

Technical Report on the Implementation of Greeting Card for 20th anniversary

Module Code: CPT205

Name : Chenrui Zhu

ID: 2361387

Degree Programme: Information and Computer Science

1 Introduction

This project implements a 2D interactive greeting card celebrating the 20th anniversary of Xi'an Jiaotong-Liverpool University (XJTLU). It aims to create a festive, lively, and visually engaging digital scene that blends artistic elegance with a sense of celebration. The card features multiple thematic elements, including colorful balloons, fireworks, fluttering ribbons, blooming flowers, and an abstract depiction of XJTLU's Central Building as the centerpiece. Each element is carefully designed to evoke joy, pride, and community, symbolizing the university's two decades of growth, academic excellence, and vibrant campus life.

Technically, the project is developed using OpenGL and the freeglut library, integrating static imagery with dynamic animations and real-time interactivity. Geometric modeling, coordinate transformations, layered rendering, particle effects, and event-driven input handling are employed to achieve smooth motion and responsive interaction. Users can engage with the scene, triggering animations such as fireworks launches or balloon releases, creating a sense of immersion. By combining artistic design with programming techniques, the project demonstrates how computational methods can enhance visual storytelling and interactive experiences.

2 Functional Overview

Cover (Page 0) The cover features an elegant and iconic scene, presenting the overall theme of Xi'an Jiaotong-Liverpool University's 20th Anniversary Celebration. The background transitions from purple to blue, representing the university's colors and evoking endless possibilities and visions. Blossoming flowers at the bottom add tranquility and liveliness, while stars scattered randomly bring a playful element to the page. The central title, 'Congratulations on the 20th Anniversary of Xi'an Jiaotong-Liverpool University,' is displayed in a vector outline font, emphasizing formality and a festive atmosphere. This page aims to create an immersive environment, offering users an entry point to commemorative experience, and slightly enlarges when the cursor hovers over the right arrow, guiding users to the next page (Figure1).



Figure 1: Cover page of the XJTLU 20th Anniversary Card

Inner Page (Page 1) When you press the arrow to go to the inner page, fireworks automatically launch from the center bottom (Figure 2). They can also be triggered by f/F, transitioning from the calm tone of the cover to a lively celebratory scene. The page features animated balloons (Figure 3a) rising with the swaying tails, an interactive ribbons launcher (Figure 3b) that releases colorful stripes when clicked with the mouse, and dazzling firework particles. The center building (CB) of Xi'an Jiaotong-Liverpool University is depicted in line art, connecting design with the university's cultural identity and reflecting its historical heritage. The color scheme shifts to warm beige and bright tones, symbolizing hope, vitality, and joy. Keyboard and mouse interactions allow users to release balloons (B/b), trigger fireworks (F/f), or switch pages, creating an interactive experience full of celebratory atmosphere.

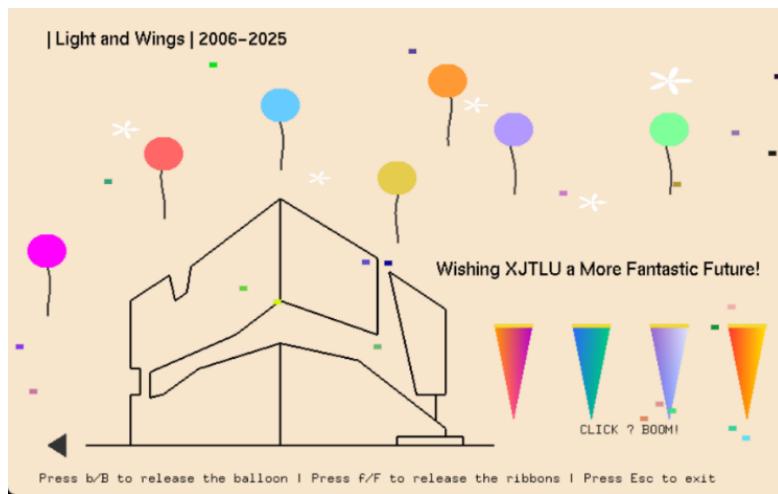


Figure 2: Inner celebration page of the XJTLU 20th Anniversary Card
(When switching pages or pressing f/F, fireworks will automatically burst from the bottom center of the page.)

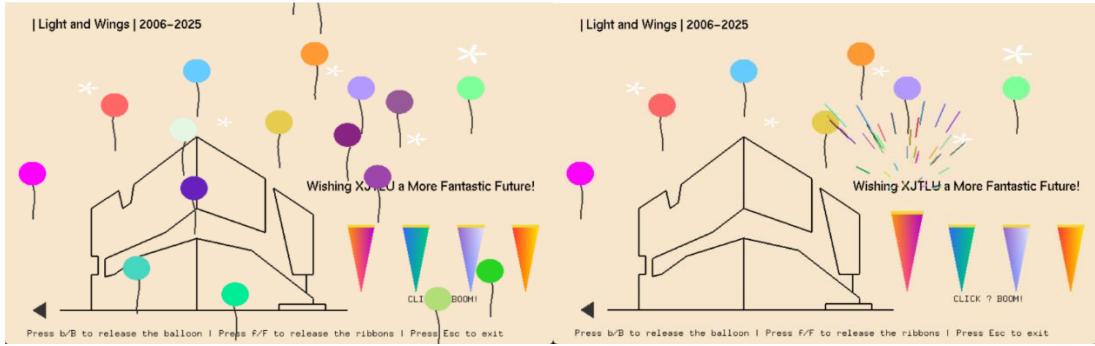


Figure 3a

Figure 3b

*Figure 3: Interactive Elements: Ribbon Launcher and Balloons
(Press b/B to release the colorful balloons | Click the launcher to launch ribbons)*

3 Optimization of visual design and interactive experience

3.1 Realistic Visual Effects

The project emphasizes realism and visual harmony through careful color blending, dynamic motion, and element layering. Each firework particle fades gradually as it rises, simulating physical dissipation. Ribbon particles are emitted at randomized angles and speeds, creating a natural celebratory spread. Balloons sway gently with animated tails using sinusoidal motion to enhance realism. Hover effects on interactive objects (such as arrows or launchers) cause smooth scaling and brightness changes, offering intuitive visual feedback and improving user engagement.

