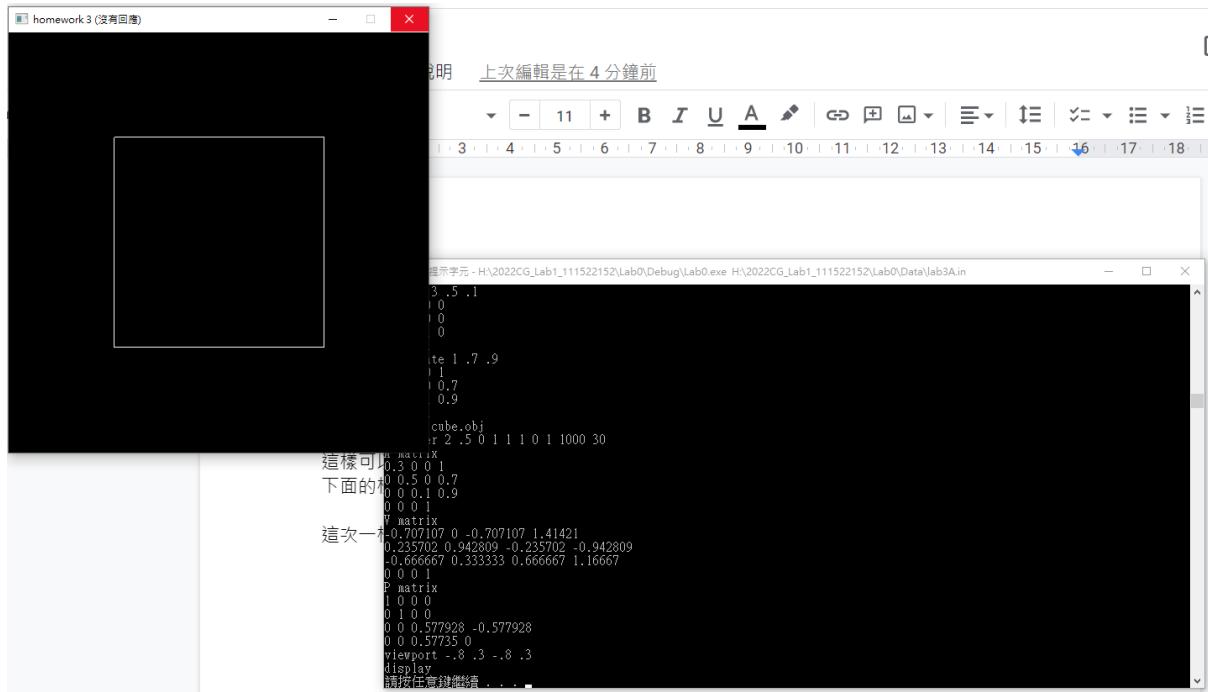


執行與編譯方法:我是使用windows來寫, 所以有地方要額外處理, 由於是用lab0做修改, 所以檔案位置(exe)與專案編譯也是與lab0相同

在程式執行後, 在homework3或CMD視窗按下任一按鍵, 就會開始執行

我這次作業尚未完成光罩的部分, 主要是完成的z buffer的部分與其他剩餘的地方

這次一樣是使用cmd將exe和測資放在一起, 但要注意我將exe與Mesh資料夾放在一起, 以方便讀取object, 若移動或刪除位於與exe同個位置的Mesh資料夾, 會使程式讀取不到object, 會發生下圖的情形



