AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF CSE

COMPUTER GRAPHICS

Spring 2023-2024

Section: G

PROJECT REPORT ON

***Titanic Sinking Simulation using OpenGL*.**

Supervised By

MAHFUJUR RAHMAN

Submitted By

Name	ID
1. MD. SHARIFUL AMAN	21-45890-1
2. MD. ASHFAQ AYUB	22-46369-1
3. FARIA AHMED RICHI	22-46371-1
5. KAZI ABDULLAH JARIF	22-46386-1
5. EISHITA RANI ACHARJEE	22-46397-1

Date of Submission: **MAY 14, 2024**

Introduction

The inception of a computer graphics project centered around simulating the sinking of the Titanic using OpenGL in C++ is currently underway. This endeavor seeks to digitally recreate the event through advanced graphics techniques. By leveraging OpenGL, the goal is to develop a visually immersive simulation that accurately portrays the sinking of the RMS Titanic. Emphasis is placed on employing sophisticated algorithms and rendering methods to replicate the intricate details of the ship's descent into the ocean. Attention is directed towards ensuring accuracy while delivering an engaging and informative visual experience. The utilization of C++ alongside OpenGL provides a robust platform for crafting an interactive and visually captivating simulation. Through this project, the aim is to engage audiences and showcase the technical prowess of computer graphics in simulating complex real-world events.

Problem Statement

The objective of this project is to develop a real-time simulation of the Titanic sinking using OpenGL. The simulation should accurately depict the sequence of events, including the collision with the iceberg, the flooding of compartments, the listing and sinking of the ship, and the subsequent rescue efforts.

Key components and Features:

1. **Main Key Feature & Function**

   • Ship • Tree • Sky • Cloud • Water • Waterlines • Sound • Ship1 • Ship2 • Tree • Building • Windmill • Ship3 • Collision with Ices • Scenerio-Over

2. **Function:**

   void update(int value);

   void SpecialInput(int key, int x, int y);

   void display();

   void display1();

   void display2();

   void display3();

```
void display4();
void display5();
void glFlush();
void ship();
void water1()
void sky1();
void cloud();
void sun();
void sun2();
void rain();
void moon();
void star();
void waterlines();
void windMill();
void trees();
void birds();
void miniices();
void miniices2();
void sound();
void sound2();
void ice();
```

I. **Ship Function:**

```
void ship()
{
  // Placeholder for drawing the ship
  //Body
  glTranslated(0,110,0);
  glScaled(28,23,0);

  glBegin(GL_POLYGON);
  glColor3ub(16,58,65);
  glVertex2f(1,5.5);
  glColor3ub(0,49,57);
  glVertex2f(3,1);
  glColor3ub(0,49,57);
  glVertex2f(19,1);
  glColor3ub(16,58,65);
```

```
glVertex2f(21,5.5);
glEnd();

glColor3f(0.5,0.5,0.5);
glBegin(GL_POLYGON);
glVertex2f(3.5,5.5);
glVertex2f(3.5,7);
glVertex2f(19.5,7);
glVertex2f(19.5,5.5);
glEnd();

glColor3f(0.5,0.5,0.5);
glBegin(GL_POLYGON);
glVertex2f(4.5,8);
glVertex2f(4.5,7);
glVertex2f(18.5,7);
glVertex2f(18.5,8);
glEnd();



//windows
glColor3f(0.9,0.9,0.9);
glBegin(GL_POLYGON);
glVertex2f(6,7);
glVertex2f(6,7.5);
glVertex2f(8,7.5);
glVertex2f(8,7);
glEnd();

glTranslated(3,-0.5,0);

glBegin(GL_POLYGON);
glVertex2f(6,7.5);
glVertex2f(6,8);
glVertex2f(8,8);
glVertex2f(8,7.5);
glEnd();

glTranslated(3,0,0);

glBegin(GL_POLYGON);
glVertex2f(6,7.5);
glVertex2f(6,8);
glVertex2f(8,8);
glVertex2f(8,7.5);
glEnd();
```

```
glTranslated(3,0,0);

glBegin(GL_POLYGON);
glVertex2f(6,7.5);
glVertex2f(6,8);
glVertex2f(8,8);
glVertex2f(8,7.5);
glEnd();

//Steam Pipes
glColor3ub(21,21,21);
glTranslated(-8,-1.6,0);
glBegin(GL_POLYGON);
glVertex2f(4.1,10);
glVertex2f(4,12.9);
glVertex2f(5.6,12.9);
glVertex2f(5.7,10);
glEnd();

glColor3ub(221,147,0);
glBegin(GL_POLYGON);
glVertex2f(4,12.9);
glVertex2f(3.9,13.5);
glVertex2f(5.5,13.5);
glVertex2f(5.6,12.9);
glEnd();

glTranslated(3,0,0);

glColor3ub(21,21,21);
glBegin(GL_POLYGON);
glVertex2f(4.1,10);
glVertex2f(4,12.9);
glVertex2f(5.6,12.9);
glVertex2f(5.7,10);
glEnd();

glColor3ub(221,147,0);
glBegin(GL_POLYGON);
glVertex2f(4,12.9);
glVertex2f(3.9,13.5);
glVertex2f(5.5,13.5);
glVertex2f(5.6,12.9);
glEnd();

glTranslated(3,0,0);
```

```
  glColor3ub(21,21,21);
  glBegin(GL_POLYGON);
  glVertex2f(4.1,10);
  glVertex2f(4,12.9);
  glVertex2f(5.6,12.9);
  glVertex2f(5.7,10);
  glEnd();

  glColor3ub(221,147,0);
  glBegin(GL_POLYGON);
  glVertex2f(4,12.9);
  glVertex2f(3.9,13.5);
  glVertex2f(5.5,13.5);
  glVertex2f(5.6,12.9);
  glEnd();
}
```

