



CPT205 ASSESSMENT 1 REPORT

2D MODELLING PROJECT

MINGYUAN LI(2145618)

ICS

NOVEMBER 5, 2023

1 Introduction

1.1 Task Overview

The assignment is to design a 2D 2024 XJTLU Graduation Ceremony invitation card with important university symbols, balloons, flowers, and more. The card should have both animated and static elements, and actions should be triggered by keyboard and mouse inputs. By using OpenGL only with the freglut library, achieving a high-quality visual effect using techniques like geometry, transformations, animation, and so on.

1.2 Project Overview

To achieve the goal of the tasks, the design of the card should be layered and clear, and the OpenGL methodology used should be comprehensive and fit the object's characteristics [1]. Thus, the complete card consists of 3 layers and 1 main object. The background layer contains a blue sky and green land; the environment layer contains the outline of the XJTLU logo and Central Building (CB); the object layer contains graduation ceremony stuff like sun, clouds, colorful balloons, and a bachelor's or master's hat requiring tassels. Only objects in the object layer will have animation effects: balloons will wiggle, clouds will move, and the sun will rotate. Through corresponding keyboard or mouse interactions, balloons can be raised and hats can be tasseled.

2 Software Environment Configuration

The software environment strictly conforms to MS VC++ and OpenGL environments as lab1 defined. And whole project only uses the freglut library and basic C++ libraries.

3 Design and Implementation

This section is for illustrating the design purpose and summary of OpenGL techniques used. The following are about the design rationale and approach.

1. Theme and elements

The graduation ceremony is usually accompanied by an upbeat atmosphere and takes place on the XJTLU campus, thus the card theme must include blue sky, green field, and the XJTLU campus. Meanwhile, to emphasize the cheerfulness of the graduation, the bachelor's or master's cap and the celebratory balloons are also taken into consideration.

2. Layer classification

After gathering elements, the graphic layer should be defined so the main objects can be highlighted. To make up the background, the sky and land field divide the card into two parts, and the horizon is created. To highlight the XJTLU environment, using

lines to depict the outline of XJTLU text and CB is worthwhile. Thus, those remaining objects are in the object layer to make the card less monotonous. It's reasonable to choose the hat as the main object because every graduate will wear it.

3. Interaction and animation

To make the object more live, adding interactions and animations is necessary. For the hat, the graduation ceremony has graduates and undergraduates, so the tassel should have 3 colors and can be switched by the keyboard. In addition, dialing the tassel is an important process, so the design of the animation of dialing the tassel is valuable. For the balloons, there are some balloons already lifted, so they should wiggle. While user can use the keyboard to release more balloons, which should have the same effect when lifted. For the sun should keep rotating, the clouds should keep moving and new clouds should enter after moving out of the screen.

4 Code Implementation and Graphic Techniques Details

MultinomialNB

This section explains the OpenGL code implementation and design of each object in detail. To make the structure clear and easy to understand, each object is explained as a unit. This helps to clarify the purpose and design logic behind its use. It explains objects in the same order as they appear in the code.

4.1 Code Framework and background setting

The code framework is divided into 4 sections: global variable declarations, reuse method declarations, display method declarations, and timer method declarations.

The window width and height are set to 1000 and 600. Using `glMatrixMode`, `gluOrtho2D` to set the OpenGL projection matrix that defines the extent of the visible area on the screen. Next, use `glScissor` to set the specified clipping area, making sure that only pixels within the clipping area are rendered in the colors of the sky and the green field. Finally, using `GL_LINES` draw a white horizontal line at the border of the two areas to clearly distinguish between the areas.

4.2 Title Text

- Custom method: `drawText`
- OpenGL method: `glutBitmapCharacter`

Even though there will be some graduate features, it is still necessary to draw a caption telling the reader that this is about the 2024 graduation ceremony. So by using the `glutBitmapCharacter`, there is a title in the bottom left.

