Technical Report on the Implementation of a 2D Birthday Card Using freeglut

Module Code: CPT205
Name : Xu Chen
ID: 2257453
Degree Programme: Information and Computer Science

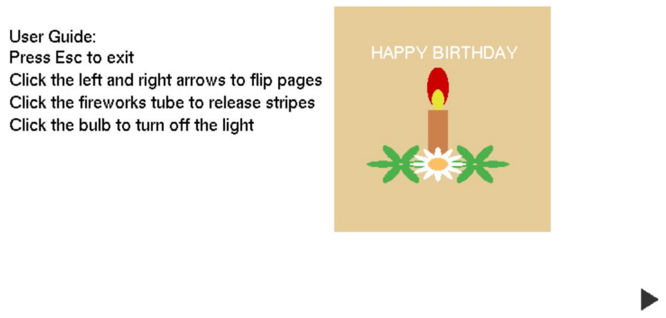
1 Introduction

A birthday card, as an important medium for expressing blessings and emotions, carries people's heartfelt wishes and deep affection for their friends and loved ones. In this project, I chose a warm color scheme as the main theme to create a cozy and pleasant atmosphere. The cover is decorated with carefully designed yellow-centered white petals and green leaves, which not only enhance the card's visual appeal but also imbue it with symbolic meaning. The white petals signify purity and hope, while the green leaves represent the continuity and growth of life. This combination lends the card an elegant and refined appearance, while also conveying hopes and blessings for the recipient's future. Inside the card, the multi-layered birthday cake is designed to create a cheerful and festive ambiance, with carefully crafted graphics and color combinations to express sincere wishes and goodwill toward the recipient.

On a technical level, this project aims to develop a 2D birthday card with high interactivity and rich visual effects. To achieve this, the project uses only the freeglut library in OpenGL, which handles window management and input devices (keyboard and mouse), while MS Visual C++ is used as the development environment for coding, compiling, and debugging. By combining various graphics techniques—including geometric creation, transformations, view settings, animation, and interaction handling—the project achieves a seamless integration of static and dynamic elements and provides a smooth interactive experience for the user. These applied techniques ensure that the birthday card meets the expected design goals in terms of visual quality and functionality, bringing users a lively and enjoyable birthday greeting experience.

2 Functional Overview

The cover shows the warm theme of birthday cards. The central "Happy Birthday" text is decorated with yellow hearts, white petals and green leaves, symbolizing purity and hope. The User Guide on the left provides detailed instructions for users to understand how to interact with the card, including exiting, turning pages, and clicking to trigger. The arrow button on the right side of the interface prompts users to click to enter the internal page to further explore the content of the card.



User Guide:
Press Esc to exit
Click the left and right arrows to flip pages
Click the fireworks tube to release stripes
Click the bulb to turn off the light

Figure 1: Birthday card cover and usage guide

When you right-click on the first picture, you will enter the second picture. The internal page is themed with layers of birthday cakes, adding a celebratory atmosphere, and candles are lit on the top of the cake to symbolize the birthday ritual. There are arrow buttons on the left side of the page for users to switch freely between different pages. In addition, the fireworks tube design on the left and right bottom of the page increases interactivity. Users can click on the fireworks tube to trigger the stripe effect, simulating the celebration scene of fireworks. At the same time, the light bulb in the upper right corner of the page can be clicked to turn off the lights, bringing users a unique experience of turning off the lights and making wishes.

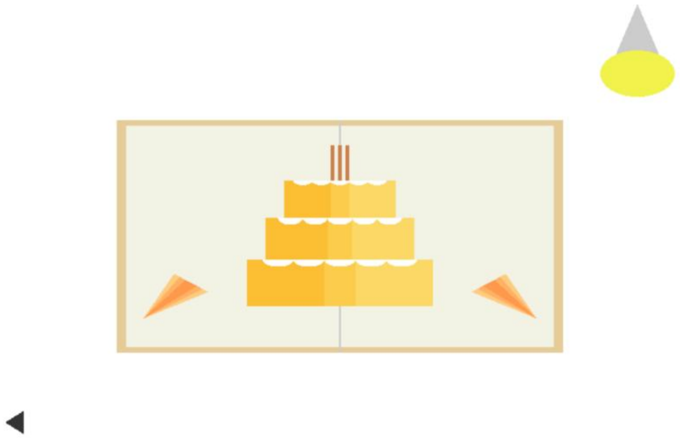
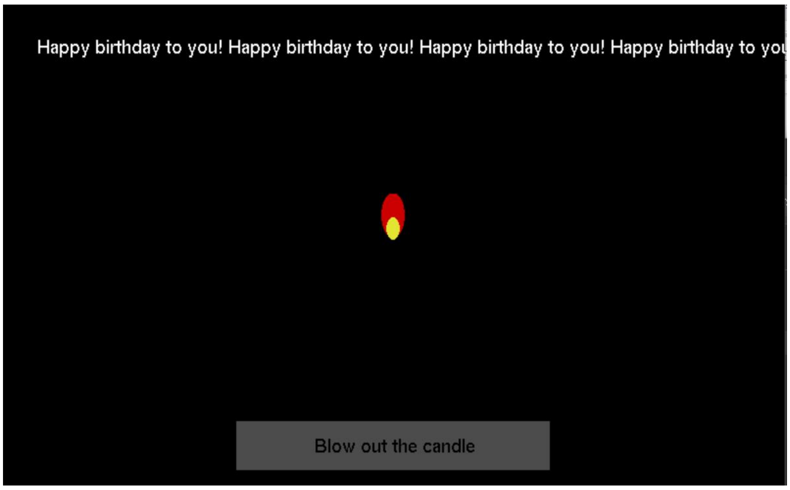


Figure 2: Birthday card interior decoration

When you click on the light bulb in Figure 2, the light turns off in Figure 3. The entire screen turns black, and the scrolling text "Happy Birthday to you!" appears at the top, adding a warm festive atmosphere to the scene. The flame pattern in the center simulates the effect of a burning candle, symbolizing the sacred moment of making a birthday wish. The "Blow out the candle" button at the bottom provides users with interactive options. After clicking this button, the page will return to the light state and display the celebration screen of the cake and fireworks again, allowing users to switch between dynamic light and shadow, enhancing the immersive experience of the birthday card.



Happy birthday to you! Happy birthday to you! Happy birthday to you! Happy birthday to yo

Figure 3: Wishing interface

3 Optimization of visual design and interactive experience

3.1 Realistic Visual Effects

In the detailed design, this project focuses on enhancing the realism and depth of the card elements to increase the overall visual appeal. The layered cake design, ***with its varied colors and refined surface decorations, adds a sense of dimensionality, allowing users to visually experience the cake's realistic texture.*** *In the flame design, the smooth transition between red and yellow effectively recreates the look of real flames, further enhancing the immersive atmosphere. Firework stripes gradually disappear as they rise, simulating the dispersing effect after an explosion. These intricate details not only add dynamic quality to the page but also allow each element to convey a celebratory atmosphere effectively.* ***Additionally, when hovering over or clicking on elements, they appropriately scale or change color to provide instant visual feedback, further enhancing user interaction and smoothness.***

3.2 Animation Smoothness

To ensure smooth animation effects, the project implements frame-by-frame updates for dynamic effects such as the rising stripes and scrolling banner. The transparency of the firework stripes gradually decreases as they ascend, creating a natural fading effect that avoids abrupt transitions. The scrolling banner text moves continuously and smoothly, maintaining a consistent speed to avoid any stuttering, thus providing users with a fluid visual experience.

3.3 Overall Layout Design

The color scheme of this project is designed to create a warm and natural visual atmosphere, fully expressing the warmth of birthday wishes. The background colors primarily consist of soft warm tones, while the cover background features a light beige shade, adding a sense of comfort to the overall aesthetic. Decorative elements, *such as flowers and flames, use vibrant colors like yellow, green, and red to create stronger visual impact and emphasize the joy of the celebration. The flame design uses a gradient of red and yellow to simulate a realistic burning effect. Additionally, the firework stripes display random color and transparency variations, simulating the vibrant aftermath of an explosion. Through the thoughtful use and combination of colors, the entire card appears harmonious and lively.*

The interface layout design emphasizes balance and layered presentation of information, ensuring users can easily access essential content during interaction. Symmetrically placed arrows and firework tubes on the left and right sides maintain a balanced layout, enhancing the overall visual harmony. Key information, such as the "Happy Birthday" text, is centered on the page as the focal point, while auxiliary information like the user guide is placed on the left side to avoid visual interference and ensure clear operational guidance. The positioning of each element is well-considered, avoiding overcrowding or excessive blank space, which ensures an overall sense of tidiness and organization.