II. Sky Function:

```
// Placeholder for drawing sky
  glPushMatrix();
  glBegin(GL_POLYGON);
  glColor3ub(168,243,255);
  glVertex2f(0,800);
  glColor3ub(32,180,220);
  glVertex2f(0,410);
  glColor3ub(168,243,255);
  glVertex2f(1000,410);
  glColor3ub(32,180,220);
  glVertex2f(1000,800);
  glEnd();
  glPopMatrix();
```

III. Water Function:

```
// Placeholder for drawing water
  glPushMatrix();

  glBegin(GL_POLYGON);
  glColor3ub(75,160,240);
  glVertex2f(0,0);
  glColor3ub(47,136,220);
  glVertex2f(0,300);
  glColor3ub(47,136,220);
  glVertex2f(1000,400);
  glColor3ub(75,160,240);
  glVertex2f(1000,0);
  glEnd();
  glPopMatrix();
```

IV. Ice Function:

```
// Placeholder for drawing icebergs
  glPushMatrix();
  glTranslated(400,150,0.0);
  glScaled(20,10,0);
  glColor3ub(102,255,51);

  glBegin(GL_POLYGON);
  glVertex2f(5.5,2.5);

  glVertex2f(12.5,19.5);
  glVertex2f(15,19.5);
  glVertex2f(12.5,19.5);
  glVertex2f(13.5,18.5);
  glVertex2f(16.5,20.5);
  glVertex2f(17.5,18.5);
  glVertex2f(18.5,3.5);
  glVertex2f(19,3);
  glVertex2f(19.5,2.5);
  glEnd();
  glBegin(GL_POLYGON);
  glVertex2f(5.5,2.5);
  glVertex2f(6,3);
  glVertex2f(8.25,3.5);
  glVertex2f(8.5,18.5);
  glVertex2f(12,15);
  glVertex2f(13,17);
  glVertex2f(12.5,19.5);
  glVertex2f(8.5,9.5);
  glVertex2f(12.5,2.5);
  glVertex2f(5.5,2.5);
  glEnd();
  glPopMatrix();
```

3. **Iceberg Collision**: Implement a realistic collision between the Titanic and an iceberg, considering factors such as the ship's design and the progression of water through the lower decks.
4. **Listing and Sinking**: Visualize the gradual listing of the Titanic it takes on water, leading to its eventual sinking.
5. **Passenger and Crew Interaction:** Populate the simulation with virtual passengers and crew members to illustrate their actions and decisions during the sinking, including the launching of lifeboats and the chaos on board.

