國立臺南大學資訊工程學系

**Computer Graphics**

第三次作業

**題目 : 3D Graphics**

**Input and Interaction**

**Transformation**

|  |  |  |
| --- | --- | --- |
| 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.12**
* **心得------------------------------------------------------P.13**

**作業說明**

1. 題目:

實作Sierpinski Gasket in a Tetrahedron，並添加旋轉相機與物體角度之功能

1. Requirements :

* 3D Gasket – Regular tetrahedron with volume subdivision
* Input Devices – Mouse
  + Trigger the menu by pressing the right mouse button
  + Press the left mouse button and move the mouse to change θ and φ according to changes in x and y.
  + Press both the left and right mouse buttons and move the mouse to change r according to changes in x and/or y
* Input Devices – Keyboard
  + Press the ‘q’ or ‘Q’ to quit the program
* Pop Menu – Make two pop menus in main menu and exit the program in the main menu.
  + One of the pop menus in main menu is to select and change the subdivision level of the displaying tetrahedron.
  + Another one is to select which direction the tetrahedron rotate in. It can rotate automatically according to the x, y and z axes in clockwise or counter-clockwise. The automatically rotation can be stopped by press the “stop” button in this submenu as well.
* Initial Subdivision Level = 0
* The window title is your Student ID
* Locate the camera position in spherical coordinates and orient the camera to the sphere.

