THE HIDING BOAT -TITANIC

CONTENTS

- **Objects**
  - **Functions**
  - **Algorithms**
- **Update Redisplay**

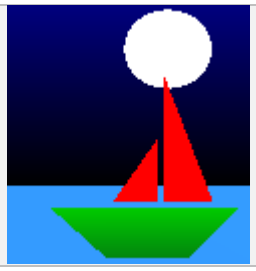
FILES LIST:

MAIN.CPP
LIBRARY_OBJECTS.CPP

Objects

The objects used in this project:

OBJECT NAME	DESCRIPTION(IMAGE)	PARAMETERS	FUNCION USES FROM GLUT(OPENGL)
NIGHT SKY		**Night(int)** --> if the value is 0 the color changes into morning theme. For 1 the color stays the same.	glColor3f(), glBegin(GL_TRIANGLE), glBegin(GL_QUADS), glVertex2i()
MOON		**Moon** -> for 0 mood hides for 35 moon reveals	Custom library: --- use circle function that creates a circle using small triangle and filling it with white color by passing rgb parameter. OPENGL LIBRARY: ---used glBegin(GL_TRIANGLE) TO CREATE SMALL TRIANGLES FROM THE CENTER AS ONE COMMON POINT.
WINDMILL		**Windmill:** Passed theta value to change the position of the triangles. Passed coordinate value to change the position of the windmill object. Passed (x+translation) to change the other 2 windmill coordinate in xy plane.	
SEA		**NO PAREMETER PASSED FOR THIS OBJECT.**	**JUST A SIMPLE GL_QUADS WITH 4 CORDINATES**

SMALLBOAT		**PASSED INCREMENTTED VALUE TO ALL THE X CORDINATE TO MOVE THE BOAT FROM LEFT TO RIGHT. SAME AS THE BIG BOAT**	**USED GL_QUADS AND GL_TRIANGLES FOR CREATING THE BOAT AND glColof3f(); for the parotic color.**
BIGBOAT (TITANIC)		**As same as the small boat passed x incremented value in the boat to move it in the x direction left->right of right->left**	Custom library: --- use circle function that creates a circle using small triangle and filling it with white color by passing rgb parameter. **Opengl library:** **Used to create glbegin->GL_QUADS TO CREATE THE SQUEARE CHIMNIS AND OTHER SQUARE SHAPES. NOT TRIANGLE LIBRARY HABE BEEN USED HERE.**

ALGORITHMS:

The algorithms use in the functions:

Circle, Rotatearoundpt functions are used to draw variable shaped circles using circle drawing algorithms.

```
void circle(int x, int y, int r, GLfloat color[]){
    //CREATING CIRCLE USING line loop mehod explicit curve
    //x+=12;y+=30;
    if(r==0)return;
    int circle_points = 100;
    GLfloat toPi = 2 * 3.1416,cp_x,cp_y;

    glBegin(GL_TRIANGLE_FAN);
        glColor3f(color[0],color[1],color[2]);
        for(int i = 0; i <= circle_points;i++){
            cp_x = x + (r * cos(i *  toPi / circle_points));
            cp_y = y + (r * sin(i * toPi / circle_points));
            glVertex2f(cp_x,cp_y);
        }
    glEnd();
}
void rotateAroundPt(int px,int py,int r,float point){
    int circle_points = 100;
    GLfloat toPi = 2 * 3.1416;
    int x =px + (r * cos(point *  toPi / circle_points));
    int y =py + (r * sin(point * toPi / circle_points));
    glVertex2i(x,y);
}
```

The mechanics for movement in the x direction for boats(all):

```
void update(int value) {
    //add the code here
    moon = 35;
    daystate=1;
    if(movetitanic>=880){
        moveboat+=2;
        moveboat1+=4;
        daystate=0;
        point+=2;
        moon=0;
    }else if (pause==0){
        movetitanic+=movetitanicspd+6;
    }
    moveboat+=4;
    moveboat1+=6;
    if(moveboat>=220 && moveboat1>=440){
        moveboat=-1100;
        moveboat1=-1000;
    }
    point+=0.05;
    //add the code here

    glutPostRedisplay(); ////Tell GLUT that the scene has changed
    //glutReshapeWindow(1366,768);
    glutTimerFunc(50, update, 0);
}
```

The full scene from a single moment: