7f);

glVertex2f(-0.027f, -0.62f);

glVertex2f(0.0f, -0.7f);

glVertex2f(-0.0f, -0.6f);

glVertex2f(0.0f, -0.7f);

glVertex2f(-0.075f, -0.67f);

glVertex2f(-0.0f, -0.7f);

glVertex2f(0.075f, -0.67f);

glVertex2f(-0.0f, -0.7f);

glEnd();

int i;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3.1418;

GLfloat e=-.05f; GLfloat f=-.65f; GLfloat radius11 =.02f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(255, 0, 0);

glVertex2f(e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e + (radius11 \* cos(i \* twicePi / triangleAmount)),

f+ (radius11 \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=0.05f; GLfloat h=-0.65f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(0, 255, 0);

glVertex2f(g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

g + (radius11 \* cos(i \* twicePi / triangleAmount)),

h+ (radius11 \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a1=0.0f; GLfloat b1=-0.6f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(0, 0, 255);

glVertex2f(a1, b1); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

a1 + (radius11 \* cos(i \* twicePi / triangleAmount)),

b1 + (radius11 \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

}

void cloud1()

{

int i;

GLfloat x=.5f; GLfloat y=.86f; GLfloat radius =.05f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3.1418;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud1\_pos + x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud1\_pos + x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a=.55f; GLfloat b=.87f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud1\_pos + a, b); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud1\_pos + a + (radius \* cos(i \* twicePi / triangleAmount)),

b + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c=.45f; GLfloat d=.83f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud1\_pos + c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud1\_pos + c + (radius \* cos(i \* twicePi / triangleAmount)),

d + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e=.52f; GLfloat f=.8f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud1\_pos + e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud1\_pos + e + (radius \* cos(i \* twicePi / triangleAmount)),

f+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=.6f; GLfloat h=.82f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud1\_pos + g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud1\_pos + g + (radius \* cos(i \* twicePi / triangleAmount)),

h+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

}

void cloud2()

{

glTranslatef(0.0,-0.25,0.0);

int i;

GLfloat x=.5f; GLfloat y=.86f; GLfloat radius =.05f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3.1418;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud2\_pos + x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud2\_pos + x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a=.55f; GLfloat b=.87f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud2\_pos + a, b); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud2\_pos + a + (radius \* cos(i \* twicePi / triangleAmount)),

b + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c=.45f; GLfloat d=.83f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud2\_pos + c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud2\_pos + c + (radius \* cos(i \* twicePi / triangleAmount)),

d + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e=.52f; GLfloat f=.8f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud2\_pos + e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud2\_pos + e + (radius \* cos(i \* twicePi / triangleAmount)),

f+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=.6f; GLfloat h=.82f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0,1.0,1.0);

glVertex2f(cloud2\_pos + g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(cloud2\_pos + g + (radius \* cos(i \* twicePi / triangleAmount)),

h+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glLoadIdentity();

}

void bird2()

{

int i;

//1st Bird

GLfloat mm=0.182; GLfloat nn=.80; GLfloat radiusmm =.01;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3.1418;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(0.4223, 0.0246, 0.82);

glVertex2f(birds2Pos + mm, nn); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(birds2Pos + mm + (radiusmm \* cos(i \* twicePi / triangleAmount)),

nn + (radiusmm \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.4223, 0.0246, 0.82);

glVertex2f(birds2Pos + 0.1,0.8);

glVertex2f(birds2Pos + 0.11,0.79);

glVertex2f(birds2Pos + 0.12,0.78);

glVertex2f(birds2Pos + 0.16,0.77);

glVertex2f(birds2Pos + 0.19,0.79);

glVertex2f(birds2Pos + 0.201,0.8);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.4223, 0.0246, 0.82);

glVertex2f(birds2Pos + 0.175,0.8);

glVertex2f(birds2Pos + 0.15,0.8);

glVertex2f(birds2Pos + 0.14,0.84);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.7584, 0.5568, 0.96);

glVertex2f(birds2Pos + 0.175,0.8);

glVertex2f(birds2Pos + 0.144,0.8);

glVertex2f(birds2Pos + 0.12,0.83);

glEnd();

}

void bird()

{

int i;

