

# CPT 205 Assessment1

## A two-dimensional(2D) card of birthday party

Module	CPT205
Module Name	Computer Graphics
Degree Programme	Information and Computing Science
Name	Yichuan Zhang
Student ID	2251668

## 1. Design Description

The birthday card is divided into two halves. The envelope is the initial section. The main card is the other. Users can use the keyboard to open the envelope. The primary card will be visible once the envelope has been opened. By clicking the mouse, the users can also experience lighting up matches for the celebration.

## 2. Feature

### Window Set

Use glutOrtho2D to display the card with a window size of (1280, 720). The position of the card depends on the user windows. It always lies in the middle of the screen. The coordinate of the card used in the OpenGL program is 1600 \* 900. The name of the window is "Happy birthday!".

### Envelope Background

The background of the envelope is drawn in two parts. The upper one will be a flappable triangle implemented by the (**GL\_TRIANGLES**). The lower one is a (**GL\_POLYGON**).



Figure 1. Envelope

### Envelope Boarder

The border of the envelope consists of many. Parallelogram with the color of light blue and pink.

They are drawn by (**GL\_POLYGON**)



Figure 2. Envelope Boarder

### Twisted Ribbon

Twisted Ribbon lies between the upper and lower envelopes. The whole twisted ribbon consists of two parts. The nature of the first part of the object is a line of points drawn by (**GL\_POINTS**) and **sin** function. The other part of the object is similar, but the trace follows the change of the **cos** function. To make sure the ribbon is visible, the size of the point is set to 10 using **glPointSize**. Then the two twisted crossed line will be moved to the target positions using **glTranslatef** and **glRotate**. There are two twisted ribbon to cover the upper end of the envelope.

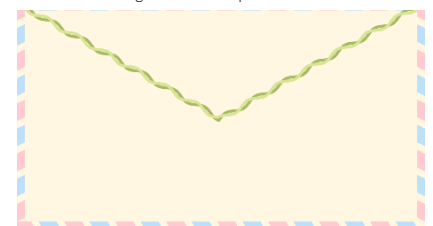


Figure 3. Twisted Ribbon

## Sealing Wax

The petal shape is created using the **parametric equation** of a flower shape and implemented with OpenGL, applying `glShadeModel(GL_FLAT)` to produce a gradient effect. A yellow stamen is then drawn using `(GL_LINE_LOOP)`. The logic is written in a loop to generate 6 parts to simulate the flower.

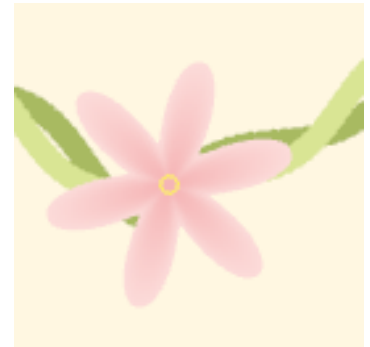


Figure 4. Sealing Wax

## Bears

The fluffy bears at the bottom of this envelope are primarily constructed using simulated circle drawn by `(GL_TRIANGLE_FAN)` in OpenGL and carefully designed relative coordinates. Among these bears, the semicircular shape of their ears is achieved by masking with an upper circular layer. The eyes are black ovals in fixed positions, drawn by `(GL_POLYGON)`, and the bear's arms and mouth are drawn using a **third-order Bézier curve**.

Functions like

```
glMap1f(GL_MAP1_VERTEX_3, 0.0, 1.0, 3, 3, &arm_right[0][0]);  
glEnable(GL_MAP1_VERTEX_3);  
glBegin(GL_LINE_STRIP); are used to draw the curve,
```



Figure 5. Bears

## Balloon

The balloon consists of three main parts: the balloon itself, the balloon valve, and a string connecting the bear to the valve. The balloon itself is created using `(GL_POLYGON)` and shaped with a heart-shaped function. It contains two layers of balloons rotated using `glRotatef`. The balloon valve is a triangle drawn with `(GL_TRIANGLES)`. The string connecting the bear, and the valve is a **third-order Bézier curve**.