**程式設計環境架構**

1. 程式語言

C++ in MS Windows 11

1. 程式開發工具

Microsoft Visual Studio

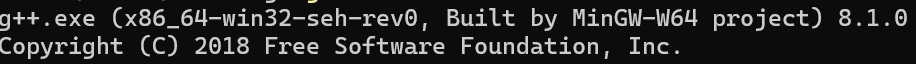
1. 電腦硬體

CPU: Intel i5-1135G7 ,

Main Memory: 16GB LPDDR4X,

SSD: PCIe 512GB

1. 編譯器



1. GLUT函式庫名稱與版本

Freegult 3.6.0



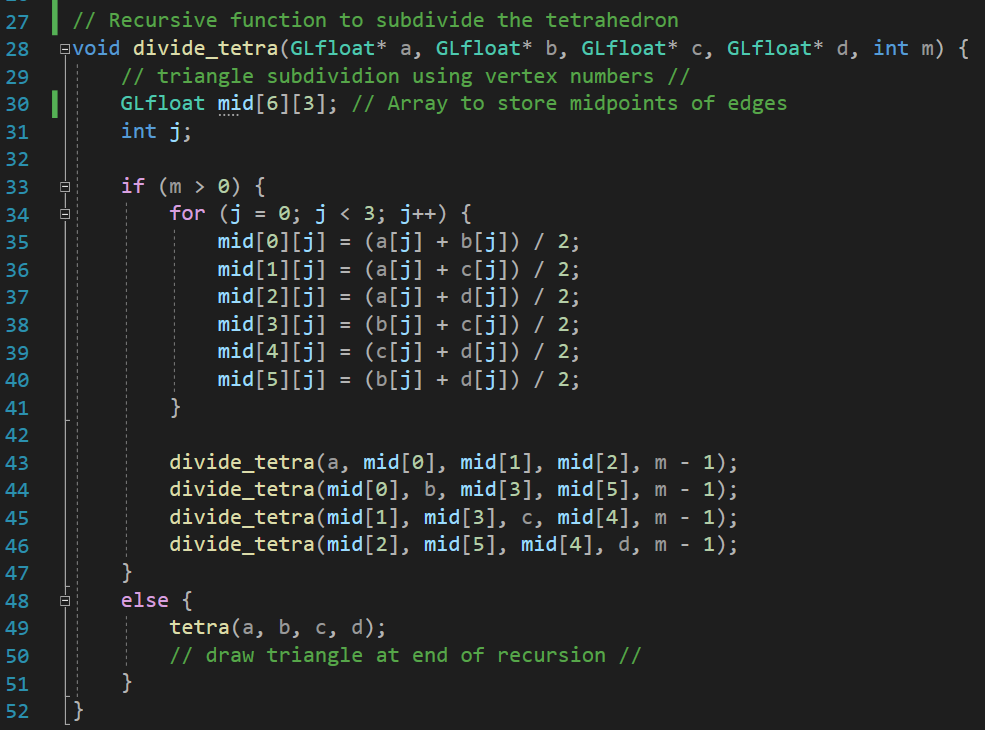
OpenGL 3.2

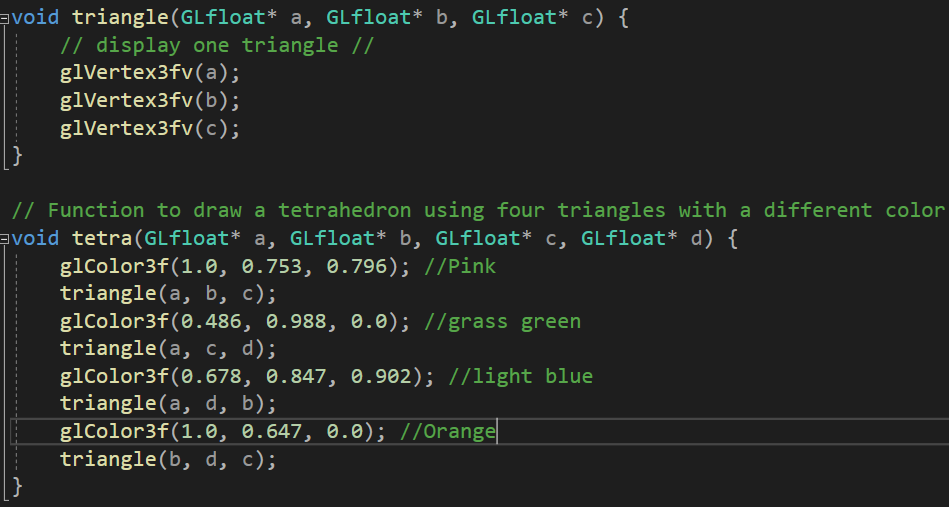


**Function展示**

1. Draw a tetrahedron

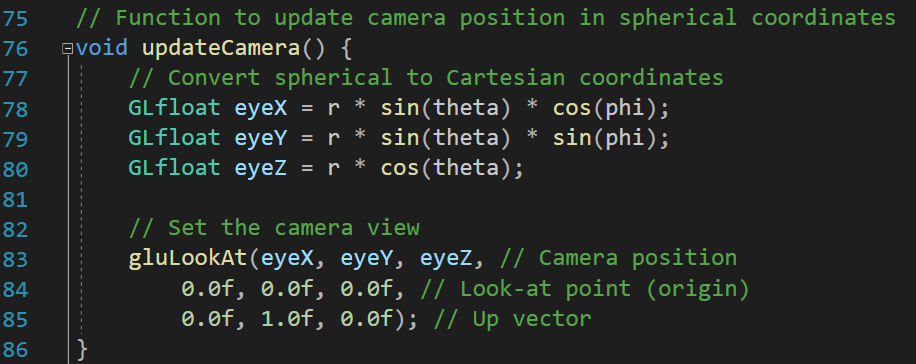
三角錐中共有4個點形成6條邊，利用二維陣列存取6條邊個別的中點座標位置(x, y, z)。以找出每條邊之中點的方式切割圖形，將切割出的中點座標位置存進mid陣列之中，以Recursion的方式切割直到Base case (n = 0)，當n = 0時則直接顯示未切割過的tetrahedron。





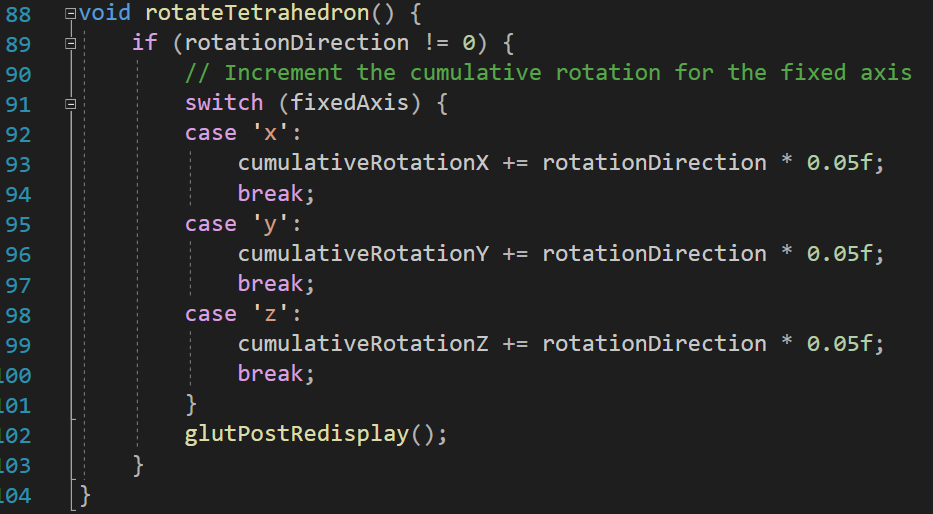
1. Rotate the Camera

將相機設置於球體座標(γ,θ,φ)中，將(γ,θ,φ)轉換成x-y座標的(x, y, z)，基於新計算出的座標值更新相機位置，以glutLookAt()設置參數調整視角。



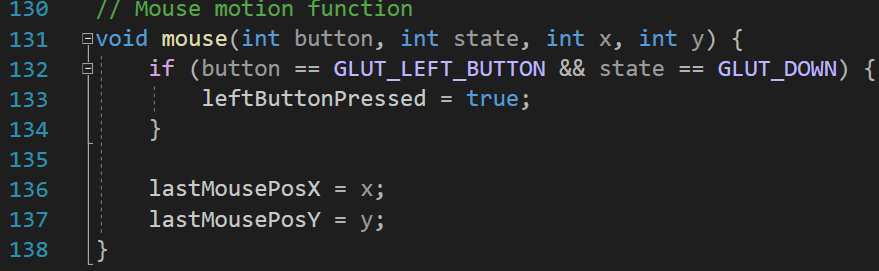
1. Rotate the Tetrahedron

若當時錐體不為停止旋轉的狀態，則判斷當前的固定軸，以每次0.05f的速度逐次增加或減少個別的旋轉角度，最後以glutPostRedisplay()重新繪製，達到持續轉動的效果。



