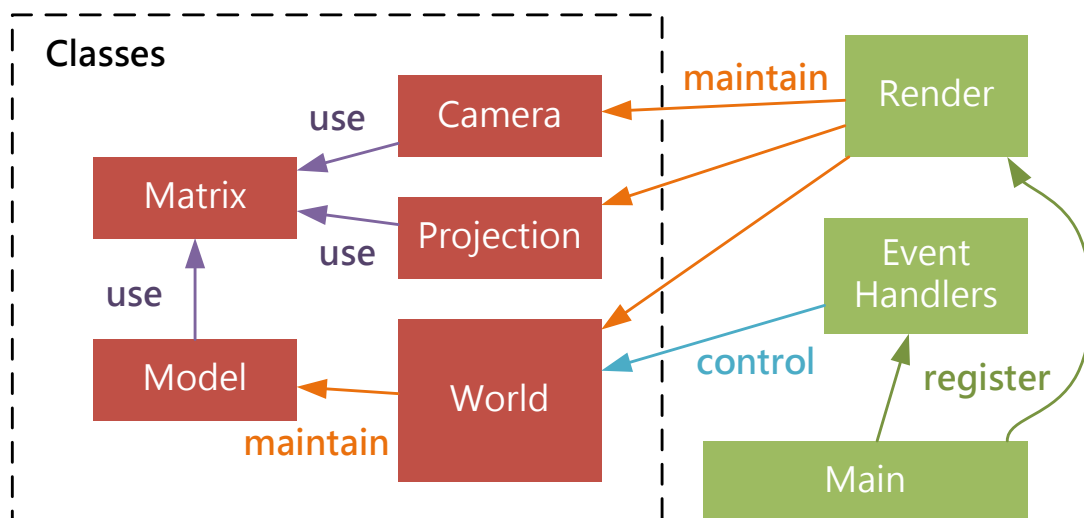


Computer Graphics Homework #2 Report

103062528, CS, Yu-Shan Lin

一、架構設計



上圖為我的系統之架構設計，大略可以分為「Classes」與「其他」兩個部分：

1. Classes

A. Matrix

基本上就是一個 4×4 的二維矩陣，並提供一些矩陣運算的 methods

B. Model

紀錄單一 model 的各項資料，包括 vertex、color，以及用於這個 model 的 transform matrixes

C. World

負責管理所有的 model，並在一開始時將 model 移到正確的位置

D. Camera

代表使用者的視角，掌控 eye、center、up vector 等資訊

E. Projection

掌控 projection 的方式，並產生對應的 transform matrix

2. 其他

A. Render

包含與繪圖相關的 functions，並且負責管理 world、camera 與 projection 的物件

B. Event Handlers

主要負責處理滑鼠與鍵盤的事件，並呼叫對應的動作

C. Main

程式的進入點，負責註冊上面兩個檔案的 functions

二、程式執行特點

本程式當 onRender 這個 function 被呼叫後，就會先產生一個存有 identity 的 matrix 物件。並從 camera 與 projection 取出預先計算好的 transform matrix，再乘上前面的 matrix。做完之後，將得到的結果轉交給 world，也就是呼叫 world 的 draw，讓 world 畫出想畫的內容。World 則會再轉交給其管理的 model，讓 model 自行根據需要繪圖。

這邊特別撰寫 matrix class，主要是為了要實現類似早期 OpenGL 之中，matrix stack 的效果。因為 function 的呼叫就如同 stack 一般，一層一層往內，同時如果使用 call by value 的方式傳遞 matrix 物件的話，那麼上層 function 中的 matrix 就會保有原本的資料，不受到下層 function 的影響。這樣就可以實現類似的效果。

