CPT205 Computer Graphics
2D Modelling Project Report

Xinling Du
2363292
Information and Computing Science

November 2025

1 Introduction

This project utilized the Freeglut library of OpenGL to develop a 2D interactive electronic greeting card in Microsoft Visual Studio to celebrate the 20th anniversary of XJTLU's establishment. This card features bright colors and multiple cute elements: butterflies representing growth and transformation, rising balloons symbolizing aspirations, and the central Xipu bird mascot embodying campus vitality. Key landmarks include XJTLU's central building and Suzhou's "Eye of Suzhou" Ferris wheel. This project integrates multiple graphics techniques including model creation, transformations, projection, animation, and interaction. These applied techniques ensure that the anniversary card meets the expected design objectives in terms of both visual quality and functionality.

The card integrates multiple graphics techniques including geometric creation, transformations, animation, and interactivity. These applied techniques ensure that the anniversary card meets the expected design objectives in terms of both visual quality and functionality.

2 Design and Features

2.1 Design Concept

The scenes depicted in this greeting card include:

- A central butterfly that triggers the main animation when clicked
- Animated balloons that rise from the ground
- Rotating Ferris wheel with colorful cabins
- Floating clouds with gentle movement
- Animated bird with flapping wings and jumping motion
- Central building in the background
- Interactive butterfly generation after the main animation

- Smooth anniversary text fade-in effect
- Softly colored grass and randomly generated bushes
- Sun with gentle light

2.2 Graphics Techniques Used

The implementation utilizes various computer graphics techniques:

- Geometry Creation: Basic shapes (triangles, quads, polygons) for objects like butterflies, birds, and buildings
- Transformations: Translation, rotation, and scaling for object positioning and animation
- Viewing: Orthographic projection with dynamic window resizing support
- Animation: Timer-based incremental updates for wing flapping and object rotation, exponential smoothing interpolation for butterfly movement, and sinusoidal functions for natural floating motions
- Color Blending: Alpha blending for smooth transitions and visual effects
- Interactivity: Mouse and keyboard events for user control

3 Program Instructions (Readme)

3.1 How to Run

1. Extract the submitted ZIP file containing the executable and source code
2. Run the `.exe` file (requires freeglut DLLs)
3. The program will open in a 1000×700 pixel window

3.2 Interactive Commands

- **Click the central butterfly** to start the main animation sequence
- **Click anywhere** after the anniversary message appears to create new butterflies
- **Resize the window** to see adaptive layout and object repositioning
- **Press 'C'** to randomize butterfly colors
- **Press 'R'** to reset the scene to initial state
- **Press ESC** to exit the program

3.3 Animation Sequence

1. Initial state: Single butterfly in center, balloons at bottom, animated bird
2. Click butterfly: 20 butterflies scatter in circular pattern
3. Balloons continue rising and eventually disappear
4. Anniversary text fades in gradually
5. After text fully appears: Click to add more butterflies anywhere

4 Implementation Details

4.1 Program Architecture

The program uses key data structures and a state machine to manage interactive animations:

Butterfly Structure:

```
struct Butterfly {
    float x, y;          // Position
    float scale;         // Size
    float wingAngle;     // Wing animation
    float targetX, targetY; // Movement targets
    float color1[3], color2[3]; // Gradient colors
};
```

Balloon Structure:

```
struct Balloon {
    float x, y;          // Position
    float radius;        // Size
    float color[3];      // RGB color
    float targetY;       // Target height
    bool active;         // Visibility state
};
```

Bird Structure:

```
struct Bird {
    float x, y;          // Position
    float wingAngle;     // Wing flapping
    float jumpOffset;    // Bouncing motion
    float tShirtColor[3]; // Clothing color
};
```

State Definitions:

```
enum ProgramState {
    SINGLE_BUTTERFLY,
    MULTIPLE_BUTTERFLIES,
```

```
    SHOW_MESSAGE
};
ProgramState currentState = SINGLE_BUTTERFLY;
```

State Transitions:

- SINGLE_BUTTERFLY → MULTIPLE_BUTTERFLIES (on butterfly click)
- MULTIPLE_BUTTERFLIES → SHOW_MESSAGE (after animation progress)
- SHOW_MESSAGE remains active for interactive butterfly creation

4.2 Animation System

The timer-driven animation system combines multiple techniques:

```
void timer(int value) {
    // Timer-based incremental animations
    bird.wingAngle += 0.3f;                    // Wing flapping
    wheelAngle += 0.3f;                        // Ferris wheel rotation
    butterflyFloatOffset += butterflyFloatSpeed; // Butterfly floating
    birdJumpPhase += birdJumpSpeed;            // Bird jumping

    // Smooth interpolation for butterfly movement
    for (auto& b : butterflies) {
        b.x += (b.targetX - b.x) * 0.08f;
        b.y += (b.targetY - b.y) * 0.08f;
    }
}
```