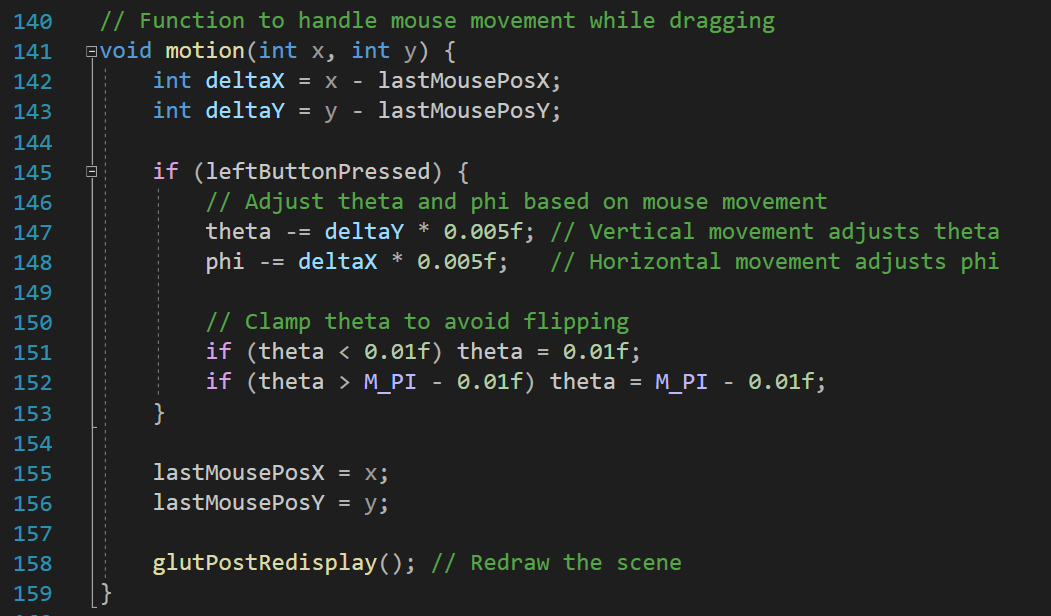
1. Trigger by mouse button

判斷滑鼠左鍵是否為按下的狀態，若是則以(x, y)記錄當前滑鼠的位置於變數中。

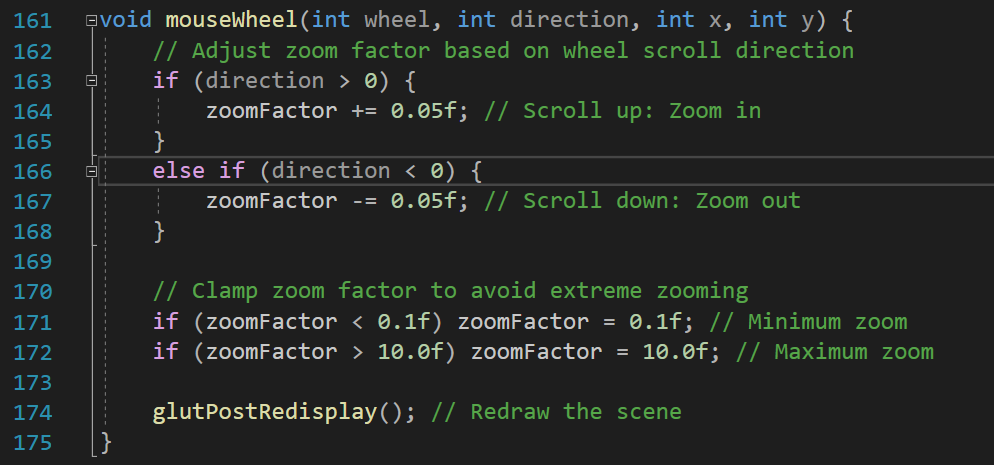


1. Drag the Mouse

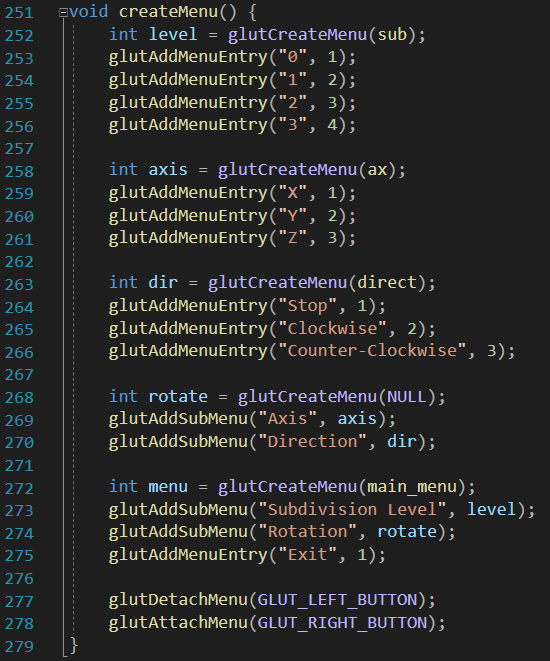
按住左鍵並拖移滑鼠去改變相機視角，以當前滑鼠位置(x, y)與按下左鍵時所記錄的位置相減，計算出x和y的變化量，再以x和y之變化量個別在垂直移動和水平移動中調整θ和φ，最後以當下滑鼠的位置(x, y)更新lastMousePosition並以glutPostRedisplay()重新繪製。

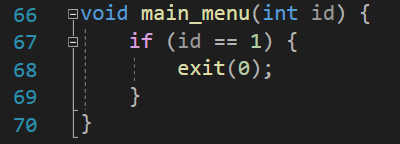


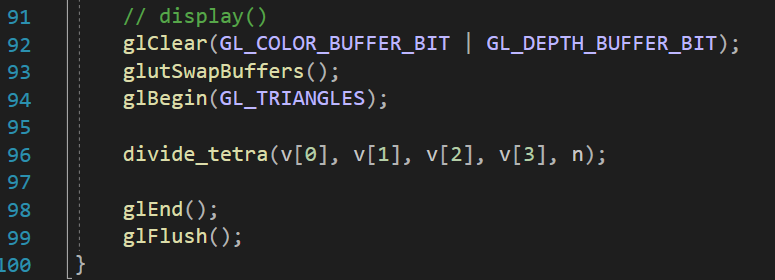
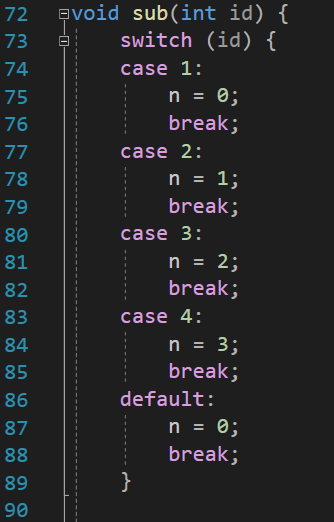
1. Zoom by Mouse wheel

