COMPUTER GRAPHICS LAB

FINAL PROJECT

TEAM MEMBERS

MEMBER-1: MD. ABDUL KHALEK KAKON

ID: 201-15-3124 (55_PC-D2)

MEMBER-2: MD. MINHAJ UDDIN MIA ASHEQ

ID: 201-15-3657 (55_PC-D1)

MEMBER-3: SAJEEB DATTA

ID: 201-15-3119 (55_PC-D2)

MEMBER-4: SUMIYA MAHMUD PROME

ID: 201-15-3361 (55_PC-D1)

MEMBER-5: MD. MOSTAFIZUR RAHMAN AKHAND

ID: 201-15-3683 (55_PC-D2)

CODE:

```
#include <windows.h>
#include <mmsystem.h>
#include <GL/glut.h>
#include <stdio.h>
#include <GL/gl.h>
#include <math.h>

//for ship
float xPos = 0.0f;
float yPos = 0.0f;
float bright = 90.0;
float bleft = -90.0;
```

```
float xPoss = 0.0f;
float yPoss = 0.0f;

//for helicopter
float xHeli = 0.0f;
float yHeli = 0.0f;

//for sun
float xSun = 0.0f;
float ySun = 0.0f;
//for moon
float xMoon = 0.0f;
float yMoon = 0.0f;

//for circle
float x,y,i;
float PI=3.1416;
int triangleAmount =50;
GLfloat radius;
GLfloat twicePi = 2.0 * PI;

//for rotetion
float angle = 0.0f;

float a=0.7f, b=0.7f, c=1.0f;



void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
```

```
glLoadIdentity();
//sky
glBegin(GL_QUADS);
glColor3f(a, b, c);
glVertex2f(0, 0);
glVertex2f(100, 0);
glVertex2f(100, 100);
glVertex2f(0, 100);
glEnd();

//sun
glLoadIdentity();
glTranslatef(xSun, ySun, 0.0f);
glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 0.7, 0);
x=40;
y=90;
radius =5;
twicePi = 2.0 * PI;
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();

//megh
```

```
glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 1, 1);
x=15;
y=87;
radius =4;
twicePi = 2.0 * PI;
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();



   glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 1, 1);
x=19;
y=87;
radius =4;
twicePi = 2.0 * PI;
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();
glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 1, 1);
x=23;
y=86;
radius =4;
twicePi = 2.0 * PI;
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();

//moon
glLoadIdentity();
glTranslatef(xMoon, yMoon, 0.0f);
glBegin(GL_TRIANGLE_FAN);
glColor3d(0.9, 0.9, 0.9);
x=70;
y=60;
radius =5;
twicePi = 2.0 * PI;
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();

//sea
glLoadIdentity();
glBegin(GL_QUADS);
glColor3f(0.0f, 0.0f, 1.0f);
glVertex2f(0, 0);
glVertex2f(100, 0);
glVertex2f(100, 80);
glVertex2f(0, 80);
glEnd();
//water

//glPushMatrix();
//glTranslatef(xPoss, 0.0f, 0.0f);
