Final Project: Ocean

111598066　資工碩一　許哲維

一、功能項目

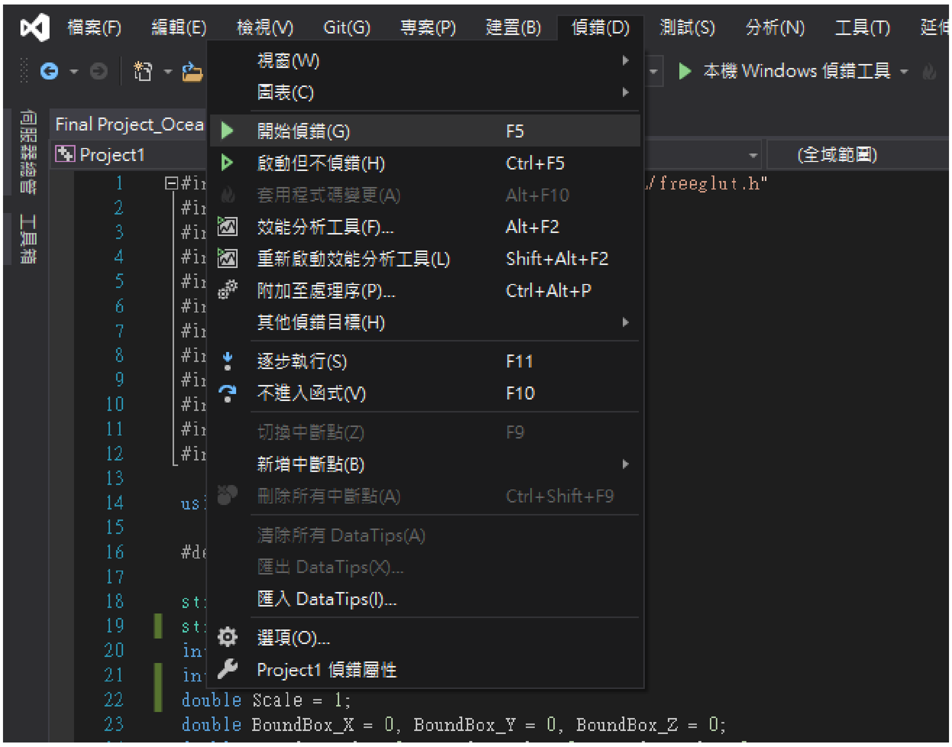
1. Setup

系統	Window 10　專業版(22H2)
環境	Visual Studio Community 2019

需將 OpenGL 的相關套件引入，並於 Visual Studio 專案屬性頁面中 C++\C 進階選項中停用特定警告 4996、在前置處理器中前置處理器定義增加 _CRT_SECURE_NO_WARNINGS，並依照先前使用 OpenGL 與 OpenCV 之步驟來使用其 Libary。修改程式碼之 `string Location_Path = "D:\\CG\\Lab\\final\\";` 改為電腦 OBJ 檔與圖像檔的位置。