//1st Bird

GLfloat mm=0.182; GLfloat nn=.80; GLfloat radiusmm =.01;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3.1418;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(0.8, 0.3776, 0.008);

glVertex2f(birdsPos + mm, nn); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(birdsPos + mm + (radiusmm \* cos(i \* twicePi / triangleAmount)),

nn + (radiusmm \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.8, 0.3776, 0.008);

glVertex2f(birdsPos + 0.1,0.8);

glVertex2f(birdsPos + 0.11,0.79);

glVertex2f(birdsPos + 0.12,0.78);

glVertex2f(birdsPos + 0.16,0.77);

glVertex2f(birdsPos + 0.19,0.79);

glVertex2f(birdsPos + 0.201,0.8);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.8, 0.3776, 0.008);

glVertex2f(birdsPos + 0.175,0.8);

glVertex2f(birdsPos + 0.15,0.8);

glVertex2f(birdsPos + 0.14,0.84);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(1, 0.648, 0.34);

glVertex2f(birdsPos + 0.175,0.8);

glVertex2f(birdsPos + 0.144,0.8);

glVertex2f(birdsPos + 0.12,0.83);

glEnd();

//2nd bird

GLfloat mmm=0.062f; GLfloat nnn=.801f + 0.05; GLfloat radiusmmm =.01f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(0.0132,0.44,0.0132);

glVertex2f(birdsPos + mmm, nnn); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(birdsPos + mmm + (radiusmmm \* cos(i \* twicePi / triangleAmount)),

nnn + (radiusmmm \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0132,0.44,0.0132);

glVertex2f(birdsPos + -0.02f,0.8f + 0.05);

glVertex2f(birdsPos + -0.01f,0.79f + 0.05);

glVertex2f(birdsPos + 0.0f,0.78f + 0.05);

glVertex2f(birdsPos + 0.04f,0.77f + 0.05);

glVertex2f(birdsPos + 0.07f,0.79f + 0.05);

glVertex2f(birdsPos + 0.081f,0.8f + 0.05);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.0132,0.44,0.0132 + 0.05);

glVertex2f(birdsPos + 0.055f,0.8f + 0.05);

glVertex2f(birdsPos + 0.03f,0.8f + 0.05);

glVertex2f(birdsPos + 0.02f,0.84f + 0.05);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.3168, 0.72, 0.3168 + 0.05);

glVertex2f(birdsPos + 0.055f,0.8f + 0.05);

glVertex2f(birdsPos + 0.024f,0.8f + 0.05);

glVertex2f(birdsPos + 0.0f,0.83f + 0.05);

glEnd();

glLoadIdentity();

}

void tree()

{

glBegin(GL\_POLYGON);

glColor3ub(102, 51, 0);

glVertex2f(-0.72f,-0.15f);

glVertex2f(-0.65f,-0.2f);

glVertex2f(-0.735f,-0.17f);

glVertex2f(-0.74f,-0.25f);

glVertex2f(-0.775f,-0.17f);

glVertex2f(-0.85f,-0.2f);

glVertex2f(-0.78f,-0.15f);

//glVertex2f(-0.7f,-0.25f);

glEnd();

glBegin(GL\_QUADS);

glColor3ub(102, 51, 0);

glVertex2f(-0.78f,-0.15f);

glVertex2f(-0.78f,0.23f);

glVertex2f(-0.72f,0.23f);

glVertex2f(-0.72f,-0.15f);

glEnd();

glBegin(GL\_QUADS);

glColor3ub(102, 51, 0);

glVertex2f(-0.76f,0.23f);

glVertex2f(-0.76f,0.3f);

glVertex2f(-0.74f,0.3f);

glVertex2f(-0.74f,0.23f);

glEnd();

glBegin(GL\_QUADS);

glColor3ub(102, 51, 0);

glVertex2f(-0.74f,0.23f);

glVertex2f(-0.71f,0.29f);

glVertex2f(-0.7f,0.28f);

glVertex2f(-0.72f,0.23f);

glEnd();

glBegin(GL\_QUADS);

glColor3ub(102, 51, 0);

glVertex2f(-0.78f,0.23f);

glVertex2f(-0.8f,0.28f);

glVertex2f(-0.79f,0.29f);

glVertex2f(-0.76f,0.23f);

glEnd();

int i;