glBegin(GL_LINES);
glColor3f(1.0f,1.0f,1.0f);
glVertex2f(5,5);
glVertex2f(10,5);
glVertex2f(15,5);
glVertex2f(20,5);
glVertex2f(25,5);
glVertex2f(30,5);
glVertex2f(35,5);
glVertex2f(40,5);
glVertex2f(45,5);
glVertex2f(50,5);
```

```
glVertex2f(55,5);
glVertex2f(60,5);
glVertex2f(65,5);
glVertex2f(70,5);
glVertex2f(75,5);
glVertex2f(80,5);
glVertex2f(85,5);
glVertex2f(90,5);
glVertex2f(95,5);
glVertex2f(100,5);
glVertex2f(0,10);
glVertex2f(5,10);
glVertex2f(10,10);
glVertex2f(15,10);
glVertex2f(20,10);
glVertex2f(25,10);
glVertex2f(30,10);
glVertex2f(35,10);
glVertex2f(40,10);
glVertex2f(45,10);
glVertex2f(50,10);
glVertex2f(55,10);
glVertex2f(60,10);
glVertex2f(65,10);
glVertex2f(70,10);
glVertex2f(75,10);
glVertex2f(80,10);
glVertex2f(85,10);
glVertex2f(90,10);
glVertex2f(95,10);
glVertex2f(5,15);
```

```
glVertex2f(10,15);
glVertex2f(15,15);
glVertex2f(20,15);
glVertex2f(25,15);
glVertex2f(30,15);
glVertex2f(35,15);
glVertex2f(40,15);
glVertex2f(45,15);
glVertex2f(50,15);
glVertex2f(55,15);
glVertex2f(60,15);
glVertex2f(65,15);
glVertex2f(70,15);
glVertex2f(75,15);
glVertex2f(80,15);
glVertex2f(85,15);
glVertex2f(90,15);
glVertex2f(95,15);
glVertex2f(100,15);
glVertex2f(0,20);
glVertex2f(5,20);
glVertex2f(10,20);
glVertex2f(15,20);
glVertex2f(20,20);
glVertex2f(25,20);
glVertex2f(30,20);
glVertex2f(35,20);
glVertex2f(40,20);
glVertex2f(45,20);
glVertex2f(50,20);
glVertex2f(55,20);
```

```
glVertex2f(60,20);
glVertex2f(65,20);
glVertex2f(70,20);
glVertex2f(75,20);
glVertex2f(80,20);
glVertex2f(85,20);
glVertex2f(90,20);
glVertex2f(95,20);
glVertex2f(5,25);
glVertex2f(10,25);
glVertex2f(15,25);
glVertex2f(20,25);
glVertex2f(25,25);
glVertex2f(30,25);
glVertex2f(35,25);
glVertex2f(40,25);
glVertex2f(45,25);
glVertex2f(50,25);
glVertex2f(55,25);
glVertex2f(60,25);
glVertex2f(65,25);
glVertex2f(70,25);
glVertex2f(75,25);
glVertex2f(80,25);
glVertex2f(85,25);
glVertex2f(90,25);
glVertex2f(95,25);
glVertex2f(100,25);
glVertex2f(0,30);
glVertex2f(5,30);
glVertex2f(10,30);
```

```
glVertex2f(15,30);
glVertex2f(20,30);
glVertex2f(25,30);
glVertex2f(30,30);
glVertex2f(35,30);
glVertex2f(40,30);
glVertex2f(45,30);
glVertex2f(50,30);
glVertex2f(55,30);
glVertex2f(60,30);
glVertex2f(65,30);
glVertex2f(70,30);
glVertex2f(75,30);
glVertex2f(80,30);
glVertex2f(85,30);
glVertex2f(90,30);
glVertex2f(95,30);
glVertex2f(5,35);
glVertex2f(10,35);
