

```
#include<windows.h>

#include <GL/glut.h>

float xPos = 0.0f;
float yPos = 0.0f;

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
//River

    glBegin(GL_QUADS);
    glColor3f(0.0, 0.0, 1.0);
    glVertex3f(-1.0, -0.35, 0.00);
    glVertex3f(1.0, -0.35, 0.00);
    glVertex3f(1.0, -1.0, 0.00);
    glVertex3f(-1.0, -1.0, 0.00);
    glEnd();
//boat

    glTranslatef(xPos, yPos, 0.0f);
    glBegin(GL_QUADS);
    glColor3f(0.90, 0.20, 0.20);
    glVertex3f(-0.50, -0.27, 0.00);
    glVertex3f(0.27, -0.27, 0.00);
    glVertex3f(0.12, -0.43, 0.00);
    glVertex3f(-0.50, -0.43, 0.00);
    glEnd();
//boat phal

    glBegin(GL_TRIANGLES);
```

```

    glColor3f(1.0, 1.0, 0.80);
    glVertex3f(-0.05, 0.60, 0.00);
    glColor3f(1.0, 0.80, 0.80);
    glVertex3f(-0.05, -0.21, 0.00);
    glColor3f(1.0, 1.0, 1.0);
    glVertex3f(-0.60, -0.21, 0.00);
    glEnd();

    glBegin(GL_TRIANGLES);
    glColor3f(1.0, 1.0, 0.8);
    glVertex3f(0.0, 0.40, 0.00);
    glColor3f(1.0, 1.0, 1.0);
    glVertex3f(0.35, -0.21, 0.00);
    glColor3f(1.0, 0.80, 0.80);
    glVertex3f(0.0, -0.21, 0.00);
    glEnd();

    glutSwapBuffers();
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'w':
            yPos += 0.0f;
            break;
        case 's':
            yPos -= 0.0f;
            break;
    }
}

```

```

case 'a':
    xPos -= 0.0f;
    break;
case 'd':
    xPos += 0.1f;
    break;
default:
    break;
}

glutPostRedisplay();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutCreateWindow("2D_MOvement");
    glClearColor(0.8f, 0.8f, 1.0f, 1.0f); // White background
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-1.0f, 1.0f, -1.0f, 1.0f);
    glMatrixMode(GL_MODELVIEW);
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

Output:

