CPT205 Assignment 1 Technical Report

Name	Zihan	Deng
ID Number	2144196	
Programme	Information and Computing Science	
Module Title	Computer Graphics	
Module Code	CPT205	
Assignment Title	Assessment 1 – 2D Modelling Project	

Introduction

This program is developed based on the Freeglut library, with the aim of creating a two-dimensional electronic greeting card for inviting participants to the 2024 XJTLU graduation ceremony. The objective is to faithfully replicate the school's official invitation using a selection of representative design elements based on the school's promotional materials.

The invitation card features the school's classic purple gradient as the background color, and includes silhouettes of the SIP campus central building, the iconic structures of the University of Liverpool, and the school's emblem as its primary elements. Figure 1 illustrates the sources of these iconic elements in the greeting card and their transformation within the program.

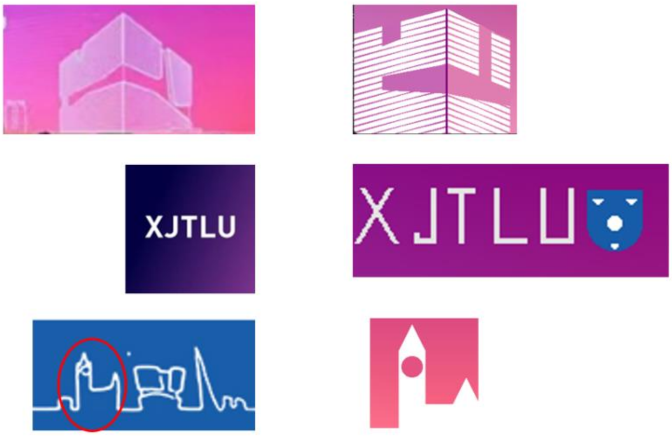
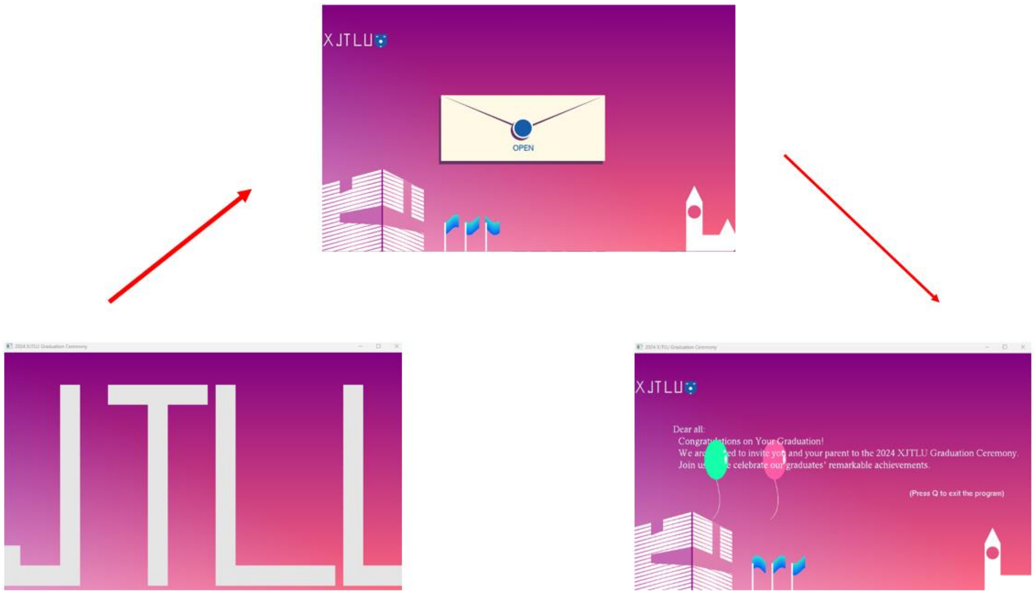


Figure.1 Typical elements on invitation card and its source (Source: ebridge.xjtlu.edu.cn, XJTLU official WeChat account)

In terms of implementation, Figure 2 provides a concise overview of the three main components of the program flow. The program commences with the scrolling display of the text "2024 XJTLU" across the full screen, followed by various objects representing the campus entering from the screen's periphery. An unopened letter appears in the center of the window, with the wax seal of the letter designed to be an interactive button, enabling users to click and reveal the invitation text, accompanied by celebratory animations such as flag waving and rising balloons. Finally, the program can be exited using keyboard input.



Dear all:
Congratulations on Your Graduation!
We are [illegible] to invite you and your parent to the 2024 XJTLU Graduation Ceremony.
Join us [illegible] celebrate our graduates' remarkable achievements.

(Press Q to exit the program)

Figure 2. Brief illustration of program process

The report will be divided into four sections. The first section offers a brief introduction to the program, while the second and third sections will provide detailed explanations of the implementation of graphical features and interactive animation effects. The final discussion will focus on program optimization and identified limitations.

Graphic Features Implementation

The features of the program's visuals are composed of graphics and text elements, with these complex shapes being constructed from basic graphic primitives.

Basic Primitives

For the sake of readability and convenience, functions for basic graphic primitives were initially designed. These basic graphic functions include: circles (`drawCircle`), rectangles (`drawRectangle`), ellipses (`drawEllipse`), parallelograms (`drawParallelogram`), rounded rectangles (`drawRoundedRectangle`), and cubic Bézier curves (`drawCubicBezierCurve`). Notably, in this program, a rounded rectangle is defined as a shape composed of two circles and one rectangle. Specifically, as the centers of the circles are located on the shorter sides of the rectangle, this function determines the circle centers' positions by examining the width and height and then calling the appropriate functions. Furthermore, the cubic Bézier curves in this program employ fixed cubic curve functions and are used for drawing balloon strings.

In addition to functions for drawing complex shapes, functions for rendering text must also be considered. Through investigation, the FreeGLUT library provides users with bitmap fonts for convenient text rendering and utilizes the `glutBitmapCharacter` function. Therefore, functions for printing bitmap text were included to achieve the effect of greeting card text.

Complex Shapes

When it comes to complex shapes, they can be roughly categorized into text and other complex graphics. Since the FreeGLUT library only supports bitmap fonts, rendering clear large fonts can be challenging. Therefore, the program uses the composition of rounded rectangles to achieve the rendering of large digits and employs rectangles and parallelograms for handling the letters "XJTLU."

Similarly, functions for other complex graphics (such as `drawCB`, etc.) are constructed by combining and manipulating basic graphic primitives. Taking the school emblem as an example, for the sake of computational simplicity, the emblem is composed of a rectangle and a semicircle, and the pattern within the emblem is also simplified to a combination of triangles and circles.

The flag on the right side of the silhouette of the Central Building is another distinctive shape. To achieve the simple waving effect of the flag, a function for drawing a rectangular shape with a waving effect was designed. This shape is composed of adjacent stripes, and its appearance is controlled through trigonometric functions and iterations to introduce an offset. The function also introduces a variable, `flagStep`, to implement changes in the flag after each redraw, which will be discussed in the next section.

Regarding color selection for the graphics, due to the gradient purple background, decorative objects are typically depicted in white. Furthermore, to simulate pseudo-shadow effects on certain elements, when drawing, a slightly offset dark-colored shape is initially drawn. In the case of drawing balloons, a simple rounded rectangle and circle are used as highlights, adding realism to the visuals.

Interaction and Animation Implementation

Animation Implementation

Many objects in the program incorporate animation effects. To achieve these animation effects, a series of `GLfloat` variables need to be defined, and timer functions are utilized. Additionally, callback functions are employed to refresh the screen. The majority of timers in the program are controlled by a reference point of the objects, utilizing conditional recursive or return operations based on a step variable. The initial animation in the program presents a marquee consisting of large captions moving in the negative x-axis direction. When the timer controlling the movement of the captions determines that they have moved out of the window, it triggers timers for the decorations required for the second scene, causing these decorations to move into the window. Once objects such as envelopes and silhouettes have completed their movements, the program waits for a mouse-click event. After interaction occurs, the upper and lower parts of the envelope move out of the frame from two different directions, followed by the activation of timers for the balloon animation and flag waving. Regarding the animation of the balloons, to simulate the realistic effect of balloons rising in the air, the timer employs a non-linear y-axis ascent and a simple cyclic x-axis movement. As mentioned earlier, the flag waving effect is achieved by a timer controlling the `flagStep` variable, redrawing each frame of the flag, creating a sinusoidal wave-like motion.

Interaction Implementation

This program incorporates clear mouse and keyboard interaction effects. Firstly, the left mouse button click operation is used for the "Open Mail" animation. When the program detects a click event within the coordinate range of the button (wax seal on the mail), it initiates a looped animation of celebratory decorations. Towards the end of the program, you can use the keyboard input "Q" to exit. Figure 3 shows the interaction button.

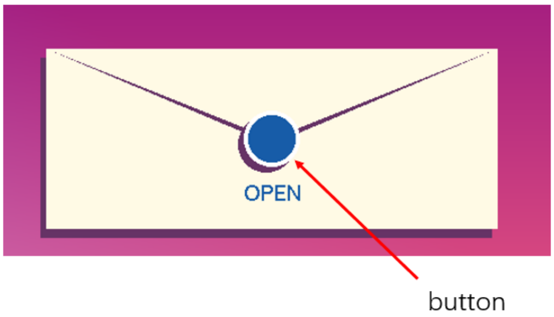


Figure 3. Envelope and button

Discussion

In summary, a diverse array of graphical rendering functions and callback mechanisms are employed to present a rich spectrum of dynamic effects and patterns. However, there remain several areas awaiting optimization. First and foremost, optimization of algorithmic complexity is under consideration. For instance, the rendering of basic shapes should lend itself to the application of curves in order to achieve greater irregularity in the plane. Furthermore, in terms of realism, there exists considerable room for improvement in certain dynamic effects. To achieve a more authentic simulation of flag and balloon physics characteristics, the integration of more intricate timing functions, such as dynamic Bézier curves, should be contemplated. Lastly, advancements can be made in the realms of particle effects and the lifespan management of graphical objects, which will contribute to the program's overall refinement in the future. Fireworks, as one of the symbols of celebration, necessitate the utilization of more intricate particle effects, potentially involving texture manipulation.