以滑鼠滾輪調整相機位置的γ，實現放大與縮小視角。若滾輪向上滾動即增加ZoomFactor；反之，滾輪向下滾動則減少ZoomFactor，設定縮放的上下限，最後基於新的ZoomFactor以glutPostRedisplay()重新繪製。

1. Create a Menu

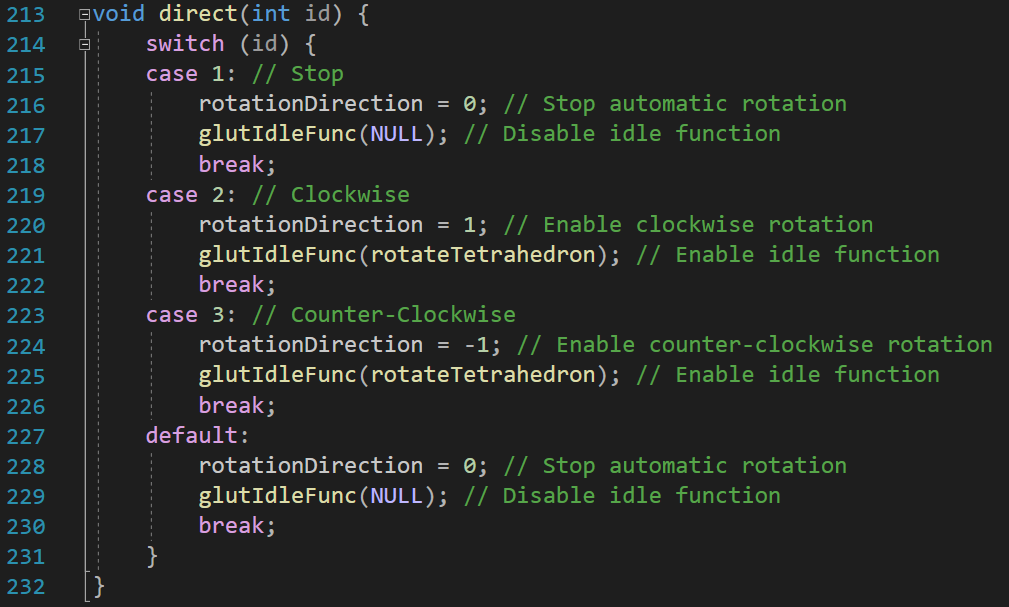
Menu是利用回傳id的方式判斷點選該欄位後需要執行的動作。先Create出各Entry的Name和預設回傳的id值，利用點選滑鼠右鍵呼叫出menu，點選滑鼠左鍵則取消顯示menu。

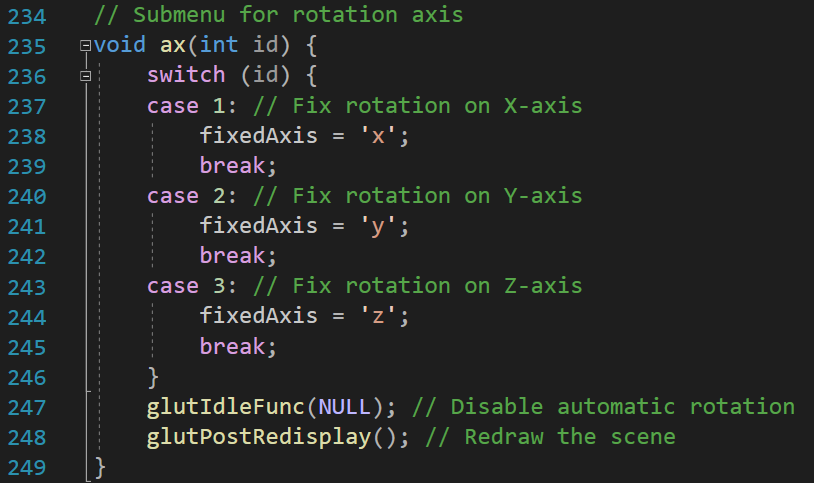
Main menu 中若回傳的id為1，則結束程式。

 第一個Submenu(sub(id))以回傳的id值決定新的n值(tetrahedron 要切割幾層)。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)