glVertex2f(15,35);
glVertex2f(20,35);
glVertex2f(25,35);
glVertex2f(30,35);
glVertex2f(35,35);
glVertex2f(40,35);
glVertex2f(45,35);
glVertex2f(50,35);
glVertex2f(55,35);
glVertex2f(60,35);
glVertex2f(65,35);
glVertex2f(70,35);
```

```
glVertex2f(75,35);
glVertex2f(80,35);
glVertex2f(85,35);
glVertex2f(90,35);
glVertex2f(95,35);
glVertex2f(100,35);
glVertex2f(0,40);
glVertex2f(5,40);
glVertex2f(10,40);
glVertex2f(15,40);
glVertex2f(20,40);
glVertex2f(25,40);
glVertex2f(30,40);
glVertex2f(35,40);
glVertex2f(40,40);
glVertex2f(45,40);
glVertex2f(50,40);
glVertex2f(55,40);
glVertex2f(60,40);
glVertex2f(65,40);
glVertex2f(70,40);
glVertex2f(75,40);
glVertex2f(80,40);
glVertex2f(85,40);
glVertex2f(90,40);
glVertex2f(95,40);
glVertex2f(5,45);
glVertex2f(10,45);
glVertex2f(15,45);
glVertex2f(20,45);
glVertex2f(25,45);
```

```
glVertex2f(30,45);
glVertex2f(35,45);
glVertex2f(40,45);
glVertex2f(45,45);
glVertex2f(50,45);
glVertex2f(55,45);
glVertex2f(60,45);
glVertex2f(65,45);
glVertex2f(70,45);
glVertex2f(75,45);
glVertex2f(80,45);
glVertex2f(85,45);
glVertex2f(90,45);
glVertex2f(95,45);
glVertex2f(100,45);
glVertex2f(0,50);
glVertex2f(5,50);
glVertex2f(10,50);
glVertex2f(15,50);
glVertex2f(20,50);
glVertex2f(25,50);
glVertex2f(30,50);
glVertex2f(35,50);
glVertex2f(40,50);
glVertex2f(45,50);
glVertex2f(50,50);
glVertex2f(55,50);
glVertex2f(60,50);
glVertex2f(65,50);
glVertex2f(70,50);
glVertex2f(75,50);
```

```
glVertex2f(80,50);
glVertex2f(85,50);
glVertex2f(90,50);
glVertex2f(95,50);
glVertex2f(5,55);
glVertex2f(10,55);
glVertex2f(15,55);
glVertex2f(20,55);
glVertex2f(25,55);
glVertex2f(30,55);
glVertex2f(35,55);
glVertex2f(40,55);
glVertex2f(45,55);
glVertex2f(50,55);
glVertex2f(55,55);
glVertex2f(60,55);
glVertex2f(65,55);
glVertex2f(70,55);
glVertex2f(75,55);
glVertex2f(80,55);
glVertex2f(85,55);
glVertex2f(90,55);
glVertex2f(95,55);
glVertex2f(100,55);
glVertex2f(0,60);
glVertex2f(5,60);
glVertex2f(10,60);
glVertex2f(15,60);
glVertex2f(20,60);
glVertex2f(25,60);
glVertex2f(30,60);
```

```
glVertex2f(35,60);
glVertex2f(40,60);
glVertex2f(45,60);
glVertex2f(50,60);
glVertex2f(55,60);
glVertex2f(60,60);
glVertex2f(65,60);
glVertex2f(70,60);
glVertex2f(75,60);
glVertex2f(80,60);
glVertex2f(85,60);
glVertex2f(90,60);
glVertex2f(95,60);
glVertex2f(5,65);
glVertex2f(10,65);
glVertex2f(15,65);
glVertex2f(20,65);
