CPT205 – Computer Graphics: Interactive Greeting Card with FreeGLUT

Xinrong Li (2363123)
Information and Computing Science

***Abstract*—This report documents the design, implementation and evaluation of a two-scene interactive greeting card written in modern C++ (C++17) using FreeGLUT and fixed-function OpenGL. It highlights the graphics techniques (gradient sky, layered clouds, parametric sun, polygonal landmark, wind-driven grass, balloons, and stroked vector text), the event-driven structure, and the runtime interactions. A concise README is included for compilation and usage.**

I. Introduction

Basic info. Module code CPT205, title Computer Graphics. Author: Xinrong Li (2363123), Programme: Information and Computing Science.

Card idea. The card presents a calm campus landscape that the user can interact with: clicking the background releases balloons; clicking the paper enters a *congratulatory* scene that displays formatted stroked text. The program runs with an orthographic 2D camera for crisp resolution-independent vector rendering.

Feature list (techniques):

- Gradient sky with multi-layer drifted clouds (parallax).
- Parametric sun with rotating rays and pulsation.
- Landmark building drawn from polygonal facets with seam sealing.
- Wind-swaying grass patches with per-blade segmentation.
- Interactive balloons with vertical rise and off-screen culling.
- Clickable paper/card (AABB hit test) and a formatted text poster using GLUT stroke fonts.
- Two scene modes: `BG_ONLY` and `CONGETS`; 60 Hz timer-driven redraw.

Runtime controls: `q` Quit; `r` Reset clouds; L/R mouse: interact (see Sec. III-C, IV).

II. System Design and Structure

A. Program Architecture

The program follows a clean object-centric layering: low-level *flat* drawing helpers (rect fill/stroke, strokes text), a set of drawable entities (*Sky*, *Sun*, *CentralBuilding*, *GrassField*/*GrassPatch*, *Tree*, *Balloon*, *Paper*, *PaperWithText*, *TextLine*), and a light scene controller that calls each entity's `render()` in the `display` callback. Window size is clamped and an orthographic projection is installed once in `reshape`.

B. Mathematical Model for Transformations

We use an orthographic projection `glOrtho(0,W,0,H,-1,1)`, mapping model coordinates directly into window pixels. For time-dependent animation (with time t in seconds):

- **Sun Rays.** For N rays with base radius R and rotation speed ω, the i-th ray direction angle is $\theta_i(t) = \frac{2\pi i}{N} + \omega t$, and its tip lies at $\mathbf{p}_i(t) = \mathbf{c} + (R + A\sin(2\pi f t))[\cos\theta_i, \sin\theta_i]$, giving combined rotation and pulsation.
- **Cloud Drift (Parallax).** For each layer ℓ with speed s_ℓ and wrap width W, cloud k has $x_{k,\ell}(t) = x_{k,\ell}(0) + s_\ell t \bmod W$; higher layers use larger s_ℓ to create depth.
- **Grass Sway.** Each blade is a polyline with S segments. Lateral offset uses a phase-shifted sinusoid: $\Delta x(y,t) = A\sin(ky + \omega t + \phi)$, producing wind-like motion while keeping the base anchored.
- **Balloon Motion.** Balloons rise as $y(t) = y_0 + vt$ with a gentle horizontal oscillation $x(t) = x_0 + a\sin(\Omega t + \psi)$; when y exceeds the screen, they are culled.
- **Paper Hit-Testing.** Axis-aligned rectangle test: a click at (x, y) hits the paper if $x \in [x_0, x_0 + w],\ y \in [y_0, y_0 + h]$.

C. Event-Driven Structure

We register FreeGLUT callbacks: `display` (draw), `reshape` (install orthographic view and clamp size), `keyboard` (hotkeys), `mouse` (click interactions), and a periodic `timer` to step the global time $t \leftarrow t + \Delta t$ (nominal $\Delta t = 0.016\,\text{s}$). The scene state machine toggles between `BG_ONLY` and `CONGETS` depending on click targets.