GLfloat x=-.75f; GLfloat y=.33f; GLfloat radius =.06f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* 3;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a=-.68f; GLfloat b=.31f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(a, b); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

a + (radius \* cos(i \* twicePi / triangleAmount)),

b + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c=-.81f; GLfloat d=.31f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

c + (radius \* cos(i \* twicePi / triangleAmount)),

d + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e=-.87f; GLfloat f=.35f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e + (radius \* cos(i \* twicePi / triangleAmount)),

f+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=-.61f; GLfloat h=.35f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

g + (radius \* cos(i \* twicePi / triangleAmount)),

h+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a1=-.61f; GLfloat b1=.4f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(a1, b1); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

a1 + (radius \* cos(i \* twicePi / triangleAmount)),

b1 + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c1=-.88f; GLfloat d1=.4f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

c1 + (radius \* cos(i \* twicePi / triangleAmount)),

d1 + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e1=-.87f; GLfloat f1=.44f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e1, f1); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e1 + (radius \* cos(i \* twicePi / triangleAmount)),

f1+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g1=-.61f; GLfloat h1=.4f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

g1 + (radius \* cos(i \* twicePi / triangleAmount)),

h1+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e11=-.64f; GLfloat f11=.44f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e11, f11); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e11 + (radius \* cos(i \* twicePi / triangleAmount)),

f11+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e12=-.75f; GLfloat f12=.44f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e12, f12); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e12 + (radius \* cos(i \* twicePi / triangleAmount)),

f12+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glBegin(GL\_QUADS);

glColor3ub(51, 204, 51);

glVertex2f(-0.85f,0.33f);

glVertex2f(-0.85f,0.44f);

glVertex2f(-0.65f,0.44f);

glVertex2f(-0.65f,0.33f);

glEnd();

GLfloat e123=-.8f; GLfloat f123=.5f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e123, f123); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e123 + (radius \* cos(i \* twicePi / triangleAmount)),

f123+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e1232=-.7f; GLfloat f1232=.5f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(51, 204, 51);

glVertex2f(e1232, f1232); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e1232 + (radius \* cos(i \* twicePi / triangleAmount)),

f1232+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

}

void boat1()

{

glBegin(GL\_POLYGON);

glColor3ub(0,0,0);

glVertex2f(boatPos + -0.2f, 0.4f);

glVertex2f(boatPos + -0.15f, 0.35f);

glVertex2f(boatPos + 0.15f, 0.35f);

glVertex2f(boatPos + 0.2f, 0.4f);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.93,0.7479,0.4929);

glVertex2f(boatPos + -0.13f, 0.4f);

glVertex2f(boatPos + -0.11f,0.48f);

glVertex2f(boatPos + -0.088f, 0.52f);

glVertex2f(boatPos + 0.13f, 0.52f);

glVertex2f(boatPos + 0.14f, 0.49f);

glVertex2f(boatPos + 0.15f, 0.4f);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.42, 0.2625, 0.042);

glVertex2f(boatPos + -0.038f, 0.57f + 0.05);

glVertex2f(boatPos + -0.038f, 0.73f + 0.05);

glVertex2f(boatPos + -0.035f, 0.75f + 0.05);

glVertex2f(boatPos + 0.064f, 0.73f + 0.05);

glVertex2f(boatPos + 0.065f, 0.71f + 0.05);

glVertex2f(boatPos + 0.065f, 0.55f + 0.05);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(136,204,0);

glVertex2f(boatPos + 0.0f, 0.52f);

glVertex2f(boatPos + 0.0f, 0.79f);

glVertex2f(boatPos + 0.01f, 0.79f);

glVertex2f(boatPos + 0.01f, 0.52f);

glEnd();

}

void boat2()

{

glBegin(GL\_POLYGON);

glColor3ub(0,0,0);

glVertex2f(boat2Pos + -0.2f, 0.4f);

glVertex2f(boat2Pos + -0.15f, 0.35f);

glVertex2f(boat2Pos + 0.15f, 0.35f);

glVertex2f(boat2Pos + 0.2f, 0.4f);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.43,0.2718,0.0172);

glVertex2f(boat2Pos + -0.13f, 0.4f);

glVertex2f(boat2Pos + -0.11f,0.48f);

glVertex2f(boat2Pos + -0.088f, 0.52f);

glVertex2f(boat2Pos + 0.13f, 0.52f);

glVertex2f(boat2Pos + 0.14f, 0.49f);

glVertex2f(boat2Pos + 0.15f, 0.4f);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.1218, 0.29, 0.2395);

glVertex2f(boat2Pos + -0.038f, 0.57f + 0.05);

glVertex2f(boat2Pos + -0.038f, 0.73f + 0.05);

glVertex2f(boat2Pos + -0.035f, 0.75f + 0.05);

glVertex2f(boat2Pos + 0.064f, 0.73f + 0.05);

glVertex2f(boat2Pos + 0.065f, 0.71f + 0.05);

glVertex2f(boat2Pos + 0.065f, 0.55f + 0.05);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(0.237,0.2685,0.3);

glVertex2f(boat2Pos + 0.0f, 0.52f);

glVertex2f(boat2Pos + 0.0f, 0.79f);

glVertex2f(boat2Pos + 0.01f, 0.79f);

glVertex2f(boat2Pos + 0.01f, 0.52f);

glEnd();