glVertex2f(25,65);
glVertex2f(30,65);
glVertex2f(35,65);
glVertex2f(40,65);
glVertex2f(45,65);
glVertex2f(50,65);
glVertex2f(55,65);
glVertex2f(60,65);
glVertex2f(65,65);
glVertex2f(70,65);
glVertex2f(75,65);
glVertex2f(80,65);
glVertex2f(85,65);
glVertex2f(90,65);
```

```
glVertex2f(95,65);
glVertex2f(100,65);
glVertex2f(0,70);
glVertex2f(5,70);
glVertex2f(10,70);
glVertex2f(15,70);
glVertex2f(20,70);
glVertex2f(25,70);
glVertex2f(30,70);
glVertex2f(35,70);
glVertex2f(40,70);
glVertex2f(45,70);
glVertex2f(50,70);
glVertex2f(55,70);
glVertex2f(60,70);
glVertex2f(65,70);
glVertex2f(70,70);
glVertex2f(75,70);
glVertex2f(80,70);
glVertex2f(85,70);
glVertex2f(90,70);
glVertex2f(95,70);
glVertex2f(5,75);
glVertex2f(10,75);
glVertex2f(15,75);
glVertex2f(20,75);
glVertex2f(25,75);
glVertex2f(30,75);
glVertex2f(35,75);
glVertex2f(40,75);
glVertex2f(45,75);
```

```
glVertex2f(50,75);
glVertex2f(55,75);
glVertex2f(60,75);
glVertex2f(65,75);
glVertex2f(70,75);
glVertex2f(75,75);
glVertex2f(80,75);
glVertex2f(85,75);
glVertex2f(90,75);
glVertex2f(95,75);
glVertex2f(100,75);

glEnd();
//glPopMatrix();

glLoadIdentity();
glTranslatef(xPos, yPos, 0.0f);
//bot body 1
glBegin(GL_QUADS);
glColor3f(0.5f, 0.5f, 0.5f);
glVertex2f(40, 25);
glVertex2f(50, 15);
glVertex2f(50, 20);
glVertex2f(40, 30);
glEnd();
//bot body 2
glBegin(GL_QUADS);
glColor3f(0.6f, 0.6f, 0.6f);
glVertex2f(50, 15);
glVertex2f(90, 15);
glVertex2f(90, 20);
```

```
glVertex2f(50, 20);
glEnd();
//bot body 3
glBegin(GL_POLYGON);
glColor3f(0.0f, 0.5f, 0.0f);
glVertex2f(50, 20);
glVertex2f(80, 20);
glVertex2f(80, 40);
glVertex2f(50, 40);
glVertex2f(40, 30);
glEnd();
//helipad big circle
glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 1, 0);
x=60;
y=29;
radius =8;
twicePi = 2.0 * PI;
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();
//helipad small circle
glBegin(GL_TRIANGLE_FAN);
glColor3d(0, 1, 0);
x=60;
```

```
y=29;
radius =7;
twicePi = 2.0 * PI;
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();
//H
glBegin(GL_QUADS);
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(57, 26);
glVertex2f(63, 26);
glVertex2f(63, 27);
glVertex2f(57, 27);
glEnd();
//H
glBegin(GL_QUADS);
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(57, 31);
glVertex2f(63, 31);
glVertex2f(63, 32);
glVertex2f(57, 32);
glEnd();
//H
glBegin(GL_QUADS);
glColor3f(1.0f, 1.0f, 1.0f);
```