4.3 The CB

- OpenGL method: `GL_LINES`, `GL_LINE_LOOP`

To create a CB outline graphic, it is wise to use `GL_LINE_LOOP` to create a closed outline and `GL_LINES` to create the line connecting the closed outline. After iterations of adjusting the endpoints of the outline, the CB was successfully created to be placed in the bottom right corner of the card.

4.4 XJTLU Logo

- OpenGL method: `GL_LINES`, `GL_LINE_LOOP`, `GL_LINE_STRIP`

In order to create the outline of the XJTLU logo, just use a way similar to creating the CB outline except that J and U because these two letters have curves. To create curves that can represent the letters, it can use for loop and trigonometry to calculate the coordinates of the curve, then use `GL_LINE_STRIP` to connect the points to form the curve. It also makes sense to change the width of the line to 3 and color it the same color as XJTLU to emphasize the logo.

4.5 The Clouds

- Custom method: `drawCloud`
- OpenGL method: `glPushMatrix`, `glPopMatrix`, `glTranslatef`, `glScalef`

A cloud consists of 3 ellipses with different orientations, so a customized method should be created to be called multiple times. The ellipse is created in a similar way to a curve, still calculating the coordinates of the points by angle and connecting them together. Rotate the two ellipses by 15 and 95 degrees and overlap them to form a cloud.

To make the cloud move, we can change the x-axis coordinates of the cloud and use the timer mechanism to render it repeatedly. When it crosses the window, we need to reset the coordinates, so it can move again.

To make the sky more realistic, it makes sense to add an inverse cloud of different sizes and speeds, which can be quickly resized by pairing `glScalef` with `glTranslatef`, which saves the time of coding.

4.6 The Hat

- OpenGL method: `glPushMatrix`, `glPopMatrix`, `glTranslatef`, `glShadeModel`, `glRotatef`

The bachelor's hat consists of a rhombus, two rectangles, and a line, where the line and the thin rectangle form the shape of a tassel. To make the hat more realistic, use `GL_SMOOTH` to simulate the feel of the light. Thus the top left corner is light black and the bottom right corner is dark black.

For the colors of the tassels, we should use black, blue, and red to represent undergraduates, masters, and PhDs.

Therefore, it can define a matrix variable to represent the color matrix and design a keyboard processing function to change the matrix number by typing the E key to change the color.

Considering that tasseling is usually done when standing, it is feasible to press the left mouse button to raise the hat straight up. For the tassel operation, the line and tassel should rotate around the center of the hat, but the tassel isn't always facing down vertically which ruins the realism. So define a temporary variable to record the angle of each rotation, and rotate the tassel the same way in reverse to simulate the feeling of gravity.

4.7 The Sun

- Custom method: `drawText`
- OpenGL method: `glutBitmapCharacter`

The sun consists of a circle and a closed line. The circle is created by `GL_TRIANGLE_FAN` in the same way as the curve, and the regular closed line is created in a similar technique to simulate the sun's rays. By using the `glRotatef` and `updateSun` functions, the regular closed line could rotate continuously to simulate real sunlight.

The sun is a circle and a closed line. To create the circle, `GL_TRIANGLE_FAN` is used to simulate the curve of the sun. To create the regular closed line, similar techniques are used to represent the rays of the sun. To simulate continuous sunlight, the regular closed line can be rotated using `glRotatef` and the `updateSun` function.

4.8 The Balloons

- Custom method: `drawText`
- OpenGL method: `glutBitmapCharacter`

A balloon consists of a circle with a line, and since multiple balloons need to be generated, it is reasonable to define it as a customized method. To simulate a balloon that is round at the top and elongated at the bottom, it can set a point 70 down from the center of the circle to create a fan shape that simulates the lower part that is not round.

Create colorful balloons suspended in the air, and make them move back and forth horizontally by using `updatingBallonX`. Of course, we should also have balloons on the ground waiting to be released by the user, touching the F key to release the balloon, and when it floating in the air, it should also have the feature of movement.