III. Scene Design and Animation Logic

A. Scene Composition

The scene layers, back-to-front: **Sky** → **Sun** → **Ground** → **Central Building** → **Trees** → **Grass Field** → **Paper** → **Balloons** → **Text Bar** → (optional) **PaperWithText** in `CONGETS`.

B. Animation Logic by Element

For each element we summarise: (1) description & animation, (2) algorithm, (3) demo picture.

Sky & Clouds: (1) Gradient sky with three drifting cloud layers (left-to-right). (2) Layered wrap-around drift with parallax speeds; clouds are billboard ovals composing soft clusters. (3) The implementation is shown in Fig. 1.

Fig. 1. Sky with multilayer clouds.

Sun: (1) Radiant sun that rotates and breathes. (2) N equally spaced rays with rotating angle and sinusoidal length modulation (see Sec. II-B). (3) The parametric sun design is illustrated in Fig. 2.

Fig. 2. Parametric sun.

Ground: (1) A soft green strip establishes the lawn baseline. (2) Single quad spanning the bottom 140 px; draws before mid-ground geometry. (3) The ground strip rendering is shown in Fig. 3.

Fig. 3. Ground strip.

Central Building: (1) Landmark composed of polygonal parts; seams are *sealed* by over-stroking selected edges. (2) Triangulated facets in model space scaled to window; a seam list is stroked with a few pixels thickness to hide sampling gaps. (3) The polygonal building structure is depicted in Fig. 4.

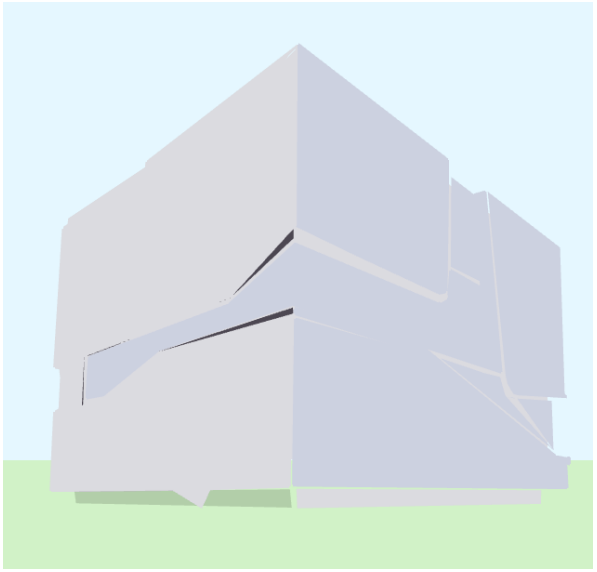

Fig. 4. Central building polygons.

Trees: (1) A small forest of simple shapes in mid-ground. (2) Instanced draw with per-tree transforms (position/scale), ordered back-to-front for correct overlap. (3) The three different tree styles are presented in Fig. 5.

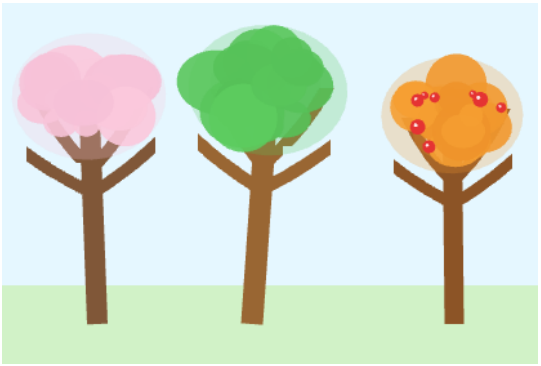

Fig. 5. Trees.

Grass Field: (1) One or more animated grass patches with wind sway. (2) Each patch owns B blades, each a S-segment polyline perturbed by $\Delta x(y, t)$ (Sec. II-B). Parameters include amplitude, speed, direction, and color. (3) The wind-swaying grass implementation is demonstrated in Fig. 6.