我打包的資料夾裡附上的exe為CMD版的, 但我編譯器裡放的是編譯版的程式碼, 這點請注意

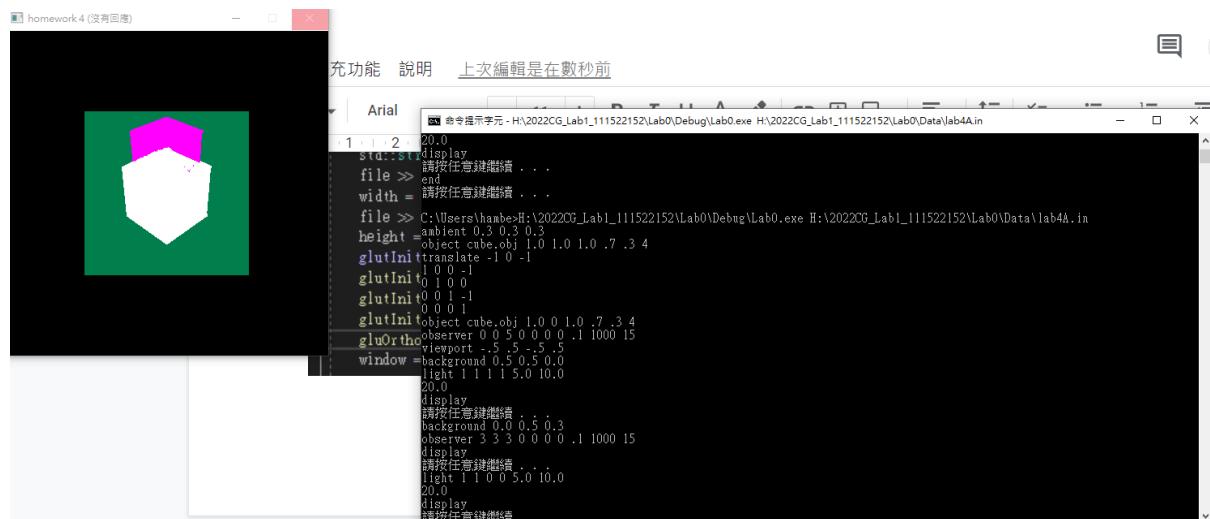
我附上兩個版本的程式碼, 分別是編譯器版和CMD版, CMD版編譯後的程式只需將程式與測資放入CMD中即可, 但高機率有錯, 若使用編譯器版則將程式碼(編譯器版).txt中的資料複製進編譯器中, 並且隨著測資修改測資位置, 按下編譯鈕在homework 4視窗中按下任一按鍵後等待結果, 有四個測資, 所以要重新編譯四次, 我用的是絕對路徑(下圖)。

```

int main(int argc, char* argv[]) {
    file.open("H:\2022CG_Lab1_111522152\Lab0\Data\lab3E.in");
    std::string command;
    file >> command;
    width = stoi(command);
    file >> command;
    height = stoi(command);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutInitWindowSize(width, height);
    glutInitWindowPosition(100, 100);
    gluOrtho2D(0, width, 0, height);
    window = glutCreateWindow("homework 3");
}

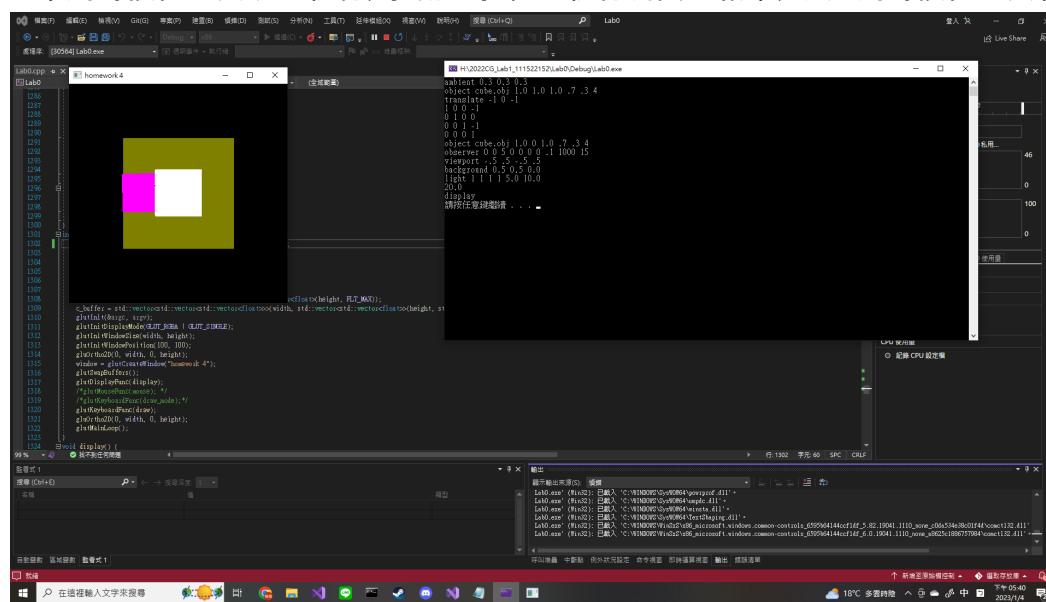
```