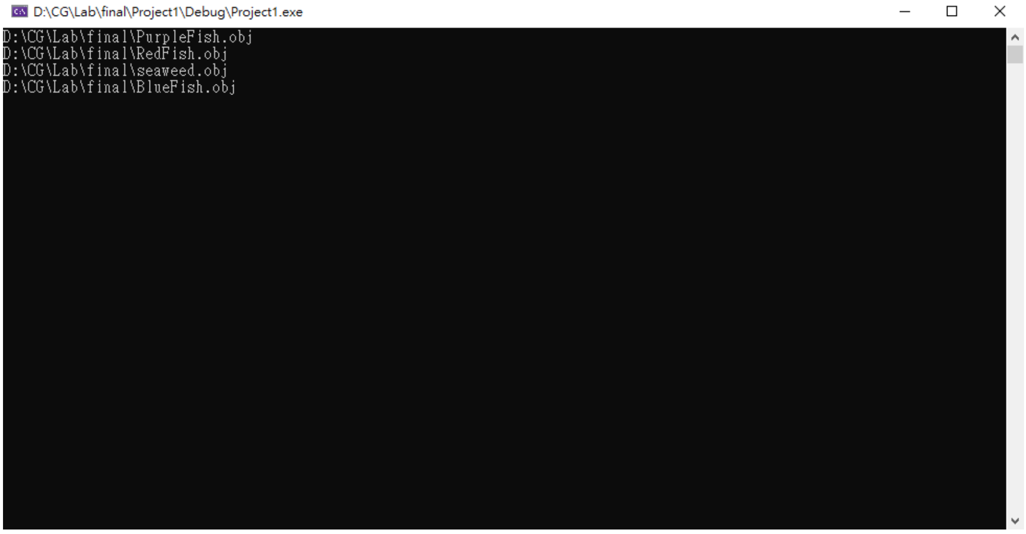
2. How to use

在 Visual Studio 按偵錯 > 開始偵錯。

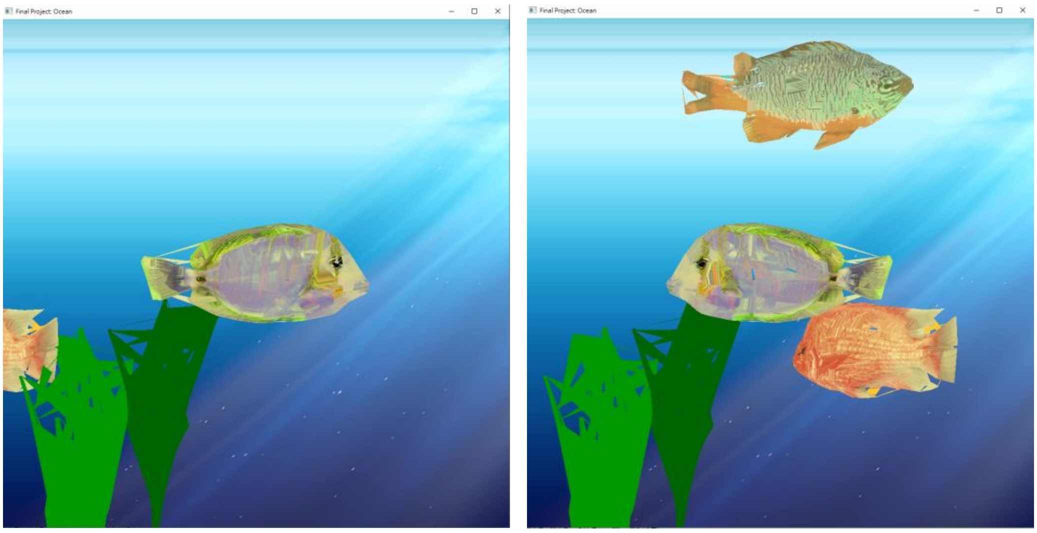


```
/freeglut.h"
double Scale = 1;
double BoundBox_X = 0, BoundBox_Y = 0, BoundBox_Z = 0;
```

執行後，會先顯示 CMD(下圖一)，初始化後自動導入預設檔與設定，並顯示視窗(下圖二)自動執行動畫。



```
D:\CG\Lab\final\PurpleFish.obj
D:\CG\Lab\final\RedFish.obj
D:\CG\Lab\final\seaweed.obj
D:\CG\Lab\final\BlueFish.obj
```



按右上 X 即可關閉程式。

※　其餘截圖可至 ScreenShot 資料夾觀看截圖。

二、程式碼

1. SetupRC

```
void SetupRC()
{
        GLbyte* pBytes;
        GLint nWidth, nHeight, nComponents;
        GLenum format;

        GLfloat whiteLight[] = { 0.45f,0.45f,0.45f,1.0f };
        GLfloat sourceLight[] = { 0.25f,0.25f,.25f,1.0f};
        GLfloat lightPos[] = { 0.f,25.0f,20.0f,0.0f };
        GLfloat light_ambient[] = { 0.2, 0.2, 0.2, 1.0 };
        GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
        GLfloat light_specular[] = { 0.0, 0.0, 0.0, 1.0 };
        GLfloat mat_specular[] = { 0.0, 0.0, 0.0, 1.0 };
        GLfloat mat_diffuse[] = { 0.8, 0.6, 0.4, 1.0 };
        GLfloat mat_ambient[] = { 0.8, 0.6, 0.4, 1.0 };
        GLfloat mat_shininess = { 20.0 };

        glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
        glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
        glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
        glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess);

        glShadeModel(GL_SMOOTH); /*enable smooth shading */
        glEnable(GL_LIGHTING); /* enable lighting */
        glEnable(GL_LIGHT0); /* enable light 0 */
        glEnable(GL_DEPTH_TEST);
        glEnable(GL_COLOR_MATERIAL);

        glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
        glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
        //
        glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
        glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE); // 設定 openGL 材質紋理的參
數和材質的組合模式
        glGenTextures(4, textures); // 註冊一個大小為 4 的陣列讓 openGL 儲存材質，名稱為 textures

        FileName = "background";
        cv::Mat Image = cv::imread(LocationPath + FileName + ".jpg"); // 利用 openCV 讀取圖片檔案
        if (Image.empty()) {
                std::cout << "Background empty\n";
        }
        else {
                // 將讀取進來的圖片檔案當作材質存進 textures 中
                cv::flip(Image, Image, 0);
                glGenTextures(1, &textures[0]);
                glBindTexture(GL_TEXTURE_2D, textures[0]);

                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);

                glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, Image.cols, Image.rows, 0, GL_BGR_EXT,
GL_UNSIGNED_BYTE, Image.ptr());
        }

        FileName = "PurpleFish";
        Image = cv::imread(LocationPath+FileName+".jpg"); // 利用 openCV 讀取圖片檔案
        if (Image.empty()) {
                std::cout << "Purple Fish empty\n";
        }
        else {
                // 將讀取進來的圖片檔案當作材質存進 textures 中
                cv::flip(Image, Image, 0);
                glGenTextures(1, &textures[1]);
                glBindTexture(GL_TEXTURE_2D, textures[1]);

                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);

                glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, Image.cols, Image.rows, 0, GL_BGR_EXT,
GL_UNSIGNED_BYTE, Image.ptr());
        }
        //
        obj1 = LoadOBJ(LocationPath + FileName + ".obj");
        ...(省略)...
```

```
		return;
}
```