6. **Sound Functionality**

```
/// Sound Funtion///
void AfterLauncesound()
{
if (!isSound)
{
        isSound=true;
       //PlaySound("Titanic-Ringtones.wav ", NULL, SND_ASYNC|SND_FILENAME);
       PlaySound("Titanic-Ringtones.wav
",NULL,SND_ASYNC|SND_FILENAME|SND_LOOP);
}
}
void Groundsound() {
if (!isSound) {
    isSound=true;
    //PlaySound("a.wav", NULL, SND_ASYNC|SND_FILENAME);
    PlaySound("powerup.wav", NULL,SND_ASYNC|SND_FILENAME|SND_LOOP);
    //PlaySound("cras.wav", NULL,SND_ASYNC|SND_FILENAME|SND_LOOP);
    }
}
```

7. **Documentation and Presentation:** Provide comprehensive documentation detailing the development process, technical aspects of the simulation, and historical research. Additionally, create a compelling presentation to showcase the project's objectives, methodology and outcomes.

By developing this Titanic sinking simulation project using OpenGL, we aim to create an educational and emotionally engaging experience that allows users to gain a deeper understanding of one of the most significant disasters of the 20th century.

Objective of the project

The objective of the project "Titanic sinking simulation using OpenGL" is to create a visually immersive and educational experience that allows users to witness, in a virtual environment, the sinking of the RMS Titanic, one of the most infamous maritime disasters

in history. By leveraging OpenGL, a powerful graphics library, the project aims to render realistic 3D models of the Titanic, its surroundings, and the events leading up to and following its sinking.

Methodology

Creating a Titanic sinking simulation graphics project using OpenGL involves specific steps tailored to this graphics library. Here is an overview of the methodology:

1. **Project Planning and Research**: Begin by outlining the scope and objectives of the project. Research historical data, including ship blueprints, historical records, survivor testimonies, and relevant literature. Understand the key events and factors surrounding the Titanic's sinking to inform the simulation.
2. **OpenGL Setup**: Set up your development environment with the necessary tools and libraries for OpenGL programming. This includes installing OpenGL drivers, an Integrated Development Environment (IDE), and OpenGL utility libraries like GLFW or GLUT.
3. **Scene Setup**: Use OpenGL to create the scene environment, including the ocean, sky, and surrounding landscape. Implement lighting and shading techniques to enhance the realism of the scene.
4. **Animation and Simulation**: Use OpenGL's rendering capabilities to animate the movement of the Titanic and other objects in the scene. Implement physics-based simulation algorithms to simulate the ship's motion, water dynamics, and interactions with objects like icebergs.
5. **Optimization and Performance**: Optimize the simulation for performance by minimizing rendering overhead and optimizing code efficiency. Implement techniques such as level-of-detail rendering and frustum culling to improve frame rates and reduce computational overhead.

6. **Testing and Debugging**: Test the simulation extensively to identify and fix any bugs, glitches, or performance issues. Solicit feedback from testers and make necessary adjustments to improve the user experience.
7. **Documentation and Deployment**: Document the project thoroughly, including user instructions, technical documentation, and any educational content integrated into the simulation. Package the simulation for deployment, whether as a standalone application or web-based experience and distribute it to users.

By following this methodology, you can create a Titanic sinking simulation graphics project using OpenGL that is immersive, educational, and historically accurate, providing users with a compelling experience that honors the memory of those affected by the tragedy.

Significant of the project:

A "Titanic sinking simulation" graphics project can hold significant educational, historical, and emotional value. Here is why:

1. **Historical Accuracy**: Simulations allow viewers to experience historical events in a dynamic way. By accurately portraying the sinking of the Titanic, viewers can gain a deeper understanding of the sequence of events, contributing factors, and the human experience during the tragedy.
2. **Educational Tool**: Such simulations can be invaluable educational tools. They can help students and enthusiasts comprehend the complexities of maritime disasters, engineering failures, and the human stories intertwined with them. Interactive elements can engage learners more effectively than static textbooks or documentaries.
3. **Technological Showcase**: Creating a simulation involves advanced graphics, physics engines, and programming skills. It can serve as a showcase of technical prowess, demonstrating the capabilities of computer graphics and simulation technology.

4. **Memorial and Remembrance**: The Titanic sinking remains one of the most iconic and tragic events in history. A simulation project can serve as a memorial, honoring the lives lost and the lessons learned from the disaster. It can evoke empathy and reflection on the human cost of hubris and technological failure.
5. **Research and Analysis**: Simulations can be based on extensive research and data analysis. They can incorporate historical records, survivor testimonies, and scientific knowledge to provide insights into the circumstances surrounding the sinking. Researchers can use such projects to test hypotheses and explore alternative scenarios.
6. **Engagement and Awareness**: In today's digital age, multimedia projects have the potential to reach wide audiences. A well-executed Titanic sinking simulation can captivate viewers and raise awareness about maritime safety, historical preservation, and the enduring legacy of the Titanic.