```
glVertex2f(59.5, 27);
glVertex2f(60.5, 27);
glVertex2f(60.5, 31);
glVertex2f(59.5, 31);
glEnd();
//bot body 4
glBegin(GL_QUADS);
glColor3f(0.9f, 0.9f, 0.9f);
glVertex2f(41, 29);
glVertex2f(50, 39);
glVertex2f(50, 40);
glVertex2f(40, 30);
glEnd();
//bot body 5
glBegin(GL_QUADS);
glColor3f(0.9f, 0.9f, 0.9f);
glVertex2f(50, 39);
glVertex2f(80, 39);
glVertex2f(80, 40);
glVertex2f(50, 40);
glEnd();
//bot bulding 1
glBegin(GL_QUADS);
glColor3f(0.95f, 0.95f, 0.95f);
glVertex2f(80, 20);
glVertex2f(90, 20);
glVertex2f(92, 25);
glVertex2f(82, 25);
glEnd();
//bot bulding 2
glBegin(GL_QUADS);
```

```
glColor3f(0.0f, 1.0f, 0.0f);
glVertex2f(82, 25);
glVertex2f(92, 25);
glVertex2f(92, 45);
glVertex2f(82, 45);
glEnd();
//potaka circle
glBegin(GL_TRIANGLE_FAN);
glColor3d(1, 0, 0);
x=87;
y=35;
radius =3;
twicePi = 2.0 * PI;
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();

//bot bulding 3
glBegin(GL_QUADS);
glColor3f(0.9f, 0.9f, 0.9f);
glVertex2f(80, 20);
glVertex2f(82, 25);
glVertex2f(82, 45);
glVertex2f(80, 40);
glEnd();
```

```
glTranslatef(xHeli, yHeli, 0.0f);
//helicopter
glBegin(GL_QUADS);
glColor3f(0.0, 0.0, 0.0);
glVertex3f(57.0f, 28.0f, 0.0f);
glVertex3f(63.0f, 28.0f, 0.0f);
glVertex3f(63.0f, 29.0f, 0.0f);
glVertex3f(57.0f, 29.0f, 0.0f);
glEnd();

glBegin(GL_QUADS);
glColor3f(0.0, 0.0, 0.0);
glVertex3f(55.0f, 30.0f, 0.0f);
glVertex3f(57.0f, 28.0f, 0.0f);
glVertex3f(57.0f, 29.0f, 0.0f);
glVertex3f(56.0f, 30.0f, 0.0f);
glEnd();

glBegin(GL_QUADS);
glColor3f(0.0, 0.0, 0.0);
glVertex3f(63.0f, 28.0f, 0.0f);
glVertex3f(65.0f, 30.0f, 0.0f);
glVertex3f(64.0f, 30.0f, 0.0f);
glVertex3f(63.0f, 29.0f, 0.0f);
glEnd();
```

```
    glBegin(GL_LINES);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(57.0f, 29.0f, 0.0f);
    glVertex3f(57.0f, 31.0f, 0.0f);
    glEnd();

    glBegin(GL_LINES);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(58.0f, 29.0f, 0.0f);
    glVertex3f(58.0f, 31.0f, 0.0f);
    glEnd();

    glBegin(GL_LINES);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(62.0f, 29.0f, 0.0f);
    glVertex3f(62.0f, 31.0f, 0.0f);
    glEnd();

    glBegin(GL_LINES);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(63.0f, 29.0f, 0.0f);
    glVertex3f(63.0f, 31.0f, 0.0f);
    glEnd();

//h-head
    glBegin(GL_POLYGON);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(54.0f, 31.0f, 0.0f);