3.2 Animation Smoothness

Smooth animation is achieved using frame-based updates controlled by `glutTimerFunc`, running at approximately 60 frames per second. Particle systems for balloons, fireworks, and ribbons are updated incrementally with each frame, adjusting position, velocity, and lifespan. Transitions such as particle fading or balloon ascent are continuous and physics-inspired, ensuring fluid motion without abrupt changes. The consistent frame rate guarantees a seamless and visually appealing user experience.

3.3 Overall Layout Design

The layout employs symmetry and balanced composition to achieve both aesthetic and functional clarity. Arrow buttons on both sides guide page transitions and maintain visual balance. Decorative flowers and stars frame the lower and upper parts of the interface respectively, adding depth to the scene. The color palette transitions from deep purples and blues on the cover to bright, warm tones inside,

symbolizing transition from reflection to celebration. Each visual layer—from the static building outline to dynamic fireworks—cooperates to present a unified celebratory atmosphere.

3.4 Performance Optimization

*To maintain smooth rendering, unnecessary calculations are minimized, and inactive particles are skipped during updates. Double buffering (**GLUT_DOUBLE**) eliminates flickering by alternating between frame buffers. This optimization ensures that even with multiple dynamic effects active simultaneously, the animation remains stable and visually coherent.*

4 Technical Implementation

4.1 Mouse interaction logic

*The system uses **glutMouseFunc()** and **glutPassiveMotionFunc()** to manage click and hover interactions. The mouse position is converted into normalized device coordinates (ranging from -1 to 1), allowing accurate detection of interactions across different resolutions. Hover detection enables dynamic scaling and color change for arrows and ribbon launchers. Click detection is used to switch between pages or trigger ribbon launches. For example, clicking the right arrow on the cover page transitions to the inner page and automatically triggers a central firework.*

*Arrow: If the mouse is hovering in the region of the left or right arrow, the program updates the **hoverLeft** or **hoverRight** boolean variable accordingly. When hovered, the arrow slightly enlarges and brightens to indicate its interactivity.*

*Clicking the right arrow on the cover page changes the variable **page** from 0 to 1, switching the interface to the inner celebration page. Clicking the left arrow performs the opposite transition (from page 1 back to 0). Each page flip also sets the **autoFireOnPageChange** flag to **true**, automatically launching a central firework animation for added celebration.*

*Ribbon launcher: Detect whether the mouse is within the boundaries of a ribbon launcher at the bottom of the screen. The launcher's hover state is set to true when the cursor is inside its hitbox, triggering a smooth scale-up effect. This helps users identify clickable launchers visually. After detecting movement, the interface is re-rendered using **glutPostRedisplay()** to reflect the updated hover state immediately.*

*Mouse clicks are processed by the **glutMouseFunc()** callback, which identifies the clicked position and triggers corresponding actions depending on the current page and the target object.*

*When a user clicks on a launcher that is currently in a hover state, the program calls `spawnLauncherRibbons()` to generate a burst of multicolored ribbon particles. Each ribbon is represented by a **RibbonParticle** structure containing parameters such as position, velocity, color, and lifespan. The ribbons spread outward under initial velocity and gradually fade as their lifespan decreases, simulating a festive cannon effect.*

4.2 Rendering Pipeline

*The rendering process follows a structured pipeline: each frame begins by clearing the color buffer, then drawing the appropriate page background followed by dynamic and interactive elements. Firework particles, ribbon lines, stars, flowers, and text are drawn using OpenGL primitives (**GL_QUADS**, **GL_TRIANGLES**, **GL_LINE_STRIP**, and **GL_TRIANGLE_FAN**). Orthographic projection (**glOrtho2D**) is used for consistent 2D rendering. Finally, **glutSwapBuffers()** is called to display the updated frame, providing smooth double-buffered animation.*

*Frame updates are handled using **glutTimerFunc()** with a 16 ms interval (approximately 60 FPS). Each timer callback updates the positions, velocities, and lifespans of all active particles — including fireworks, ribbons, and balloons — and then triggers a redraw using **glutPostRedisplay()**. This mechanism ensures continuous animation, producing a smooth and stable frame rate for real-time visual effects.*

4.3 Geometric Transformations

*Various transformations such as translation, scaling, and rotation (**glTranslatef**, **glScalef**, **glRotatef**) are used to position and adjust scene elements like the Central Building outline, flowers, and arrows. These transformations allow flexible composition and ensure consistent visual layout across different resolutions and aspect ratios.*

5 Conclusion

This project presents an interactive 2D celebration card built with OpenGL and freeglut. By integrating visual design, animation, and interaction, it delivers an engaging experience that captures the spirit of XJTLU's 20th anniversary. The work demonstrates how graphical programming and creative design can unite to produce expressive and technically refined digital art. Overall, the project successfully integrates art and technology to celebrate XJTLU's 20-year journey."

[1] **XJTLU Visual Identity Guidelines**, Xi'an Jiaotong-Liverpool University, 2024.
Available: <https://www.xjtlu.edu.cn/zh/about/overview/xjtlu-identity>