1. Explain the whole program’s structure.

只要符合OpenGL所規定的格式，調換順序是沒有問題的，在main函式中第一步是設定基本的視窗，這部需要設定一些視窗大小，視角位置等；而再來要設定呼叫的函式，在最後運行時，參考的函示都會依照這步所設定的，這步基本上會影響最多程式的獨特性；或著可以做初始化，這部只需要執行一次，和第一步差不多，就是做些個人的設定。

1. How do you implement the revolution and rotation by glPushMatrix() and glPopMatrix()?

第一步畫出太陽，太陽就在原點故也不用去紀錄。

drawSun();

第二步讓陣列記住位移的方式，但沒有物件只需要單純位移，這部開始有用到time變數，以模擬時間。

glRotatef(X / 365.0f \* time, 0.0f, 0.0f, 1.0f);

glTranslatef(18.0f, 0.0f, 0.0f);

glPushMatrix();

第三步加上自轉和傾斜即符合地球的運動方式，並再次紀錄。

glRotatef(23.5f, 1.0f, 0.0f, 0.0f);

glRotatef(X / 1.0f \* time, 0.0f, 0.0f, 1.0f);

drawEarth();

glPushMatrix();

第四步把地球自轉的部分退掉到第二步，讓軸對齊原點，只要平移即可。

glPopMatrix();

glTranslatef(0.0f, 0.0f, -2 \* Y);

drawAxis();

glPushMatrix();

第五步再退回去第二步，先做完繞地球公轉再移過去即能符合月球的運行。

glPopMatrix();

glRotatef(X / 28.0f \* time, 0.0f, 0.0f, 1.0f);

glTranslatef(3.0f, 0.0f, 0.0f);

glRotatef(X / 28.0f \* time , 0.0f, 0.0f, 1.0f);

drawMoon();

匯出即可呈現在畫面上。

glutSwapBuffers();

1. How do you draw the planets?

我把它分成兩部分，頂點的扇形和中間的條形，分別使用glBegin()的GL\_TRIANGLE\_FAN 和GL\_TRIANGLE\_STRIP，而我的stack是依據z軸去等分切，再來就能算出xy的座標，個人覺得比較方便，在依據slice的數目去等份繞出一個圓，這邊只要使用簡單的sin cos帶入(2\*PI\*n)就能算出。依照這樣的模式可以推出全球的xyz座標。

算出來後再帶入glBegin()並做些基本設定即可。

GL\_TRIANGLE\_FAN: 會以第一個加入的點作為起始點，和之後的每個連續兩點畫出一個三角形。

GL\_TRIANGLE\_STRIP: 從加入的第三點之後，每連續三點都會畫出一個三角形。

因為我們要繞一圈，所以這步是需要標到起始第1、2點的。

需要注意的是底端GL\_TRIANGLE\_FAN加入時也要注意要先畫出頂點並且要注意第一個三角形為逆時鐘轉，以確保顏色能在外層。