Overall, a Titanic sinking simulation graphics project can blend art, technology, education, and commemoration, making it a significant endeavor with multifaceted importance.

Conclusion:

The Titanic sinking simulation graphics project stands as a testament to the intersection of technology, education, and remembrance. Through meticulous research, advanced graphics, and historical accuracy, this project endeavors to bring the tragedy of the Titanic to life in a compelling and immersive manner. By leveraging simulation technology, it offers viewers a dynamic and educational experience, allowing them to gain deeper insights into the sequence of events, the human stories involved, and the lessons learned from this historic disaster.

As a tool for education, the simulation provides a unique opportunity for students and enthusiasts to engage with history in a way that traditional textbooks or documentaries cannot match. Its interactive nature encourages active learning, fostering a deeper

understanding of maritime disasters, engineering challenges, and the human experience amidst catastrophes.

Moreover, the project serves as a technological showcase, highlighting the capabilities of modern graphics and simulation technology. Its creation involved a fusion of artistic vision, technical expertise, and historical research, resulting in a visually stunning and intellectually stimulating representation of the Titanic's fateful voyage.

Beyond its educational and technological significance, the simulation project also serves as a memorial, honoring the lives lost aboard the Titanic and reminding us of the enduring legacy of this tragedy. Through its portrayal of the sinking, it evokes empathy, reflection, and remembrance, ensuring that the stories of those who perished are not forgotten.

In conclusion, the Titanic sinking simulation graphics project embodies the power of multimedia to inform, inspire, and commemorate. It is a testament to the enduring fascination with the Titanic story and a poignant reminder of the human cost of hubris and tragedy.