    glVertex3f(65.0f, 31.0f, 0.0f);
    glVertex3f(65.0f, 37.0f, 0.0f);
    glVertex3f(64.5f, 37.5f, 0.0f);
```

```
    glVertex3f(64.0f, 38.0f, 0.0f);
    glVertex3f(54.0f, 38.0f, 0.0f);
    glVertex3f(52.0f, 37.0f, 0.0f);
    glVertex3f(52.0f, 36.0f, 0.0f);
    glVertex3f(52.0f, 33.0f, 0.0f);
    glVertex3f(52.5f, 31.5f, 0.0f);
    glEnd();

//1st-door
    glBegin(GL_POLYGON);
    glColor3f (0.0, 1.0, 1.0);
    glVertex3f(52.0f, 33.0f, 0.0f);
    glVertex3f(55.0f, 33.0f, 0.0f);
    glVertex3f(55.0f, 35.0f, 0.0f);
    glVertex3f(54.0f, 36.0f, 0.0f);
    glVertex3f(52.0f, 36.0f, 0.0f);
    glEnd();

//2nd-door
    glBegin(GL_QUADS);
    glColor3f(0.0, 1.0, 1.0);
    glVertex3f(56.0f, 33.0f, 0.0f);
    glVertex3f(59.0f, 33.0f, 0.0f);
    glVertex3f(59.0f, 36.0f, 0.0f);
    glVertex3f(56.0f, 36.0f, 0.0f);
    glEnd();

//3nd-door
    glBegin(GL_QUADS);
    glColor3f(0.0, 1.0, 1.0);
    glVertex3f(60.0f, 33.0f, 0.0f);
```

```
    glVertex3f(64.0f, 33.0f, 0.0f);
    glVertex3f(64.0f, 36.0f, 0.0f);
    glVertex3f(60.0f, 36.0f, 0.0f);
    glEnd();

//h-tail
    glBegin(GL_QUADS);
    glColor3f (0.0, 1.0, 1.0);
    glVertex3f(65.0f, 33.0f, 0.0f);
    glVertex3f(75.0f, 33.0f, 0.0f);
    glVertex3f(75.0f, 35.0f, 0.0f);
    glVertex3f(65.0f, 35.0f, 0.0f);
    glEnd();

    glBegin(GL_TRIANGLES);
    glColor3f (0.0, 1.0, 1.0);
    glVertex3f(72.0f, 35.0f, 0.0f);
    glVertex3f(75.0f, 35.0f, 0.0f);
    glVertex3f(75.0f, 38.0f, 0.0f);
    glEnd();//finish_h-tail

//dot
    glBegin(GL_QUADS);
    glColor3f (0.0, 0.0, 0.0);
    glVertex3f(73.0f, 34.0f, 0.0f);
    glVertex3f(74.0f, 34.0f, 0.0f);
    glVertex3f(74.0f, 35.0f, 0.0f);
    glVertex3f(73.0f, 35.0f, 0.0f);
    glEnd();
```

```
//h-body-pakha
//2nd
   glBegin(GL_QUADS);
   glColor3f (0.0, 0.0, 0.0);
   glVertex3f(59.0f, 38.0f, 0.0f);
   glVertex3f(60.0f, 38.0f, 0.0f);
   glVertex3f(60.0f, 40.0f, 0.0f);
   glVertex3f(59.0f, 40.0f, 0.0f);
   glEnd();



   //big blate
   glTranslatef(59.5, 40, 0.0);
   glRotatef(angle, 0.0f, 0.0f, 1.0f);
   glBegin(GL_QUADS);
   glColor3f(1.0, 0.0, 0.0);
   glVertex2f(-0.5, -10.0);
   glVertex2f(0.5, -10.0);
   glVertex2f(0.5, 10.0);
   glVertex2f(-0.5, 10.0);
   glEnd();

   //small blate

   glLoadIdentity();
   glTranslatef(xPos, yPos, 0.0f);
   glTranslatef(xHeli, yHeli, 0.0f);
   glTranslatef(73.5, 34.5, 0.0);
   glRotatef(angle, 0.0f, 0.0f, 1.0f);
```