}

void line()

{

glBegin(GL\_LINES);

glLineWidth(0.05);

glColor3f(0.0,0.0,0.0);

glVertex2f(-1.0,-0.33);

glVertex2f(1.0,-0.33);

glEnd();

}

void timer(int)

{

glutPostRedisplay();

glutTimerFunc(1000/60.0, timer, 0);

if(cloud1State == 0)

{

if(cloud1\_pos < 0.8)

{

cloud1\_pos+=.0015;

}

else

cloud1State = 1;

}

if(cloud1State == 1)

{

if(cloud1\_pos > -1.5)

{

cloud1\_pos-=.0015;

}

else

cloud1State = 0;

}

if(cloud2State == 0)

{

if(cloud2\_pos < 0.8)

{

cloud2\_pos+=.0025;

}

else

cloud2State = 1;

}

if(cloud2State == 1)

{

if(cloud2\_pos > -1.5)

{

cloud2\_pos-=.0025;

}

else

cloud2State = 0;

}

if(birdsPos < 1.5)

{

birdsPos+=0.002;

}

else

birdsPos = -1.8;

if(birds2Pos < 1.5)

{

birds2Pos+=0.005;

}

else

birds2Pos = -1.6;

if(boatPos > -1.9)

{

boatPos-=0.0025;

}

else

boatPos = 1.8;

if(boat2Pos < 3.1)

{

boat2Pos+=0.0015;

}

else

boat2Pos = -3.1;

}

void display()

{

//Clear display to initial color

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

sea();

land();

line();

glTranslatef(0,0.2,0.0);

sun();

glTranslatef(-.8,0.0,0.0);

grass();

glLoadIdentity();

glTranslatef(-.7,-0.20,0.0);

grass();

glLoadIdentity();

cloud1();

glLoadIdentity();

cloud2();

glLoadIdentity();

//glTranslatef(0.0,-0.30,0.0);

glScaled(0.75,0.75,0.75);

bird();

glLoadIdentity();

glTranslated(0.0,-0.2,0);

glScaled(-1,1,1);

bird2();

glLoadIdentity();

glScaled(0.35,0.35,0.35);

glTranslated(0,0.5,0);

boat2();

glLoadIdentity();

glScaled(0.65,0.65,0.65);

glTranslated(0,-0.15,0);

boat1();

glLoadIdentity();

glScaled(1.5,1.5,1.5);

glTranslated(1.2,-0.1,0.0);

tree();

glLoadIdentity();

glLoadIdentity();

//Drawing the shape

glutSwapBuffers();

}

void init()

{

//Setting initial color

glClearColor(0.0096,0.5165,0.96,1.0);

//Change the matrix mode

glMatrixMode(GL\_PROJECTION);

//Refresh screen

glLoadIdentity();

//Set the coordinate scaling

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

}

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

{

//Initialize project

glutInit(&argc, argv);

//Set display color mode and buffer size

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE);

//Set display window size

glutInitWindowSize(1500, 700);

//Set display window position

glutInitWindowPosition(250, 300);

//Create window named as parameter

glutCreateWindow("W1");

//Actual view of window

glutDisplayFunc(display);

//Timer to call display repeatedly

glutTimerFunc(0,timer,0);

//Enable settings at init function

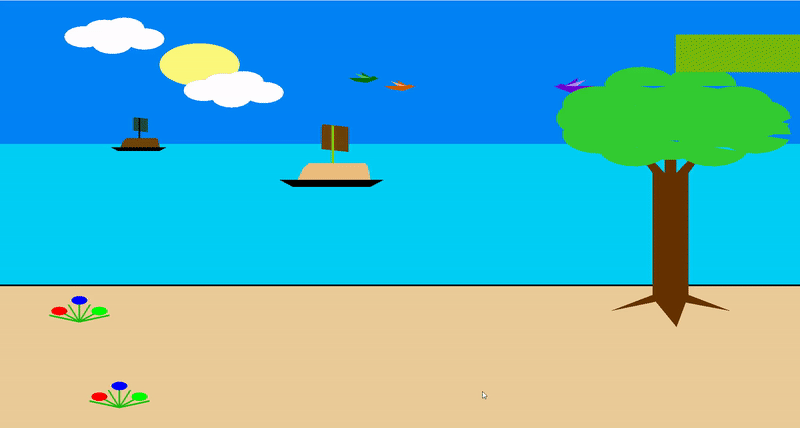
init();

//Loop the main function

glutMainLoop();

}

**Output**

****