

國立臺南大學資訊工程學系

## **Computer Graphics**

### 第二次作業

#### **題目 : 3D Graphics Input and Interaction**

班級 : 資工三

姓名 : 卓楚庭

學號 : S11159020

## 目錄

● 作業說明-----	P.3
● 程式設計環境架構-----	P.4
● Function 展示-----	P.6
● 執行成果展示-----	P.9
● 心得-----	P.10
● 參考資料-----	P.10

# 作業說明

## 1. 題目：

實作 Sierpinski Gasket in a Tetrahedron

## 2. Requirements :

- 3D Gasket – Regular tetrahedron with volume subdivision
- Input Devices – Mouse
  - Trigger the menu by pressing the right mouse button
- Input Devices – Keyboard
  - Press the 'q' or 'Q' to quit the program
- Pop Menu – Make a submenu to select and change the subdivision level of the displaying tetrahedron, and exit the program in the main menu
- Initial Subdivision Level = 0
- The window title is your Student ID

# 程式設計環境架構

## 1. 程式語言

C++ in MS Windows 11

## 2. 程式開發工具

Microsoft Visual Studio

## 3. 電腦硬體

CPU: Intel i5-1135G7 ,

Main Memory: 16GB LPDDR4X,

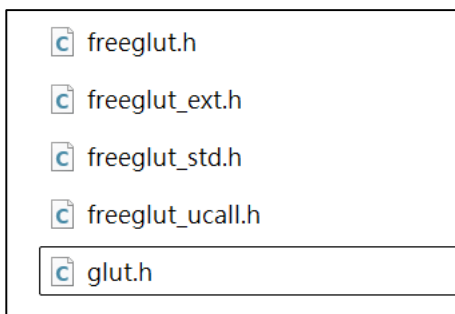
SSD: PCIe 512GB

## 4. 編譯器




```
g++.exe (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0  
Copyright (C) 2018 Free Software Foundation, Inc.
```

## 5. GLUT 函式庫名稱與版本

Freeglut 3.6.0



OpenGL 3.2

✓ 上個月	
 glut.dll	2024/10/20 14:29
 glut.h	2024/10/20 14:29
 glut.lib	2024/10/20 14:29
 glut32.dll	2024/10/20 14:29
 glut32.lib	2024/10/20 14:29

# Function 展示

## 1. Draw a tetrahedron

三角錐中共有 4 個點形成 6 條邊，利用二維陣列存取 6 條邊個別的中點座標位置(x, y, z)。以找出每條邊之中點的方式切割圖形，將切割出的中點座標位置存進 mid 陣列之中，以 Recursion 的方式切割直到 Base case ( $n = 0$ )，當  $n = 0$  時則直接顯示未切割過的 tetrahedron。

```
27 // Recursive function to subdivide the tetrahedron
28 void divide_tetra(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, int m) {
29     // triangle subdivision using vertex numbers //
30     GLfloat mid[6][3]; // Array to store midpoints of edges
31     int j;
32
33     if (m > 0) {
34         for (j = 0; j < 3; j++) {
35             mid[0][j] = (a[j] + b[j]) / 2;
36             mid[1][j] = (a[j] + c[j]) / 2;
37             mid[2][j] = (a[j] + d[j]) / 2;
38             mid[3][j] = (b[j] + c[j]) / 2;
39             mid[4][j] = (c[j] + d[j]) / 2;
40             mid[5][j] = (b[j] + d[j]) / 2;
41         }
42
43         divide_tetra(a, mid[0], mid[1], mid[2], m - 1);
44         divide_tetra(mid[0], b, mid[3], mid[5], m - 1);
45         divide_tetra(mid[1], mid[3], c, mid[4], m - 1);
46         divide_tetra(mid[2], mid[5], mid[4], d, m - 1);
47     }
48     else {
49         tetra(a, b, c, d);
50         // draw triangle at end of recursion //
51     }
52 }
```

```
void triangle(GLfloat* a, GLfloat* b, GLfloat* c) {
    // display one triangle //
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(c);
}

// Function to draw a tetrahedron using four triangles with a different color
void tetra(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d) {
    glColor3f(1.0, 0.753, 0.796); //Pink
    triangle(a, b, c);
    glColor3f(0.486, 0.988, 0.0); //grass green
    triangle(a, c, d);
    glColor3f(0.678, 0.847, 0.902); //light blue
    triangle(a, d, b);
    glColor3f(1.0, 0.647, 0.0); //Orange
    triangle(b, d, c);
}
```