SetupRC 用於載入 OBJ 檔、圖像等等。

2. RenderScene

```
void RenderScene(void)
{
		glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
		glShadeModel(GL_SMOOTH);
		glMatrixMode(GL_MODELVIEW);
		glLightfv(GL_LIGHT0, GL_POSITION, vLightPos);
		glLoadIdentity();
		glEnable(GL_DEPTH_TEST);
		glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
		glPushMatrix();
		//
		glDisable(GL_LIGHTING);
		glColor3ub(255, 255, 255);
		glEnable(GL_TEXTURE_2D); //  啟動 openGL 的 2D 材質填充模式

		glBindTexture(GL_TEXTURE_2D, textures[0]);
		glBegin(GL_QUADS);
		glTexCoord2f(0.0f, 1.0f);
		glVertex3f(-500.0f, 500.0f, -999.0f);
		glTexCoord2f(0.0f, 0.0f);
		glVertex3f(-500.0f, -500.0f, -999.0f);
		glTexCoord2f(1.0f, 0.0f);
		glVertex3f(500.0f, -500.0f, -999.0f);
		glTexCoord2f(0.0f, 1.0f);
		glVertex3f(500.0f, 500.0f, -999.0f);
		glEnd();

		glDisable(GL_TEXTURE_2D);
		glPopMatrix();
		//
		ColorMode *= -1;
		glPushMatrix();
		glTranslatef(-350+ ColorMode, -560, -180);
		glRotatef(150, 0, 1, 0);
		glColor3f(0, 0.6, 0);
		glBegin(GL_QUADS);
		obj3.SetupResize();
		for  (vector<vector<float>> ::iterator  line_iter  =  obj3.Face.begin(); line_iter  !=  obj3.Face.end();
line_iter++)
		{
				vector<float>   ::iterator line_iter2;
				line_iter2 = line_iter->begin();
				glTexCoord2fv(&(obj3.VertexTexture)[*line_iter2 - 1][0]);
				glVertex3fv(&(obj3.Vertex)[*line_iter2 - 1][0]);
				line_iter2++;
				glTexCoord2fv(&(obj3.VertexTexture)[*line_iter2 - 1][0]);
				glVertex3fv(&(obj3.Vertex)[*line_iter2 - 1][0]);
				line_iter2++;
				glTexCoord2fv(&(obj3.VertexTexture)[*line_iter2 - 1][0]);
				glVertex3fv(&(obj3.Vertex)[*line_iter2 - 1][0]);
				line_iter2++;
				glTexCoord2fv(&(obj3.VertexTexture)[*line_iter2 - 1][0]);
				glVertex3fv(&(obj3.Vertex)[*line_iter2 - 1][0]);
				line_iter2 = line_iter->begin();

				glVertex3fv(&(obj3.Vertex)[*line_iter2 - 1][0]);
				glTexCoord2fv(&(obj3.VertexTexture)[*line_iter2 - 1][0]);
		}
		glEnd();
		glPopMatrix();
		...(省略)...
		/*Fish 1*/
		glPushMatrix();
		glTranslatef(0, 0, 0);
		Angel += 5;
		glRotatef(-1*Angel, 0, 1, 0);
		glColor3f(1, 0.9588, 0.6392);
		glEnable(GL_TEXTURE_2D);
		glBindTexture(GL_TEXTURE_2D, textures[1]);
		glBegin(GL_QUADS);
		obj1.SetupResize();
		for  (vector<vector<float>> ::iterator  line_iter  =  obj1.Face.begin(); line_iter  !=  obj1.Face.end();
line_iter++)
		{
				vector<float>   ::iterator line_iter2;
				line_iter2 = line_iter->begin();
				glTexCoord3fv(&(obj1.VertexTexture)[*line_iter2 - 1][0]);
				glVertex3fv(&(obj1.Vertex)[*line_iter2 - 1][0]);
				line_iter2++;
```

```
        glTexCoord3fv(&(obj1.VertexTexture)[*line_iter2 - 1][0]);
        glVertex3fv(&(obj1.Vertex)[*line_iter2 - 1][0]);
        line_iter2++;
        glTexCoord3fv(&(obj1.VertexTexture)[*line_iter2 - 1][0]);
        glVertex3fv(&(obj1.Vertex)[*line_iter2 - 1][0]);
        line_iter2++;
        glTexCoord3fv(&(obj1.VertexTexture)[*line_iter2 - 1][0]);
        glVertex3fv(&(obj1.Vertex)[*line_iter2 - 1][0]);
        line_iter2 = line_iter->begin();

        glVertex3fv(&(obj1.Vertex)[*line_iter2 - 1][0]);
        glTexCoord3fv(&(obj1.VertexTexture)[*line_iter2 - 1][0]);
    }
    glEnd();
    glDisable(GL_TEXTURE_2D);
    glPopMatrix();
    ...(省略)...
    glutSwapBuffers();
    return;
}
```