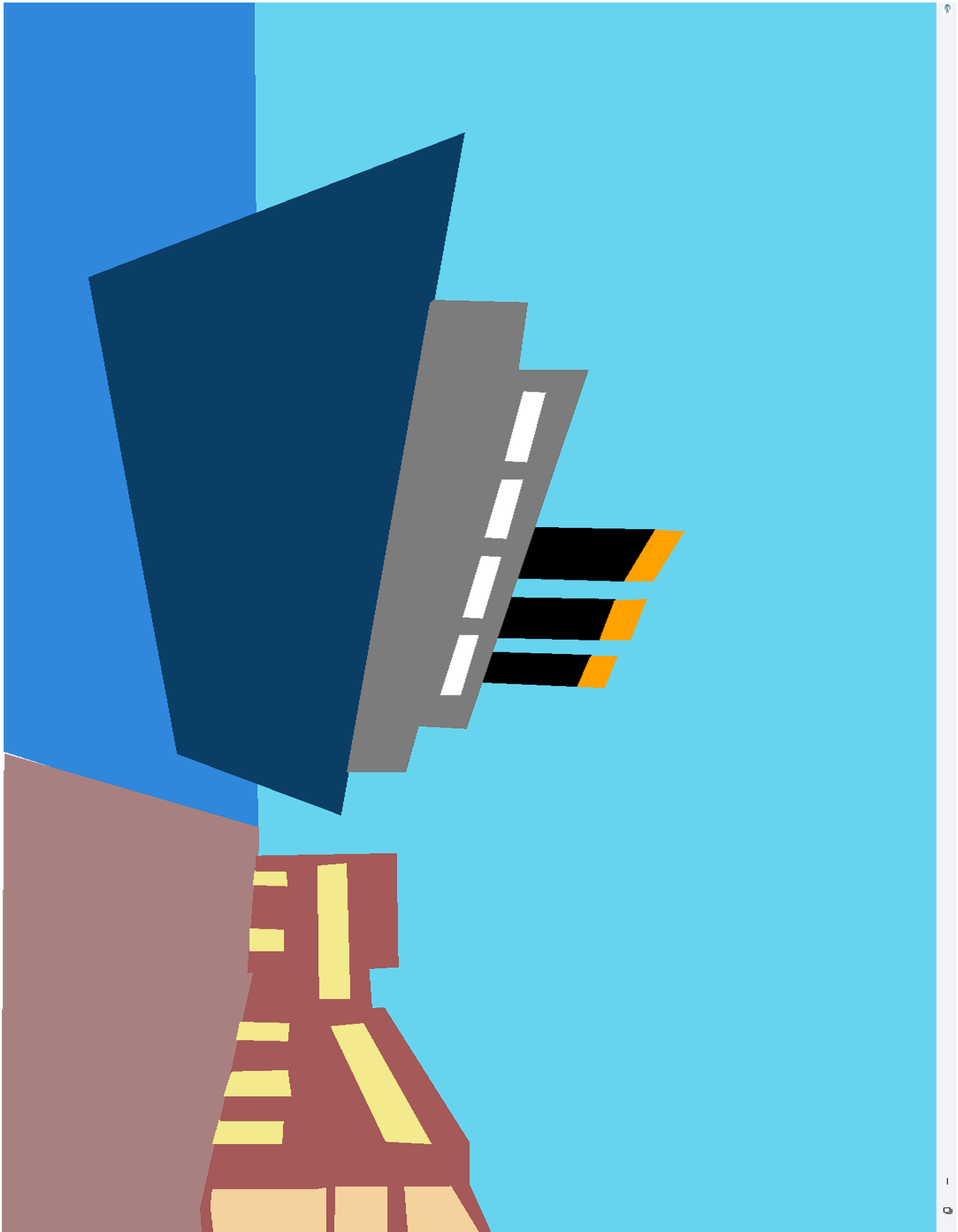
Reference

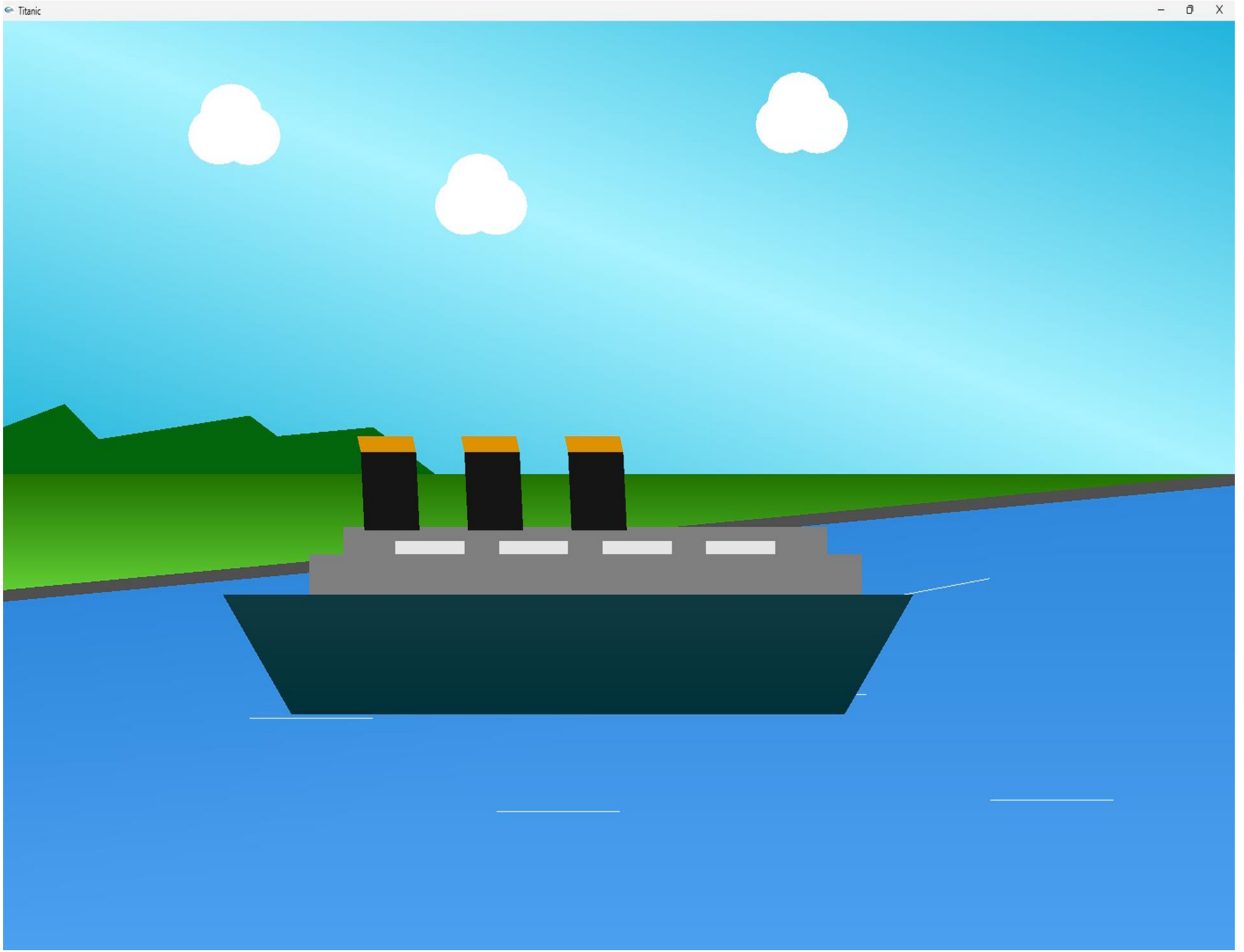
1. **GitHub Repository Link:**