4 Technical Implementation

4.1 Mouse interaction logic

*When the mouse moves, **glutPassiveMotionFunc()** captures the current mouse position and converts it to normalized coordinates relative to the window. The program determines whether it is hovering over an interactive element, such as an arrow, a firework cannon, or a light bulb based on the mouse position. The coordinate range of each element is predefined, and the hover state re-renders the element through the drawing function to show the immediate change of color or size.*
*---- Arrow: If the mouse is hovering in the range of the left or right arrow, update the **hoveredArrow** state to LEFT or RIGHT, and trigger the visual feedback of the corresponding arrow (such as color change or enlargement).*
*---- Firework cannon: Detect whether the mouse is hovering at the left or right firework cannon position, and update the **leftHovered** or **rightHovered** state variable to trigger the zoom effect.*
*---- Light bulb: Detect the hover state of the upper right light bulb, update the **bulbHovered** state and trigger the zoom effect.*

*Detect the click position in **glutMouseFunc()**, identify the element clicked by the user and trigger the corresponding event.*
---- Arrow click: Clicking the left and right arrows will trigger page switching: If you click the left arrow, set the page state to 0 (cover page). If you click the right arrow, set the page state to 1 (cake page). Re-render the page according to the changes in page and update the interface content.
*---- Firework tube click: Click the left or right firework tube to trigger the generation process of firework stripes. Call the **createStripes()** function to generate a certain*

*number of stripes at the position of the firework tube where the mouse is clicked. The **FireworkStripe** structure of each stripe contains properties such as position, color, angle, speed, etc. The speed, angle and color of the stripes are randomly generated to give the stripes diversity. Update the stripes array and add the generated stripes to the array to prepare for display in subsequent rendering.*
*---- Light bulb click: Click the light bulb to trigger the light-off effect. Set **bulbClicked** to true, switch the page to the light-off state, and trigger the scrolling banner animation. Call **glutTimerFunc** to start the **updateBanner** function to update the banner position regularly to achieve text scrolling. The page background turns black, with only the candle flame and the Blow out the candle button showing.*

4.2 Animation implementation logic

*---- Firework stripe animation: Update the position and transparency of each stripe in the stripes array frame by frame in the **updateStripes** function. The x and y coordinates of each stripe gradually change according to its speed and angle to achieve the rising effect of the stripes. Reduce the transparency (alpha) of the stripe in each frame. When the transparency is lower than a certain threshold, delete the stripe from the stripes array to achieve a natural disappearing effect. After each **updateStripes** run, call **glutPostRedisplay()** to request to redraw the page to present the dynamic effect of stripe rising and disappearing.*

---- Scrolling banner animation: The updateBanner function is called by glutTimerFunc at regular intervals (16 milliseconds) to regularly update the horizontal offset of the banner text to simulate the scrolling effect. The initial position of the text is on the right side of the screen, and the horizontal offset gradually decreases, causing the text to scroll to the left. When the offset of the banner is less than a certain value (for example, -4.0), reset the offset to the right side of the screen to achieve seamless loop scrolling.

5 Conclusion

This project uses freeglut libraries to implement a highly interactive and visually rich two-dimensional birthday card, successfully creating a warm and pleasant celebration atmosphere. Through meticulous color selection, smooth animation effects, and instant user interaction feedback, the greeting card brings a vivid and immersive experience to users. The entire implementation process includes page initialization, mouse hover and click event capture, frame-by-frame animation updates, and real-time rendering according to different page states. The organic combination of these functional modules makes the greeting card visually beautiful and realistic, and highly interactive. Through the grasp of details and the optimization of interaction logic, the project has achieved the expected design goals and provided users with an interesting and immersive birthday blessing experience.