## 2. Create a Menu

Menu 是利用回傳 id 的方式判斷點選該欄位後需要執行的動作。先

Create 出各 Entry 的 Name 和預設回傳的 id 值，利用點選滑鼠右鍵呼叫出 menu，點選滑鼠左鍵則取消顯示 menu。

```
102 void createMenu() {
103     int submenu = glutCreateMenu(sub);
104     glutAddMenuEntry("0", 1);
105     glutAddMenuEntry("1", 2);
106     glutAddMenuEntry("2", 3);
107     glutAddMenuEntry("3", 4);
108
109     int menu = glutCreateMenu(main_menu);
110     glutAddSubMenu("Subdivision Level", submenu);
111     glutAddMenuEntry("Exit", 1);
112
113     glutAttachMenu(GLUT_RIGHT_BUTTON); // trigger the menu
114     glutDetachMenu(GLUT_LEFT_BUTTON); // leave the menu
115 }
```

Main menu 中若回傳的 id 為 1，則結束程式。

```
66 void main_menu(int id) {
67     if (id == 1) {
68         exit(0);
69     }
70 }
```

Submenu 中以回傳的 id 值決定新的 n 值(tetrahedron 要切割幾層)。

```
72 void sub(int id) {
73     switch (id) {
74     case 1:
75         n = 0;
76         break;
77     case 2:
78         n = 1;
79         break;
80     case 3:
81         n = 2;
82         break;
83     case 4:
84         n = 3;
85         break;
86     default:
87         n = 0;
88         break;
89     }
90 }
```

n 值被更新後再重新畫出新的三角錐。(這裡是將原本 display function 中的內容複製過來，重新繪製圖形)

```
91     // display()
92     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
93     glutSwapBuffers();
94     glBegin(GL_TRIANGLES);
95
96     divide_tetra(v[0], v[1], v[2], v[3], n);
97
98     glEnd();
99     glFlush();
100 }
```

### 3. Trigger by Keyboard

判斷鍵盤回傳的值，若按下‘Q’或‘q’會結束程式，按下其他按鍵則不做任何動作。

```
void key(unsigned char k, int x, int y) { // keyboard trigger
    switch (k) {
        case 'Q':
            exit(0);
            break;
        case 'q':
            exit(0);
            break;
        default:
            break;
    }
}
```

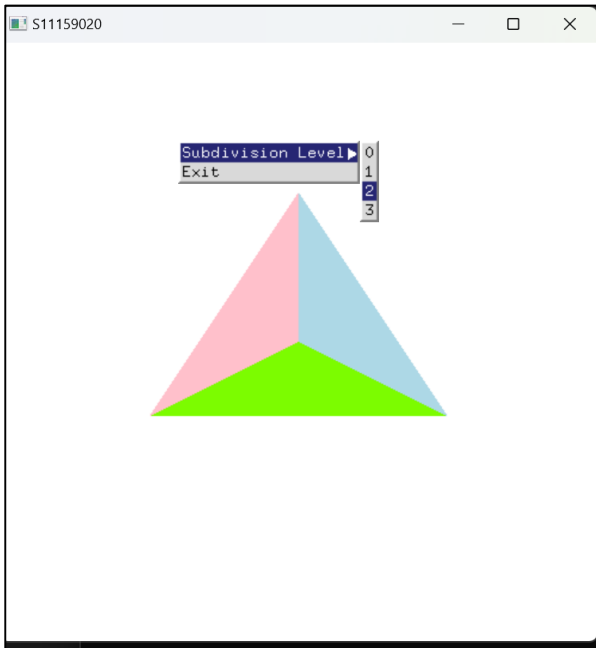
### 4. Main Function

```
156 int main(int argc, char** argv) {
157
158     glutInit(&argc, argv);
159     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
160     glutInitWindowSize(500, 500);
161     glutInitWindowPosition(0, 0);
162     glutCreateWindow("S11159020");
163
164     createMenu();
165
166     glutKeyboardFunc(key);
167     glutReshapeFunc(Reshape);
168     glutDisplayFunc(display);
169
170     init();
171
172     glutMainLoop();
173     return 0;
174 }
```