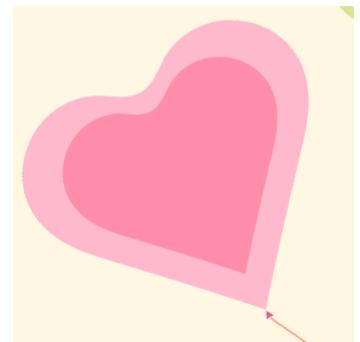


Figure 6. Balloon

## Birthday Cake

The cake is divided into two layers. Layers are drawn in two ellipses and a rectangle between them. The ellipses are created with `(GL_POLYGON)`, while the rectangle uses `(GL_QUADS)`. Then, `(GL_LINE_LOOP)` is used to add an outline.

The cream layer is created by using `(GL_POLYGON)` to draw white ellipses evenly spaced around the cake layer. The candles follow a similar logic to the cake layer's cylinder. The cake's frosting is drawn with triangles using `(GL_TRIANGLES)` and achieves a gradient effect through `(GL_BLEND)` mode. Each triangle has precise coordinates, and they are drawn one by one to complete the effect.



Figure 7. Birthday Cake

## Fire

The flame is the most striking part of this project. Each flame particle is represented as a rectangle drawn with **(GL\_QUADS)**. The effect of the fire is achieved by the particle system.



Figure 8. Fire

## Text Box

This text box is created by drawing three nested rectangles using **(GL\_QUADS)**. Each rectangle's parameters are calculated through a specific formula. In the top-right and bottom-left corners, there are two rectangles colored with **glColor3f**. The heart in the bottom-right corner of the text box is drawn using the same method as the balloon on the envelope.

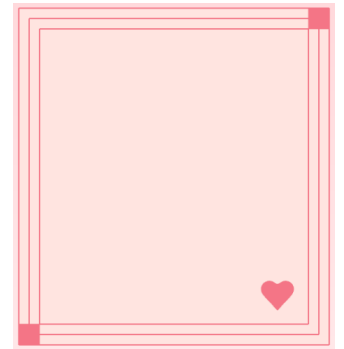


Figure 9. Text Box

## Confetti

The sticker confetti consists of two parts: the Party Popper and the confetti effect. Party Popper is created by repeatedly drawing multiple triangles with **(GL\_TRIANGLES)** at intervals and using **(GL\_LINE\_LOOP)** for outlining. The confetti effect is generated through a particle system, which randomly creates points with varying colors at specific angles using **(GL\_POINT\_SMOOTH)** and **(GL\_POINTS)**.



Figure 10. Confetti

## Fireworks

The fireworks are drawn using **(GL\_TRIANGLE\_FAN)** to create a 20-sided triangular fan, with **glRotatef** used to adjust the position of the flames and **glScalef** to control the generation and fading of the flames.

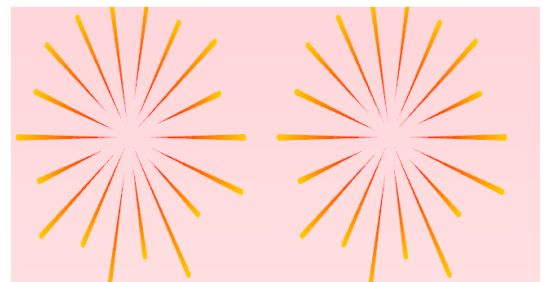


Figure 11. Fireworks

## Letters

The text is handled using `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`, the `HFONT` class, and the `HDC class`, with the font set to Lucida Handwriting. The font for hints is Times New Roman.

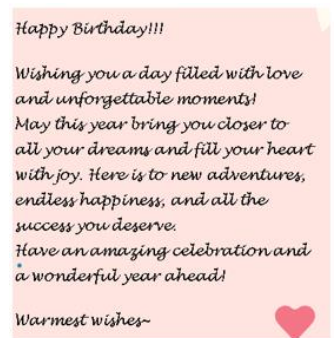


Figure 12. Letters

## Plane

The plane is created using `(GL_TRIANGLES)` and `(GL_POLYGON)`, and then `glTranslatef`, `glRotatef`, and `glScalef` are applied to assemble the overall plane structure.



Figure 13. Plane

## 3. Instruction

### Keyboard interaction

#### At envelope:

Press 'o' or 'O' to open the envelope.

#### At letter:

Press 'q' or 'Q' to quit the card.

### Mouse interaction

#### At letter:

Click left to light up the candle one by one.

## 4. Screenshots play