三、程式使用方式

使用滑鼠拖曳或者滾輪來調整 x, y, z 數值

想要切換模式可以按以下按鍵：

Model Transform

T: Translate Mode

R: Rotate Mode

S: Scale Mode

M: Switch model

Camera Control

E: Eye Control Mode

C: Center Control Mode

U: Up Vector Control Mode

其他指令

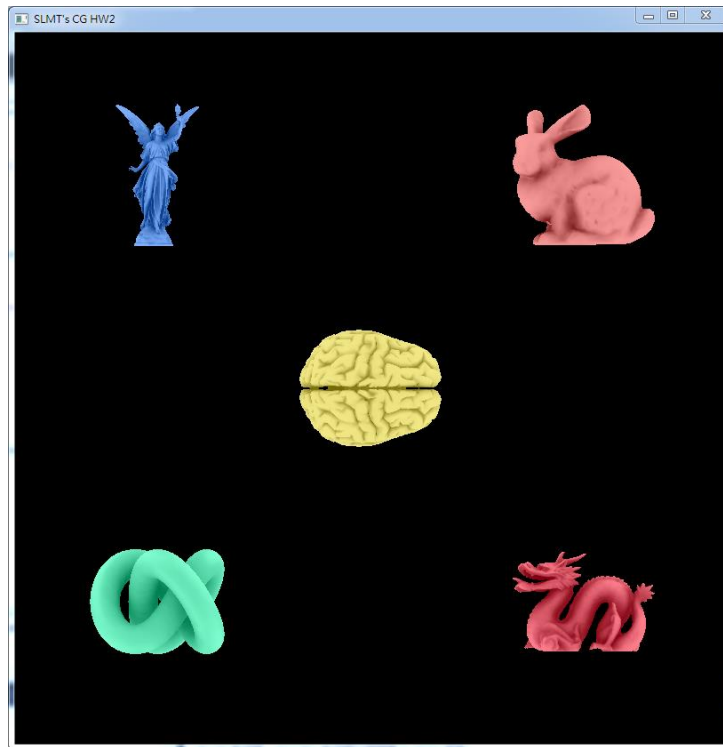
P: 切換 Projection Mode

Enter: 重設 Camara 或者 Model (依照現在模式判斷)