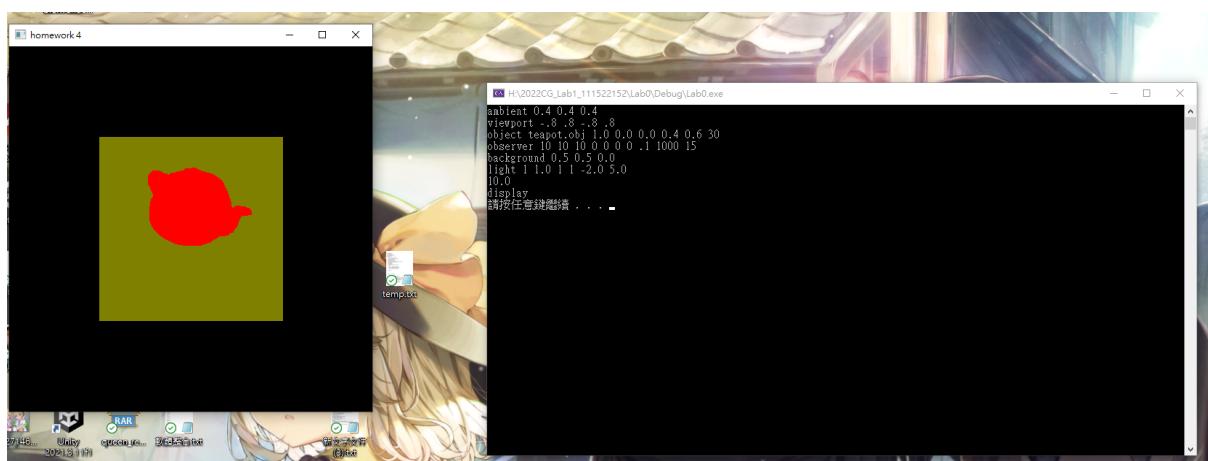
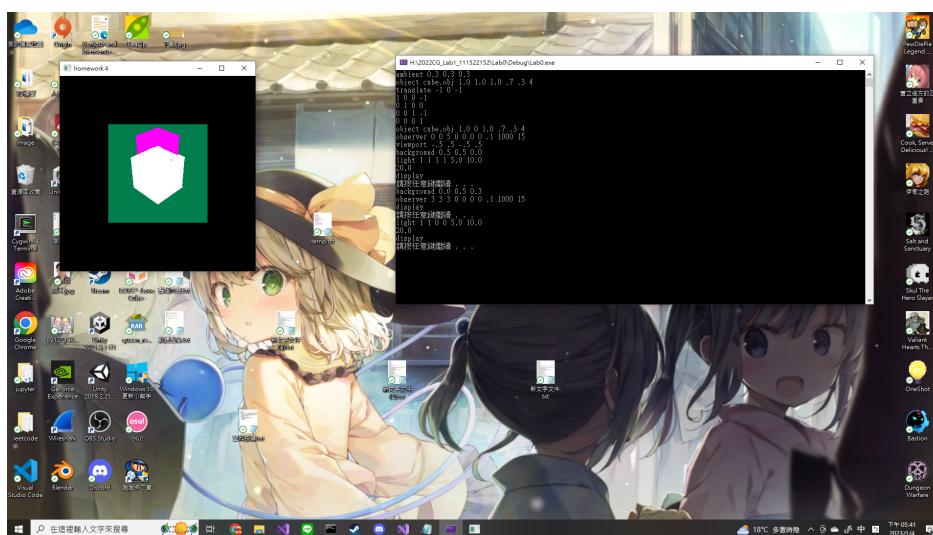
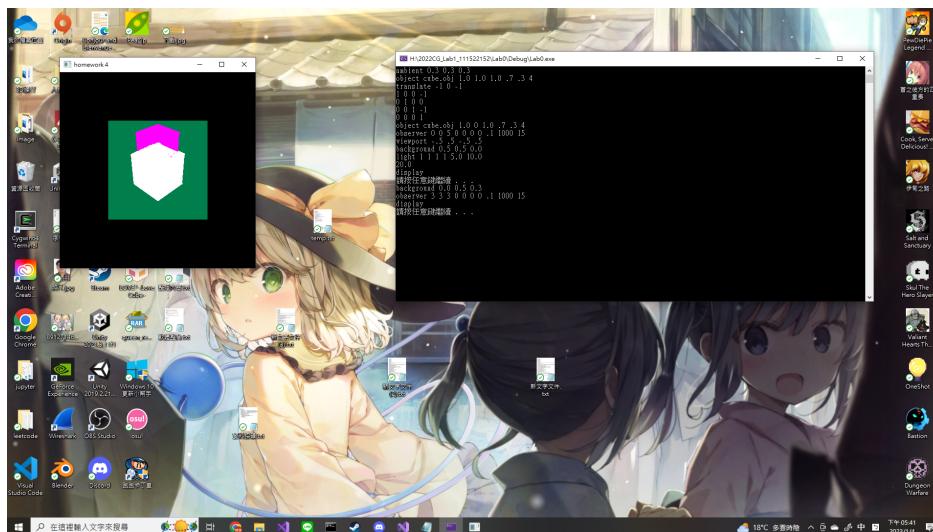
## CMD版exe執行方法如下