RenderScene 用於動畫效果、貼圖等等。

3. LoadOBJ

```
OBJfile LoadOBJ(string filename)
{
    char temp_string[256];
    OBJfile obj;
    FILE* file = fopen((filename).c_str(), "r");
    cout << (filename).c_str() << endl;

    if (file == NULL)
    {
        printf("Impossible to open the file !");
        return obj;
    }

    while (1) {
        int eof = fscanf(file, "%s", temp_string);

        if (eof == EOF)
        {
            break;
        }

        if (strcmp(temp_string, "v") == 0)
        {
            vector<float> temp_vertices;
            float X_Y_Z[3];

            fscanf(file, "%f %f %f", &X_Y_Z[0], &X_Y_Z[1], &X_Y_Z[2]);
            obj.objBoundBox = GetBoundingBox(X_Y_Z[0], X_Y_Z[1], X_Y_Z[2], obj);

            for (int i = 0; i < 3; i++)
            {
                temp_vertices.push_back(X_Y_Z[i]);
            }
            obj.Vertex.push_back(temp_vertices);
        }
        else if (strcmp(temp_string, "vt") == 0)
        {
            vector<float> temp_vertices;
            float dontcare;
            float X_Y_Z[3];

            fscanf(file, "%f %f %f", &X_Y_Z[0], &X_Y_Z[1], &X_Y_Z[2]);

            for (int i = 0; i < 3; i++)
            {
                temp_vertices.push_back(X_Y_Z[i]);
            }
            obj.VertexTexture.push_back(temp_vertices);
        }
        else if (strcmp(temp_string, "f") == 0)
        {
            float X_Y_Z[4];
            float vX_vY_vZ[4];
            int dontcare[8];
            vector<float> temp_vertices, temp_verticesT;

            fscanf(file, "%f/%f/%f %f/%f/%f %f/%f/%f %f/%f/%f", &X_Y_Z[0], &vX_vY_vZ[0],
&dontcare[1], &X_Y_Z[1], &vX_vY_vZ[1], &dontcare[3],
                &X_Y_Z[2], &vX_vY_vZ[2], &dontcare[5], &X_Y_Z[3], &vX_vY_vZ[3], &dontcare[7]);

            for (int i = 0; i < 4; i++)
```

```
                        {
                                temp_vertices.push_back(X_Y_Z[i]);
                        }
                        obj.Face.push_back(temp_vertices);
                }
        }
        fclose(file);
        return obj;
}
```

LoadOBJ 用於載入畫四邊形之 OBJ 檔，並保存 Vertex、VertexTexture、Face 使 RenderScene 能順利載入。

※ 其餘程式碼可至 Final Project_Ocean.cpp 觀看。

三、技術難題

Q: OBJ 載入問題。

A: 由先前的 Midterm Project 中直接載入 OBJ 檔會發生錯誤，經與同學討論、上網搜尋資料發現是 OBJ 檔有些資訊的不同，修改後已可以正確載入。

Q: 動畫製作困難。

A: 因對於動畫效果不是非常清楚，因此花費的許多時間在搜尋資料、與同學討論以完成該要求。

Q: Shadow 製作問題。

A: 因時間、能力等因素尚未正確實現該功能。