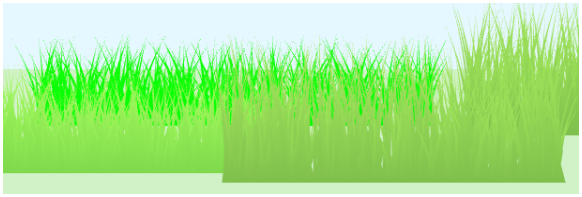

Fig. 6. Grass patches.

***Paper**:* (1) A clickable, rounded-rect card in the center foreground; entering `CONGETS` is triggered by clicking inside it. (2) Drawn with fill + thin border; AABB hit test $(x, y) \in [x_0, x_0 + w] \times [y_0, y_0 + h]$. (3) The clickable paper element is shown in Fig. 7.

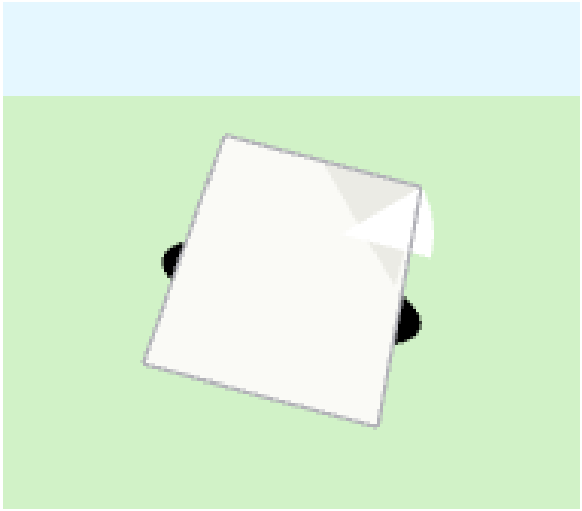

Fig. 7. Clickable paper.

***Balloons**:* (1) Random balloons rise from clicks on the background; off-screen balloons are removed. (2) One-way vertical motion with gentle lateral oscillation and a life-state (*alive*/culled). (3) The balloon animation effect is illustrated in Fig. 8.

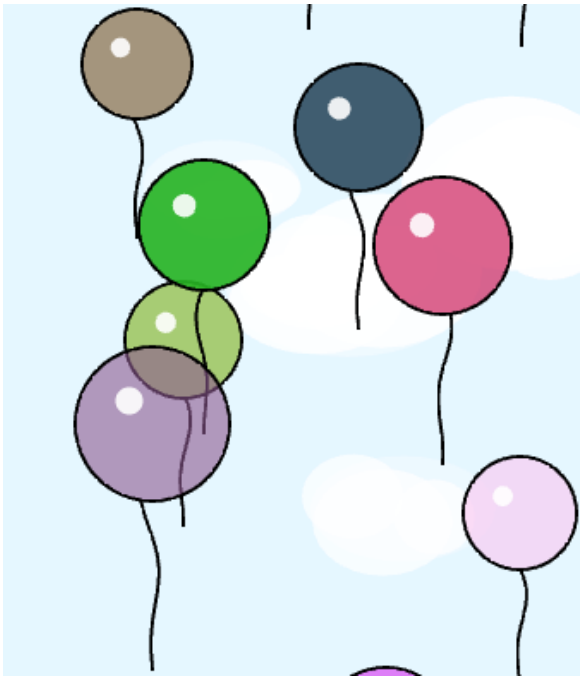

Fig. 8. Balloons.

***Text Bar**:* (1) A bottom banner showing simple stroked text; updates every frame. (2) Uses GLUT stroke fonts with per-frame color/time modulation. (3) The text bar rendering is displayed in Fig. 9.

Fig. 9. Text bar.

***PaperWithText** (CONGETS only):* (1) A formatted poster with a headline and body text; click outside to return. (2) Typeset by computing margins, line width, and stroke scaling; optional ribbon can overlay. (3) The CONGETS scene layout is presented in Fig. 10.