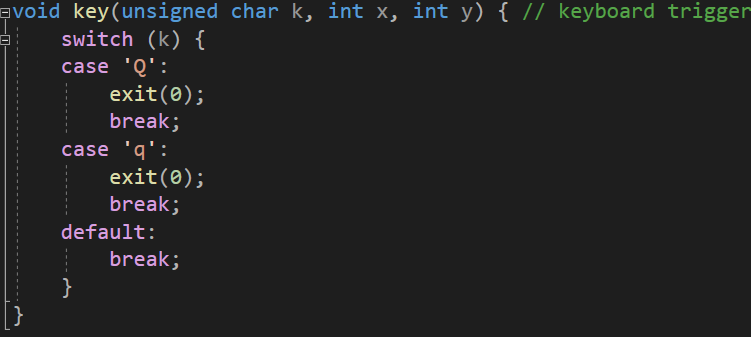
第二個 Submenu再另外延伸兩個submenus，決定錐體旋轉的方向。

* + - 1. 以回傳值決定新的旋轉方向為(-1, 1, 0)，決定重新繪製的錐體該由逆時針方向、順時針方向旋轉或停止旋轉
      2. 以回傳值決定新的固定軸(x, y, z)，更改當前錐體應該以固定哪一個軸去做自轉。

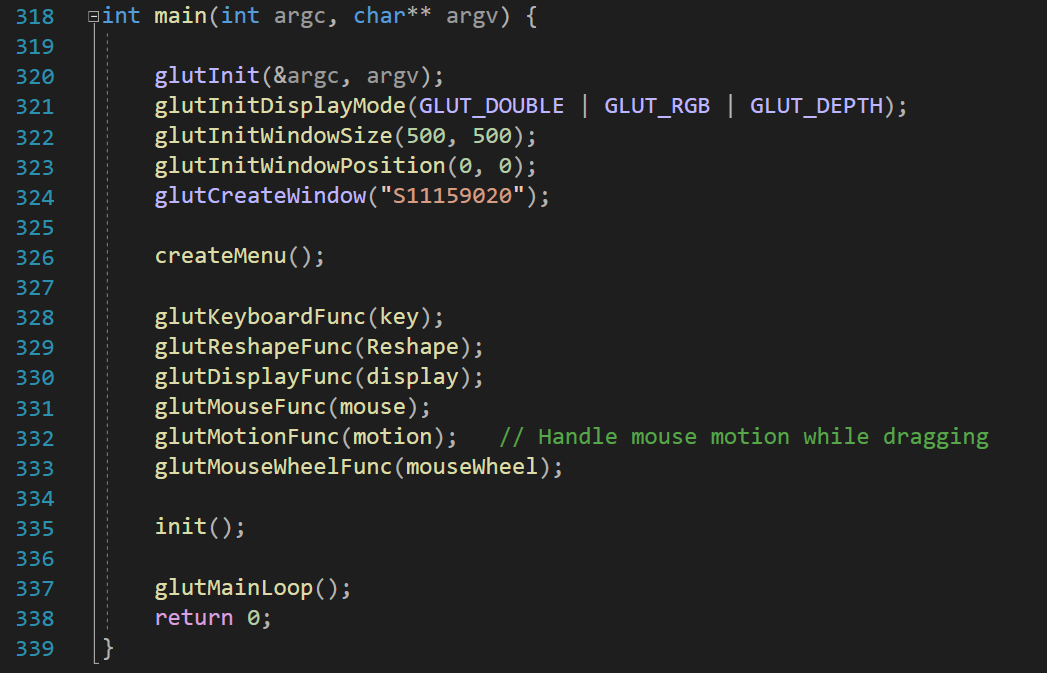




1. Trigger by Keyboard

 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。

1. Main Function



**執行成果展示**

[**https://www.youtube.com/watch?v=1qmahte6PvU**](https://www.youtube.com/watch?v=1qmahte6PvU)

**心得**

延續上一份作業，這次額外新增了相機視角與轉變視角的功能，自己對球體座標不熟悉導致在轉換到卡式座標時遇到了一點困難，難以單純用想像的方式去模擬相機與物體之間的相對關係，以及應該會在轉動後呈現何種效果，也因為沒有弄懂作業說明的細項，造成實作出的結果與助教預期檢查到的成果有落差，驗收當天聽完解釋後才趕緊把缺失的改變相機視角之功能補上。這次作業所探討的主體很有趣卻也覺得很抽象。