- Don’t modify any files or folders on GitHub that don’t belong to your scenes.
- Try to push any changes on GitHub right after working on your scene.
- Upload the relevant files (e.g., graph plot from Geogebra) to your scene folder on GitHub.
- The *gluOrtho2D* function must use the parameters:
  gluOrtho2D(-200, 200, -200, 200);
- Try to maintain a **coding format like this** for better understanding and easier integration:

Declaration Part	1. Declare all variables with initial values first, (if applicable).
	2. Declare the mouse and keyboard functions.
	3. Declare the draw functions (functions that are responsible for drawing different objects) with appropriate parameters.
	4. Declare the transition functions (e.g., car/boat moving) with appropriate parameters.
Implementation Part	5. Implement the mouse and keyboard functions
	6. Implement the logic for the object-drawing functions.
	7. Implement the logic for the transition functions.
	8. Define the *display()* function to handle rendering.
	9. Define the *init()* function to initialize OpenGL settings.
	10. Define the *main()* function to set up the GLUT window and start the main loop.

Example:

```
#include <GL/glut.h>
#include <iostream>
using namespace std;

// Declaration Part

// Step 1: Declare all variables with initial values first (if applicable)
float boat_move = 0.6f;
float boat_speed = 0.1f;
bool boat_direction = true;

// Step 2: Declare the mouse and keyboard functions
void mouse(int button, int state, int x, int y);
void keyboard(unsigned char key, int x, int y);
```

```
// Step 3: Declare the draw functions (functions responsible for drawing different objects)
void drawBoat(float x, float y);

// Step 4: Declare the transition functions (e.g., car/boat moving)
void moveBoat();

// Implementation Part

// Step 5: Implement the mouse and keyboard functions
void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        boat_move = -100;
    }
}

void keyboard(unsigned char key, int x, int y) {
    if (key == 'q') {
        exit(0);
    }
}

// Step 6: Implement the logic for the object-drawing functions
void drawBoat(float x, float y) {
    glColor3f(1.0f, 0.0f, 0.0f);
    glBegin(GL_POLYGON);
    glVertex2f(x, y);
    glVertex2f(x + 20, y);
    glVertex2f(x + 20, y + 10);
    glVertex2f(x, y + 10);
}

// Step 7: Implement the logic for the transition functions
void moveBoat() {
    if (boat_direction) {
        boat_move += boat_speed;
    } else {
        boat_move -= boat_speed;
    }

    if (boat_move > 100) boat_direction = false;
    if (boat_move < -100) boat_direction = true;

    glutPostRedisplay();
}

// Step 8: Define the display() function to handle rendering
void display() {
```

```
    glClear(GL_COLOR_BUFFER_BIT);
    drawBoat(boat_move, -20);
    glFlush();
}

// Step 9: Define the init() function to initialize OpenGL settings
void init() {
    gluOrtho2D(-200, 200, -200, 200);
    glutMouseFunc(mouse);
    glutKeyboardFunc(keyboard);
}

// Step 10: Define the main() function to set up the GLUT window and start the main loop
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Boat Animation");
    init();
    glutDisplayFunc(display);
    glutTimerFunc(16, moveBoat, 0);
    glutMainLoop();
    return 0;
}
```

- Each object must be represented using a function. For example: if there are 4 trees in a scene, there will be 4 separate functions. And each function must have a *unique object ID*.
  Make sure to include the object ID in the comment next to each function, both in the declaration and implementation of the draw function of any object.
  Follow this naming format for your *unique object ID*:

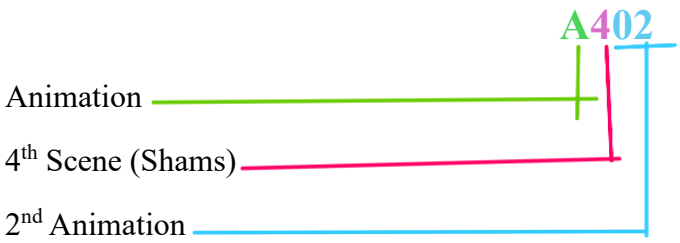
O – Your Scene Number – Object Number of Your Scene

Example of an object ID:

- Each animation must be represented using a function. And each animation must have a *unique animation ID*.
  Make sure to include the animation ID in the comment next to each function, both in the declaration and implementation of the animation functions.
  Follow this naming format for your *unique animation ID*:

A – Your Scene Number – Animation Number of Your Scene

Example an animation ID:



Example of Code with Object ID and Animation ID:

```
#include <GL/glut.h>
#include <iostream>
using namespace std;

float tree1_position = -30.0f;
float tree2_position = 20.0f;
float boat_position = 0.0f;
float boat_speed = 0.1f;

void mouse(int button, int state, int x, int y);
void keyboard(unsigned char key, int x, int y);

void drawTree1(float x, float y);   //  O201
void drawTree2(float x, float y);   //  O202
void drawBoat(float x, float y);    //  O203
void moveBoat();                    //  A201

void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        boat_position = -100;
    }
}
```

```
void keyboard(unsigned char key, int x, int y) {
    if (key == 'q') {
        exit(0);
    }
}

void drawTree1(float x, float y) {      // O201
    glColor3f(0.0f, 1.0f, 0.0f);
    ……
    glEnd();
}

void drawTree2(float x, float y) {      // O202
    glColor3f(0.0f, 0.8f, 0.0f);
    ……
    glEnd();
}

void drawBoat(float x, float y) {       // O203
    glColor3f(0.8f, 0.2f, 0.2f);
    ……
    glEnd();
}

void moveBoat() {                       // A201
    boat_position += boat_speed;
    if (boat_position > 100) boat_position = -100;
    glutPostRedisplay();
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    drawTree1(tree1_position, -20);  // O201
    drawTree2(tree2_position, -20);  // O202
    drawBoat(boat_position, -30);    // O203
    glFlush();
}

void init() {
    gluOrtho2D(-200, 200, -200, 200);
    glutMouseFunc(mouse);
    glutKeyboardFunc(keyboard);
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
```

```
    glutCreateWindow("Scene with Objects and Animations");
    init();
    glutDisplayFunc(display);
    glutTimerFunc(16, moveBoat, 0);  // A201
    glutMainLoop();
    return 0;
}
```

- There's a docx file in the GitHub in your scene folder. You must fill these tables up in the docx file within your scene folder according to your scene:

Table 1:

SL#	Object ID	Function Name	Object Name

Table 2:

SL#	Animation Function ID	Animation Function	Object/Scene

Please remember that if you do not fill out these tables in the docx file within your scene folder on GitHub, your part will be considered incomplete in the project report.