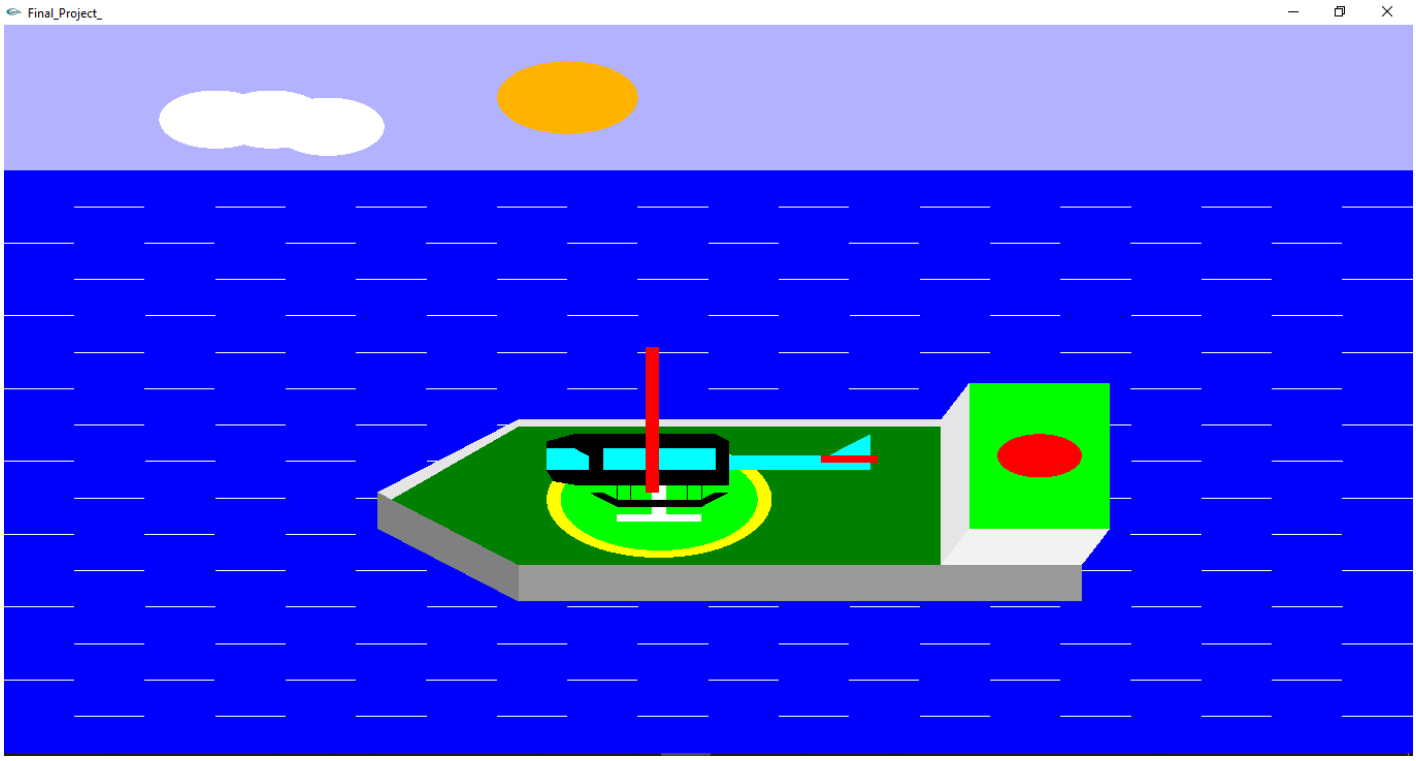
```
    glBegin(GL_QUADS);
    glColor3f(1.0, 0.0, 0.0);
    glVertex2f(-2.0, -0.5);
    glVertex2f(2.0, -0.5);
    glVertex2f(2.0, 0.5);
    glVertex2f(-2.0, 0.5);
    glEnd();

    glutSwapBuffers();
}

void update(int value)
{
    angle+=1000.0f;
    if(angle>360)
    angle-=360;
    glutPostRedisplay();
    glutTimerFunc(20, update, 0);
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
    case 'a':
        xPos -= 0.1f;
        break;
    case 'd':
        xPos += 0.1f;
        break;
    case 'w':
```

```
    yPos += 0.1f;
    break;
case 's':
    yPos -= 0.1f;
    break;
case 'h':
    xHeli -= 2.0f;
    break;
case 'k':
    xHeli += 1.1f;
    break;
case 'u':
    yHeli += 0.8f;
    break;
case 'j':
    yHeli -= 0.1f;
    break;
case 'o':
    yMoon -= 0.1f;
    ySun += 0.1f;
    break;
case 'l':
    yMoon += 0.1f;
    ySun -= 0.1f;
    break;

case 'c':
    a=0.5f, b=0.5f, c=1.0f;
    break;
case 'v':
    a=0.7f, b=0.7f, c=1.0f;
```

```
        break;
    default:
        break;
    }



    if(xPos>bright)
    {
        xPos=bleft;
    }
    else if(xPos<bleft)
    {
        xPos=bright;
    }
    else if(xHeli>bright)
    {
        xHeli=bleft;
    }
    else if(xHeli<bleft)
    {
        xHeli=bright;
    }



    glutPostRedisplay();
}



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

```
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(700, 500);
    glutCreateWindow("Final_Project_");
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  // White background
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 100.0, 0.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutTimerFunc(25, update, 0);
    sndPlaySound("project song.wav",SND_ASYNC);
    glutMainLoop();
    return 0;
}
```

OUTPUT:


Final_Project_