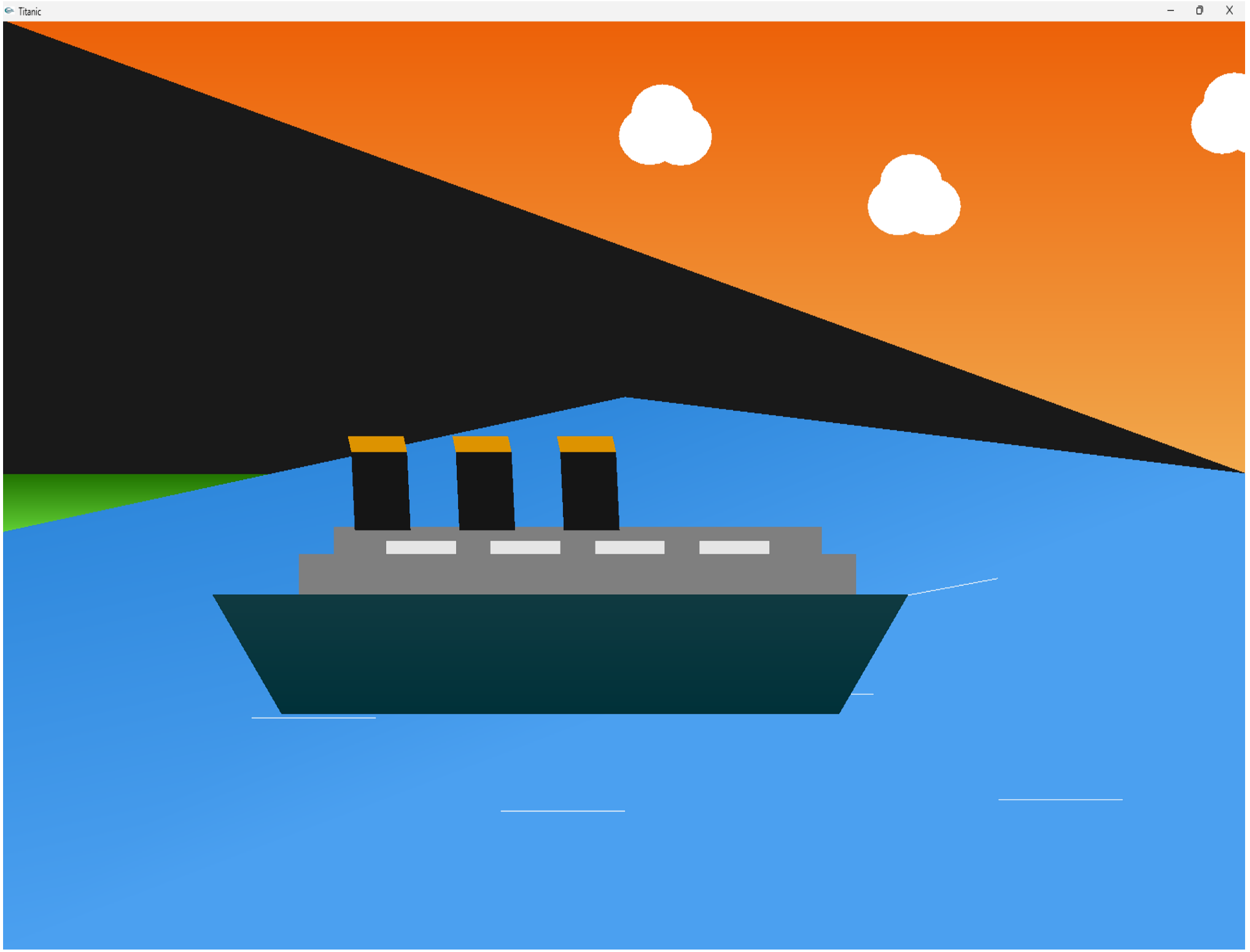
   https://github.com/abdullahjarif/GRAPHCS-_Project-_TITANIC-SINKING-SIMULATION