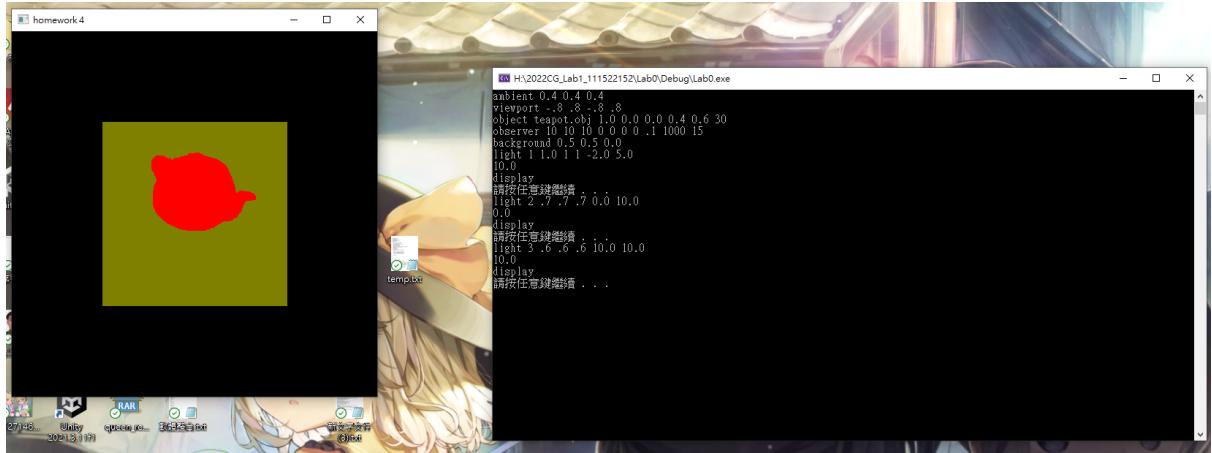
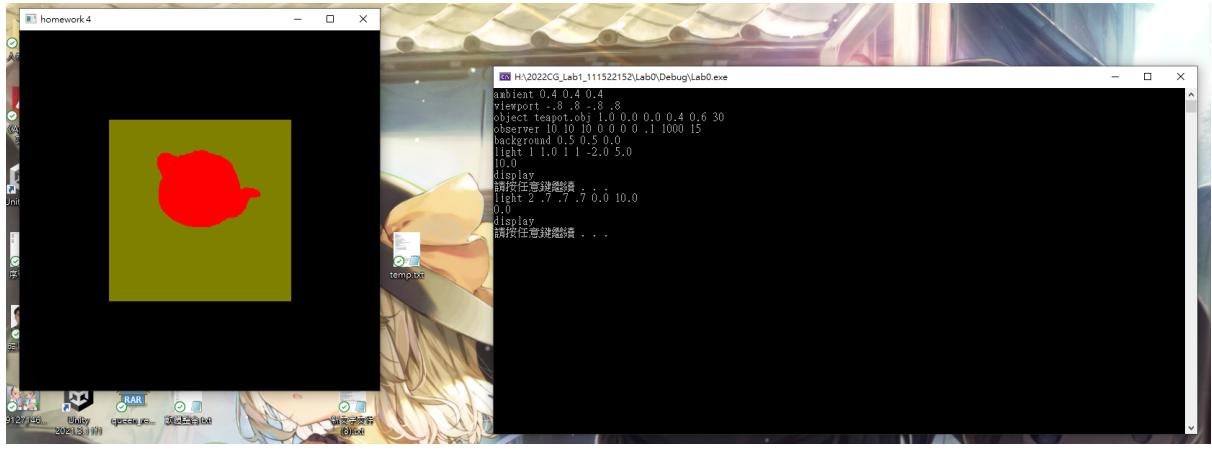


