Xi'an Jiaotong-Liverpool University
西交利物浦大学

University of Xi'an Jiaotong-Liverpool

School of Advanced Technology

CPT205 Assessment 1 Report

ID:	*2034675*
Name:	Kaijie Lai
Email Address:	Kaijie.Lai20@student.xjtlu.edu.cn

Nov. 6, 2022

Introduction

CPT205 Assessment 1 is focusing on 2-dimensional modeling, which calls for a combination of theoretical understanding from the first half of the semester, including concepts like polygons, viewing, transformation, etc., and also programming expertise in OpenGL and C++. The assignment's objective is to use a mix of the aforementioned knowledge to make a New Year’s Greeting Card to celebrate. The main idea of this piece is to illustrate how traditional New Year's celebrations have fallen out of favor because people have started to celebrate the New Year more on their electronic devices, like being stuck in a “Pixel Maze”, however, the world behind the phone is often more exciting.

To achieve greater efficiency and manageability of the code, the code project is object-oriented programming with dynamic container management and specific comments for easy reading of the code.

Instruction

1. After entering the debugging scene or running the scene, you can find a nine-box maze in the middle of the screen which is disordered and we need to recover the order to enter the next stage.
2. For interaction, you can use the mouse and keyboard to trigger the corresponding effects. First, a **mouse-left-click** on a cube will turn the selected cube red to indicate that it is being selected. Secondly, you can use the **“w” “s” “a” “d” keys** to move your selected move up, down, left, and right (in the English input method).
3. If you feel like moving around too much and want to restart the puzzle again, you can click the **restart button** or hit **“r”** on the keyboard.
4. At the same time, there are tips in the console command window throughout to help you solve the puzzles. If that's too much trouble for you, I've also listed the steps for moving the specific letters cubes here:

   “R” (→), “PY” (↑), “ER” (←), “R” (↓), “PY” (→), “EW” (↓), “AP” (←), “PY” (↑)
5. When you move all the cubes into the correct position, New Year's fireworks will be triggered automatically. By the way, you can also use **mouse-right-click** anywhere in the window to trigger one firework with one click.

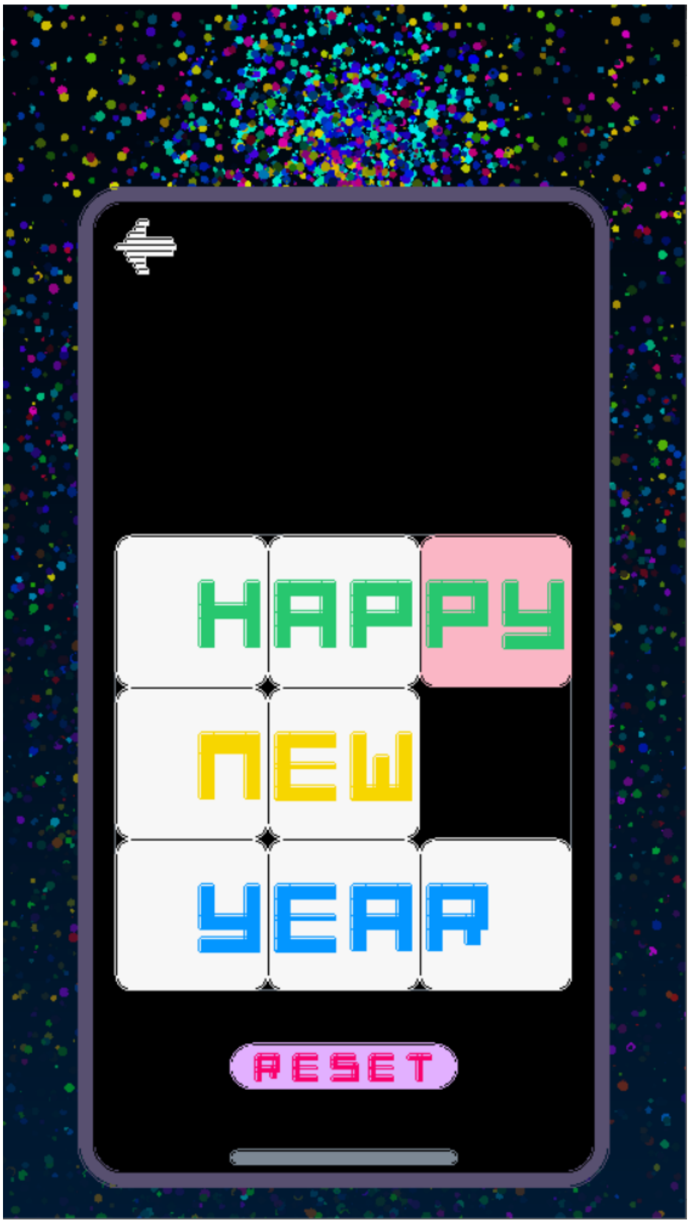

figure 1 Final scene of the program

Base Object

A base object is a basic unit consisting of a round rectangle and ball to generate more complex elements like the fireworks and all the elements in the phone scene. As a result, it should be given such properties like:

- Center coordinates
- Color (in HSV color space)
- Luminance (opacity between 0 and 1)
- Velocity (x and y direction) (Pixel / s)

The reason to choose HSV (Hue, Saturation, Value) color space which is more closely aligned with the way human vision perceives color-making attributes instead of RGB color space is for modifying the color more easily. So, all the units generated by the Base object store values in struct `HSV_t` and use function `hsv2rgb` to RGB values and stores in struct `RGB_t` to be used when elements need to be created.

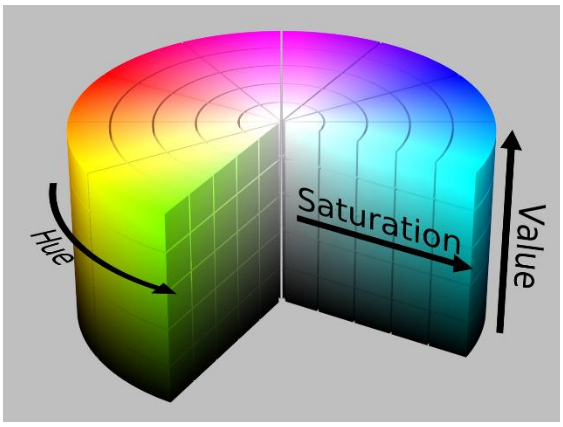

figure 2 HSV Color Space

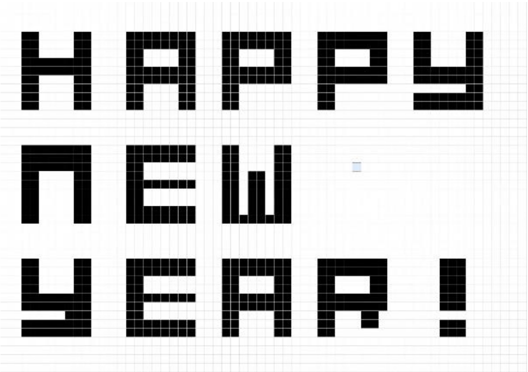

figure 3 Original Design Alphabet

One of the most important units in Base Object is a round rectangle which creates an elegant round rectangle instead normal square. In functions `RoundRec` and `FillRoundRec`, with attributes `Width`, `Height,` and `CirR` (stands the edge circle of a round rectangle), I choose to draw a round rectangle with a line first and fill the color inside after then. In this way, we can control how to draw a perfect rectangle only by OpenGL. By the way, all of the visual-letters in this Window are designed by an Alphabet which is constructed by round rectangles.

Fireworks Object

A fireworks explosion is created by creating a large number of balls in base objects (typically 360) with similar color hues and starting positions, but with many random properties. The following are the properties of the small balls it generates and how it applies the random function:

- **Initial Positions**: Same point.
- **Initial Direction of Velocity**: Equidistant distribution around 360°.
- **Initial Magnitude of Velocity:** Random between [0 ~ `maxSpeed` of firework].
- **Color:** `centerHue` (in degree) plus random between -10 and 10 degrees.
- **Luminance Decay Factor:** Random between 0.2 and 0.8.

Each firework has a maximum speed and a center hue. The firework's properties are also randomly chosen when it generates a new spark. Because there is air resistance, the maximum speed of the firework determines its size of the firework. This is the same as in reality. Each ball in the firework has a different color, and the color of each ball is randomly chosen according to the firework's theme color. Each firework object has a vector to store the balls. If the brightness of a ball is too low, it is marked with a lazy delete tag and the ball is not drawn even if it is hit. We maintain a list of fireworks objects, and whenever we add a new fireworks object, we first check to make sure it is visible. If it isn't, we add it to the list and continue traversing it. Finally, if the fireworks object is visible, we delete it from the list.

GlobalState Object

In this section, the choice is made to use a vector to manage the implementation of all change variables in the global state. In this object, the mapping of the mouse and keyboard is set up and managed, where two functions are used to implement the conversion between the World Coordinate system, the Distribution Matrix, and the corresponding pattern distribution matrix.

Here is the end of this report.