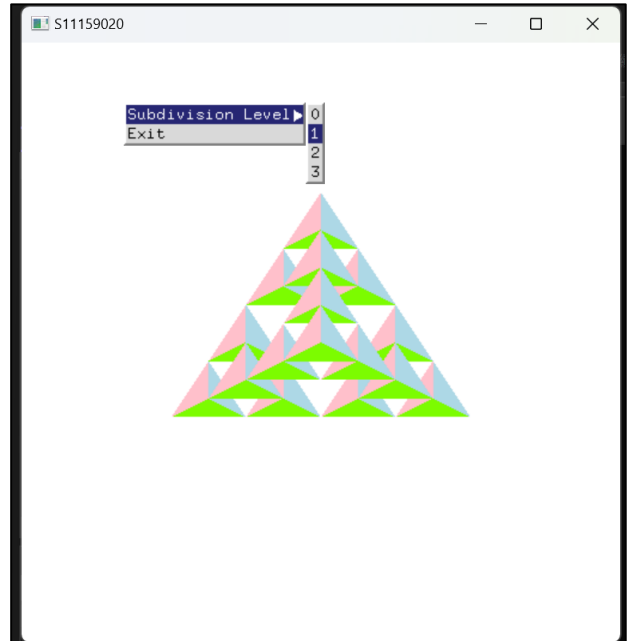


## 執行成果展示

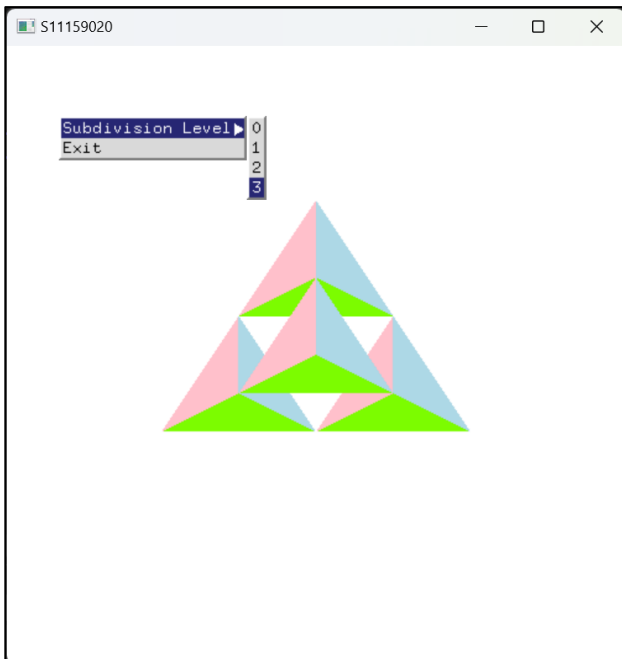
Level = 0



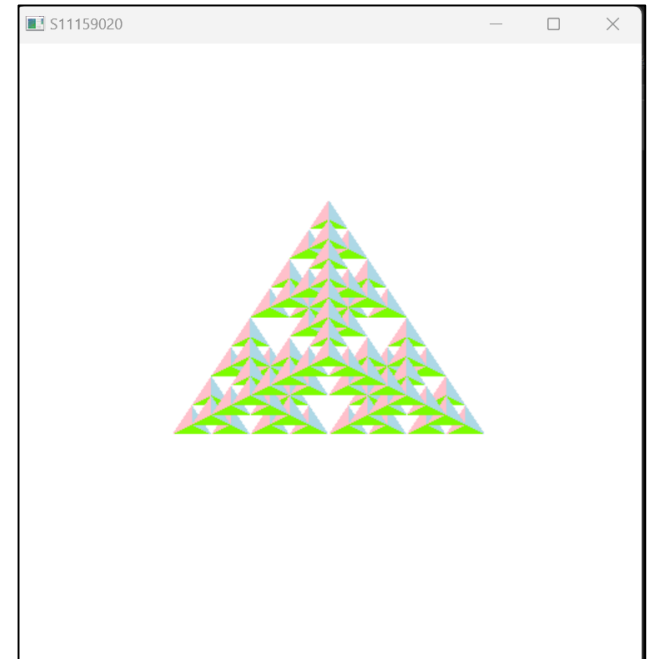
Level = 2



Level = 1



Level = 3



## 心得

起初先利用課程簡報上的範例程式做練習，做出了 2D 的三角形切割，後來要實作 3D 的才發現有透視與不透視兩種圖形，簡報後面補充的函式會畫出透視的 3D，與本次作業切割的方法些微不同，同樣都是找兩點之間的中點位置，但三角錐需要 4 個點 6 條邊都計算到，而 Menu 與 Keyboard 的功能實作部分讓我想起物件導向程式設計課程的回憶，雖然不完全熟悉 FreeGlut 所有 callback function 能做到的功能，但優點就是每增加一個功能就只要寫出該 function 的內容，再到 main function 中呼叫即可，與原先其餘的程式並不衝突，無需做過多的更動。

## 參考文獻

- [1] <https://medium.com/ericzhan-publication/opengl%E7%AD%86%E8%A8%98-%E5%89%B5%E5%BB%BA%E5%88%9D%E5%A7%8B%E5%8C%96%E8%A6%96%E7%AA%97-dbe48fbb7a06>
- [2] [https://blog.csdn.net/xie\\_zi/article/details/1963383](https://blog.csdn.net/xie_zi/article/details/1963383)
- [3] <https://medium.com/ericzhan-publication/opengl%E7%AD%86%E8%A8%98-%E5%BD%88%E5%87%BA%E9%81%B8%E5%96%AE-pop-up-menu-620ee0451148>
- [4] <https://www.cnblogs.com/flyinggod/p/12927108.html>
- [5] <https://www.youtube.com/watch?v=zN1HPjTiSoY&t=1011s>
- [6] <https://cglabprojects.blogspot.com/2013/03/sierpinski-gasket-in-tetrahedron.html>