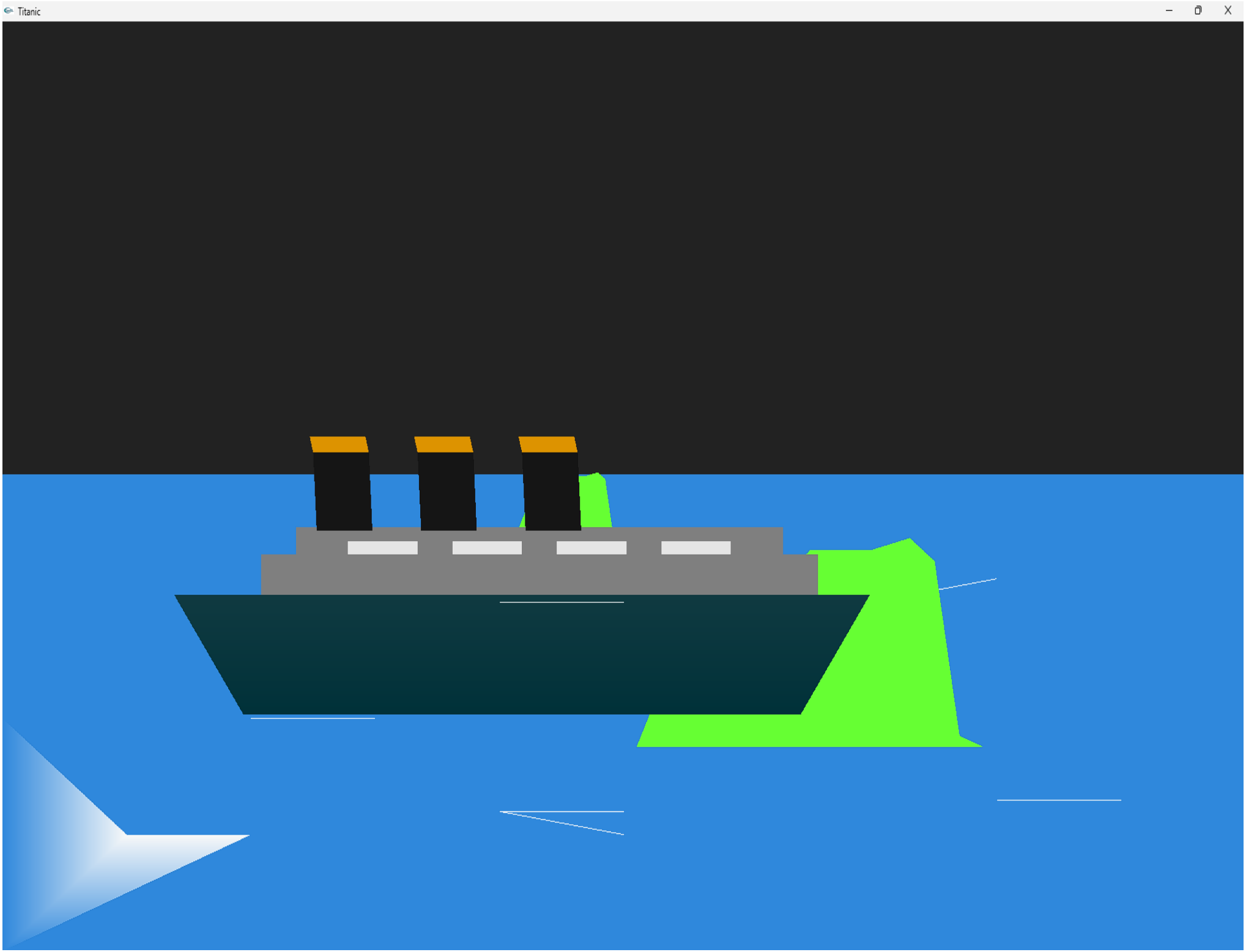
2. **YouTube Channel Link:**

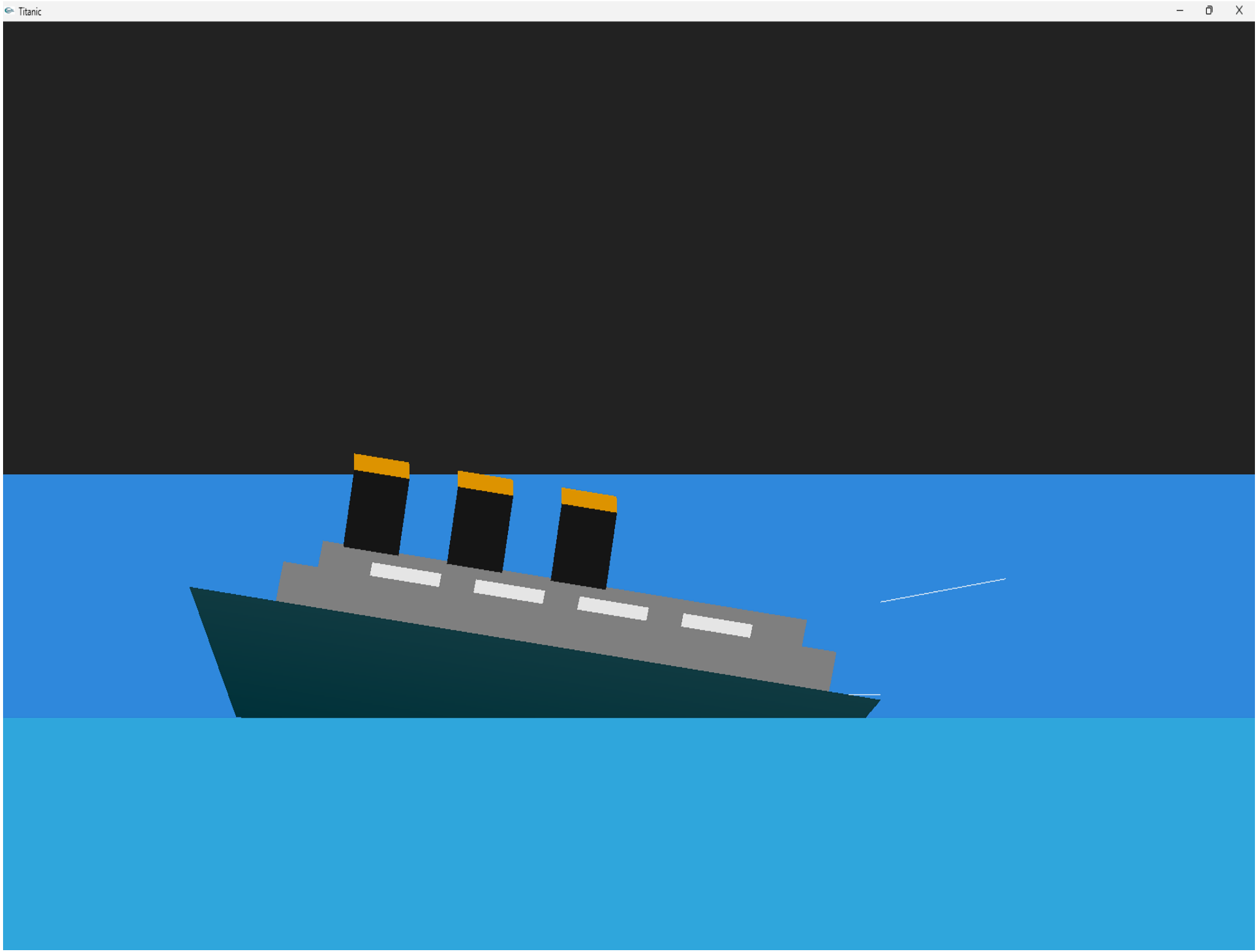
   https://youtu.be/ZVBDicopGzw




Titanic


Titanic


Titanic


Titanic