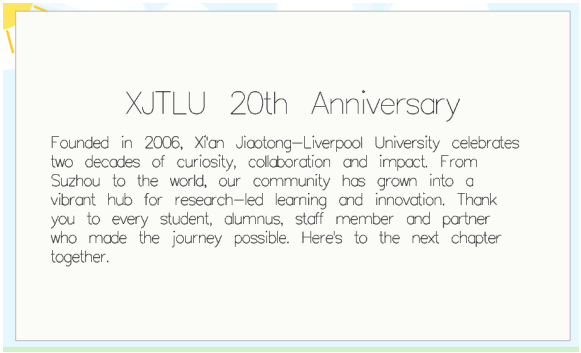

Fig. 10. PaperWithText scene.

C. Keyboard and Mouse Controls

Description: Keyboard hotkeys provide program-level actions; mouse clicks drive scene transitions and spawn balloons. Left or right click on the background creates a balloon; clicking the paper toggles into the text scene; clicking anywhere *outside* the poster returns to the background.

TABLE I
KEYBOARD & MOUSE INTERACTION SUMMARY

Input	Context	Effect
`q`	Global	Quit the program.
`r`	Global	Reset/reseed the clouds.
L/R Click (background)	`BG_ONLY`	Spawn a rising balloon at the click x.
L/R Click (paper)	`BG_ONLY`	Enter `CONGETS` (show poster).
L/R Click (outside poster)	`CONGETS`	Return to `BG_ONLY`.

IV. Runtime Demonstration

A. BG_ONLY Mode

(a) Description and Animation Logic: The background scene continuously animates the sky, sun, trees and grass. The paper remains clickable. User clicks on the background generate balloons that rise and get culled when exiting the screen. The bottom text bar renders every frame.

TABLE II
TYPICAL INTERACTIONS IN BG_ONLY

Action	Observed Behaviour
Idle	Clouds drift; sun rotates/pulses; grass sways; text bar updates.
Click background	A new balloon appears and starts rising; stale balloons are culled.
Click paper	Transition to CONGETS; poster is shown.
Press r	Clouds reset positions/seeds.
Press q	Program exits.

Fig. 11. BG_ONLY mode overview.

B. CONGETS Mode

(a) Description and Animation Logic: In CONGETS mode, the formatted poster with congratulatory text is displayed over the background. The poster contains a headline and body text rendered using GLUT stroke fonts. Users can click anywhere outside the poster to return to BG_ONLY mode. The background animation continues to run underneath the poster overlay.

Fig. 12. CONGETS mode demonstration with formatted poster.

V. PERFORMANCE ANALYSIS AND DISCUSSIONS

A. Frame Rate and Timing

Rendering is timer-driven at 16 ms per tick ($\approx$ 60 Hz). Fixed-function draws (quads/lines/polylines) are lightweight; the primary costs scale with the number of blades, balloons and clouds. The per-frame complexity is roughly linear in entity counts $O(B + N_{\text{balloons}} + N_{\text{clouds}})$. Double buffering and multisampling improve visual quality at negligible CPU overhead on modern machines.

B. Visual Quality vs. Cost

Anti-aliased stroke text and multisampling produce clean edges. Grass sway and cloud parallax add depth while keeping geometry simple. Seam sealing on the building hides polygon cracks at scale 1.0, avoiding z-fighting or sampling gaps.

C. Numerical Stability

Time is accumulated in a single precision float; for long sessions, periodic re-basing (e.g., subtracting an integer number of seconds) can reduce drift. Trigonometric updates remain stable at the target rate.

D. Profiling & Optimisation Ideas

Batching cloud sprites per layer and limiting maximum alive balloons cap worst-case cost. If needed, grass segments can be reduced on low-end machines. Future work could port hot paths to VBO/modern OpenGL for further efficiency while preserving the current API.

E. Maintainability and Extensibility

The code separates concerns: entities encapsulate state and a `render` method; the scene controller orchestrates order and interaction; callbacks are short and declarative. New elements (e.g., fireworks) can be added as another drawable with minimal coupling. Configuration structs (colors, speeds, counts) make it straightforward to tune visuals without touching logic.

VI. CONCLUSION

We presented a compact, maintainable, and visually pleasing interactive card with a clear event-driven architecture. The scene layers, parametric animations and simple interactions achieve an engaging effect while keeping the implementation approachable and extensible.