5 Program Execution and Results

As described in the previous section, there are several keyboard and mouse interactions in the card, and this section is designed to describe them and show the corresponding pictures.

The following image shows the initial stage:

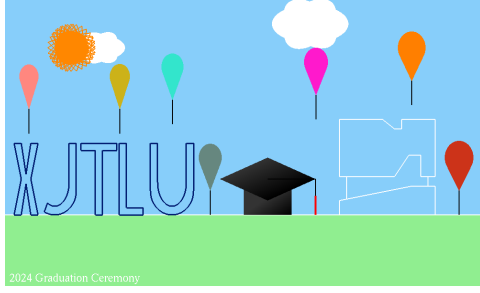


Figure 1: Initial Stage

Stage 1: By clicking the left mouse button, the user can raise the hat to the specified height. The following image shows the stage 1:

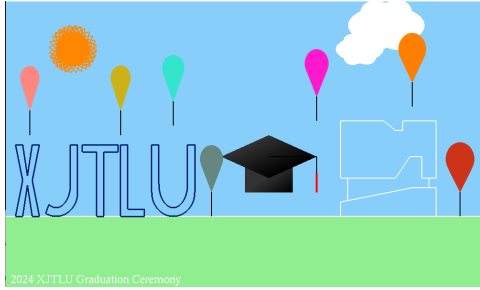


Figure 2: Stage 1

Stage 2: By pressing the E key on the keyboard, the color of the tassel can be switched (black-blue-red). The following image shows the stage 2 (blue):

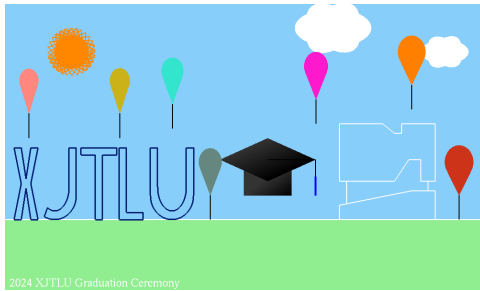
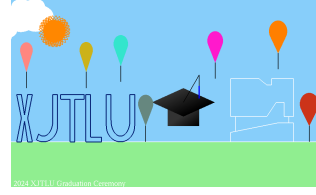


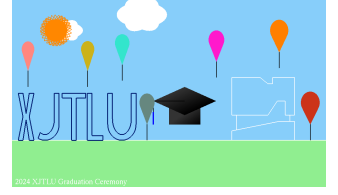
Figure 3: Stage 2

Stage 3: By pressing the R key on the keyboard, the position of the tassel can be changed, 30 degrees at a time, until it is 180 degrees (on the other side). The following images show the stage 3:

Stage 4: By clicking the left mouse button, the user can release the balloons on the ground. The following image shows the stage 4:



(a) Stage 3 part1 (processing)



(b) Stage 3 part2

Figure 4: Stage 3

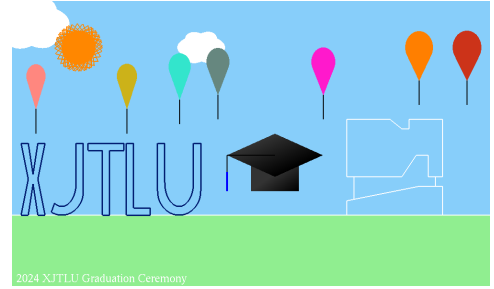


Figure 5: Stage 4

6 Conclusion

By using multiple OpenGL methods, the project successfully produced a 2024 graduation card and met the task requirements. The card can achieve those goals through the construction of geometry, keyboard and mouse interactions, viewing, transformations, and animation. Three layers are created: the background layer, the environment layer, and the object layer. One main object is created: the hat that represents graduation. Create keyboard and mouse actions to highlight key objects and convey an exciting graduation atmosphere.

References

- [1] Microsoft. (Access2023-11-05) OpenGL functions. [Online]. Available: <https://learn.microsoft.com/zh-cn/windows/win32/opengl/gl-functions>