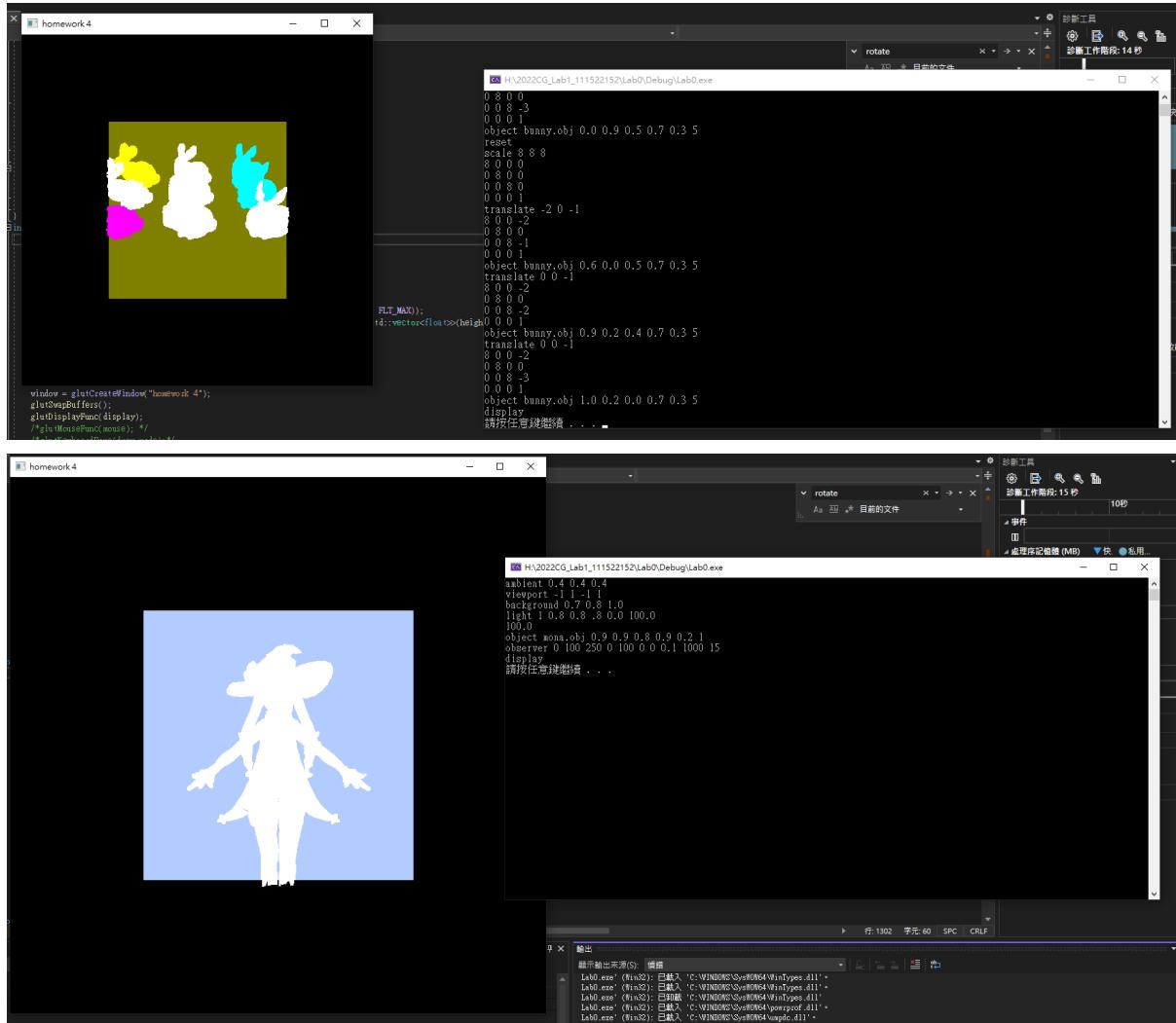
在CMD視窗中可看見，要將exe與測資的絕對路徑放入，非檔名即可

如果到時候仍無法成功執行，我附上我程式執行測資的結果，以免到時候仍無法執行成功









最後的馬斯克的頭沒有結果，我想是在加入z buffer之後所出現的bug，所以結果就不呈現