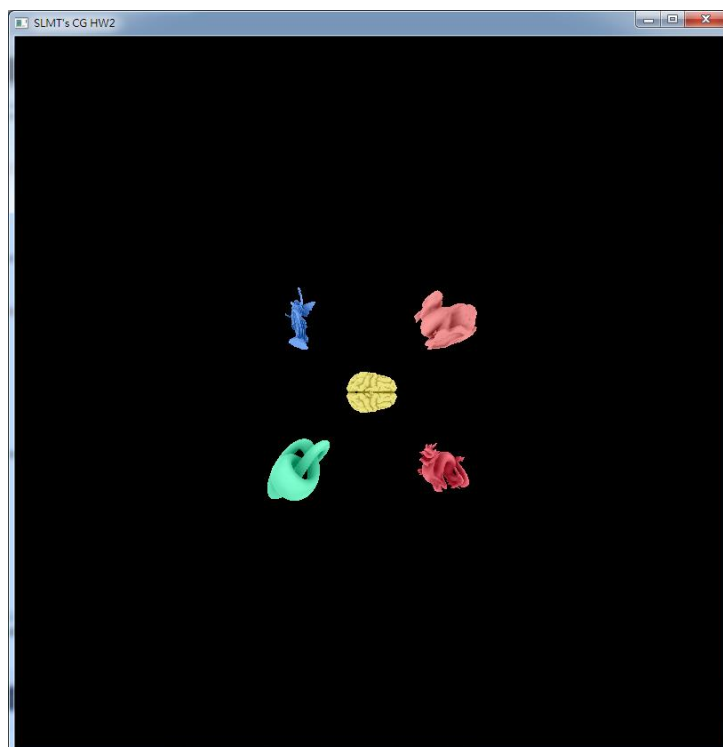
H: 顯示幫助清單

四、程式截圖

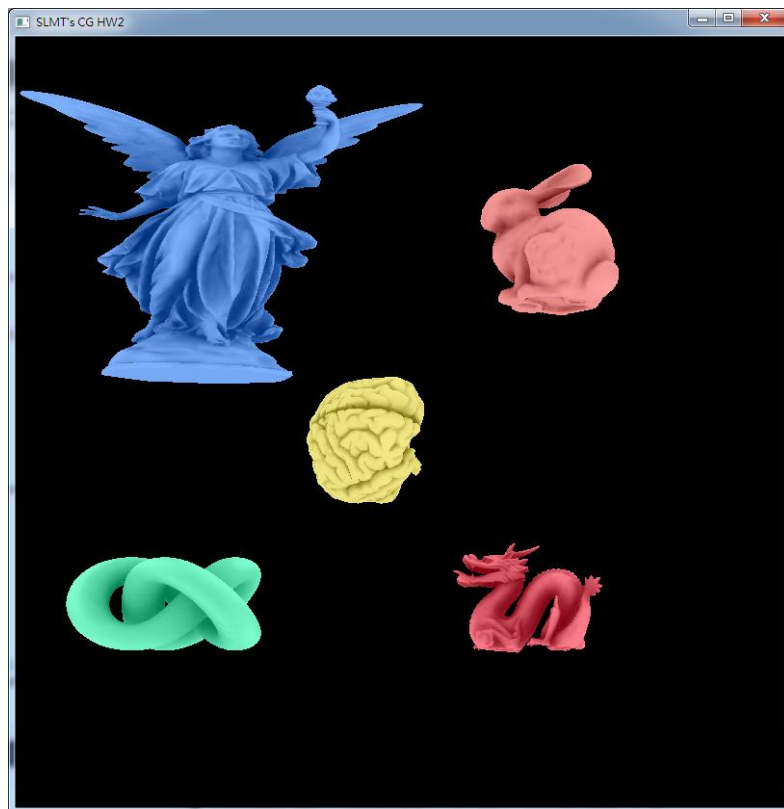
1. Orthogonal Projection



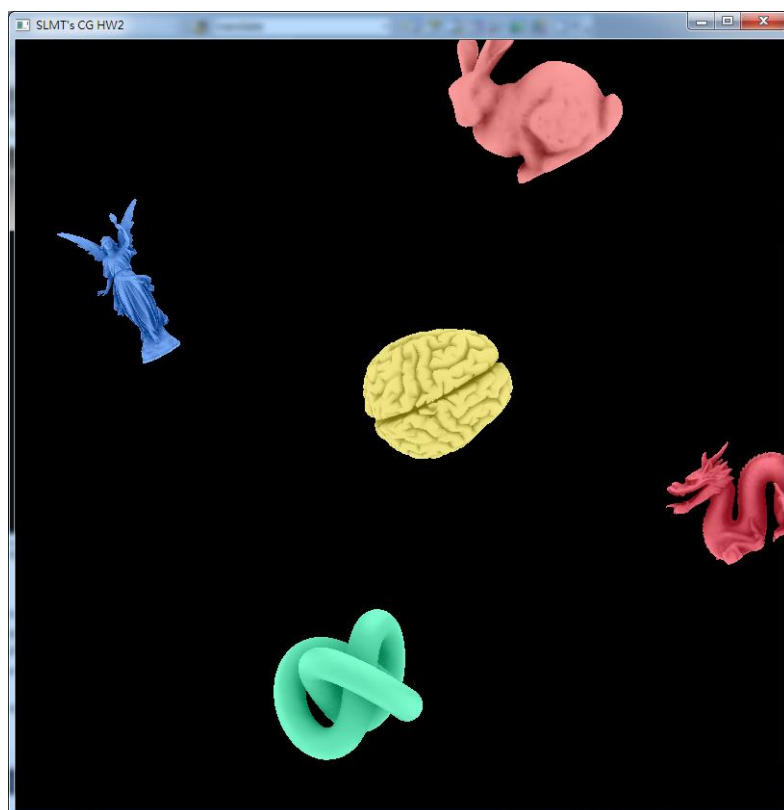
2. Perspective Projection



3. Model Transform



4. View Transform



五、心得

這次作業花最多時間的是在設計架構吧。因為想要設計出方便讓多個 model 共存的架構，又可以在以後快速加入新功能。因此就設計出現在這樣簡潔漂亮的架構。不過代價就是遲交一周，而且還為了打這份報告又多遲交十幾分鐘，所以總共遲交兩周 (扣 6 分 Orz)