4.3 Object Creation

Each graphical object is implemented with custom drawing functions:

- **Butterflies**: Triangle fans with gradient colors and floating motion
- **Balloons**: Circular shapes with strings and highlight effects
- **Bird**: Composite shape with flapping wings and bouncing motion
- **Structures**: Line-drawn Ferris wheel and XJTLU building
- **Environment**: Multi-layered clouds and gradient backgrounds

4.4 Window Resize Handling

The program dynamically adapts to window size changes:

- Objects reposition smoothly to new locations
- Butterfly formation adjusts to new center point
- Balloons and background elements scale appropriately
- Text remains centered regardless of window dimensions

5 Screenshots

Below are representative screenshots showing the program's visual progression:

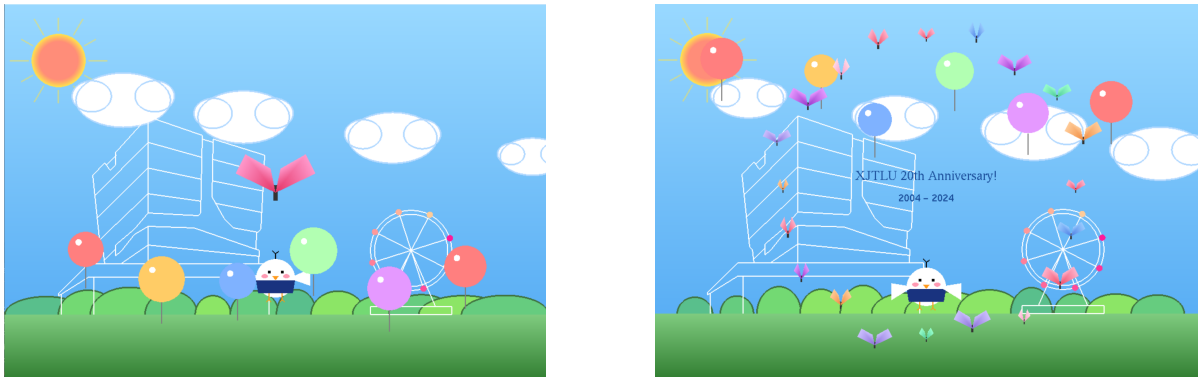

Figure 1: (a) Initial state with central butterfly (b) Butterflies scattering with text fade-in

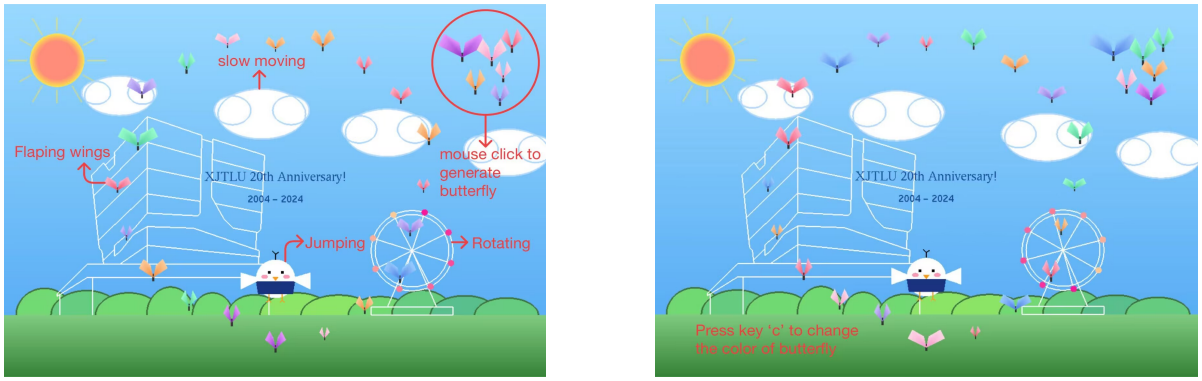

Figure 2: (a) Full animation with all elements (b) Color change after pressing 'C'

6 Conclusion

This project utilized graphic techniques covered in the module to develop an interactive 2D greeting card, which meets the project requirements. The greeting card incorporate both static and animated objects, while also show some backgrounds of XJTLU, and has achieved mouse and keyboard interaction. It not only displays exciting 20th anniversary celebration of XJTLU, but also creates a positive experience for digital greeting card usage.