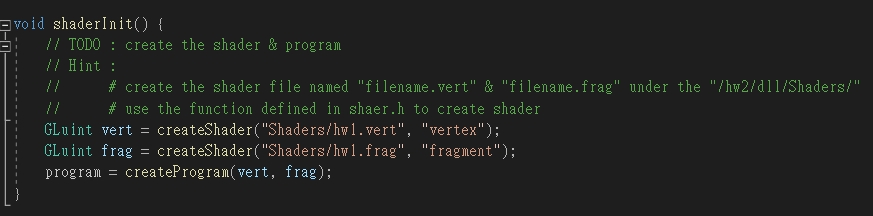
0816028 周孟謙

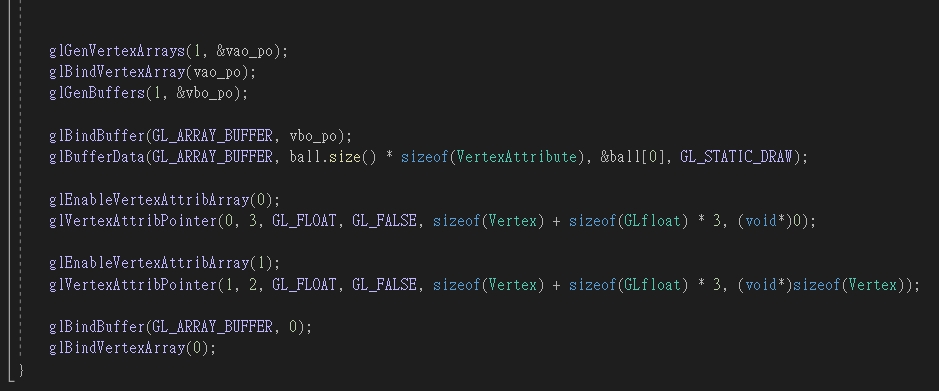
ShaderInit:

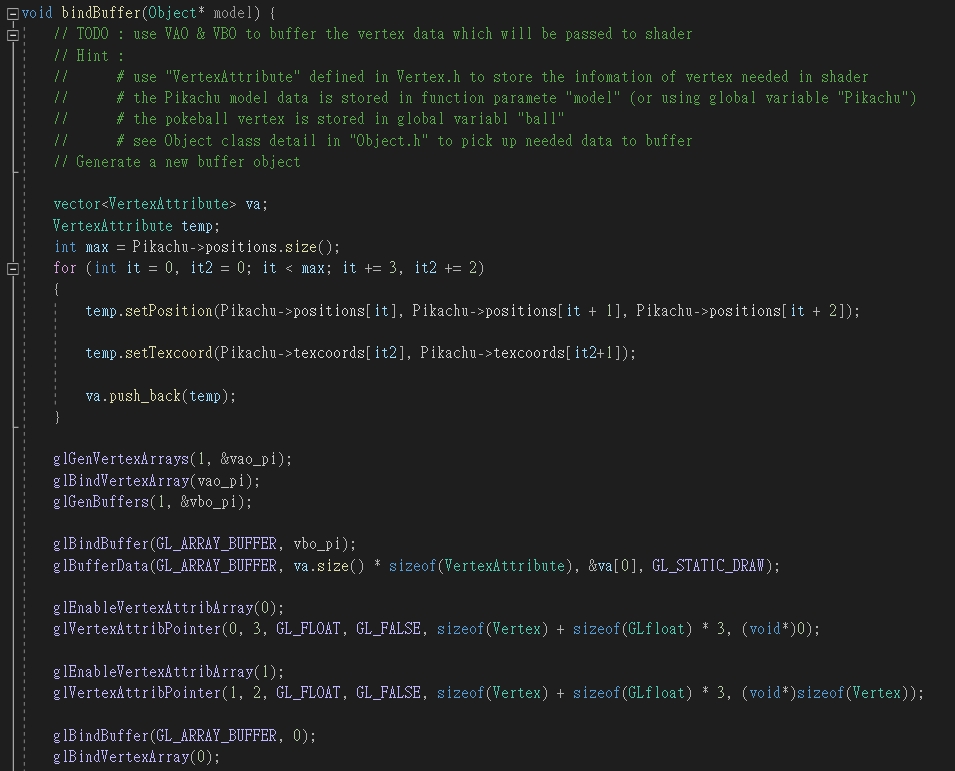
一開始先用glCreateShader創造vertex和fragment shader，再用他們去creates a program object。



BindBuffer:

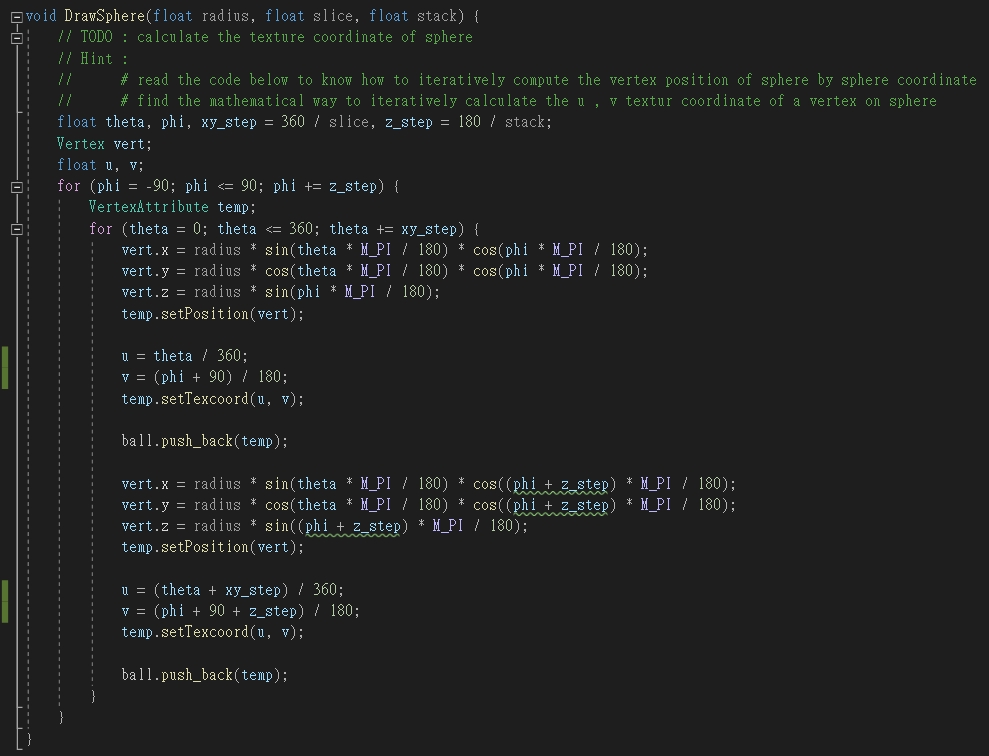
創一個vector<VertexAttribute> va，把皮卡丘的position和textcoords給放進去，使其之後能用glBufferData()複製到vbo\_pi裡面。接下來就是generate vao和vbo buffer並bind。其中vao pointer對vbo的指法為前三個是target的x, y, z，開頭為(void\*)0是因為在整個buffer的最前面。後面兩個為texcoord的x, y，開頭為(void\*)sizeof(Veertex)是因為前面的xyz大小為一個Vertex，而stride皆為sizeof(Vertex) + sizeof(GLfloat) \* 3是因為每個VertexAttribute皆有宣告Vertex position和GLfloat texcoord[3]。Pokeball的原理跟pikachu的一模一樣，只是一開始資料已經被塞好進vector<VertexAttribute>了。





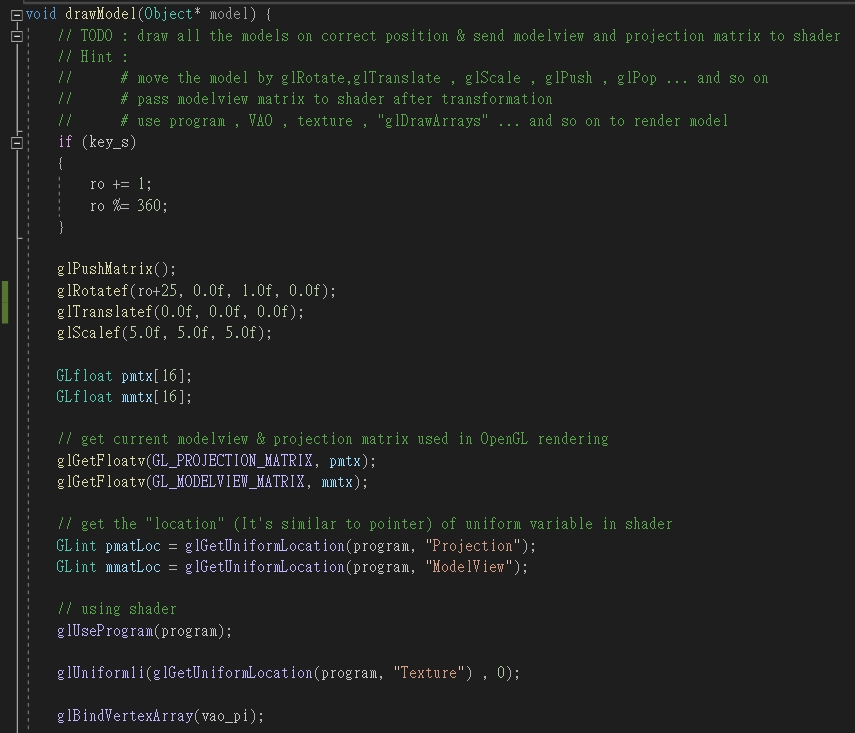
DrawSphere:

利用theta從0到360把u從0到1掃一遍，再利用phi從-90到90把v從0到1掃一遍，把textcoord也塞進ball的陣列。而xy\_step和z\_step遇到的鋸齒狀問題會在後面講解。

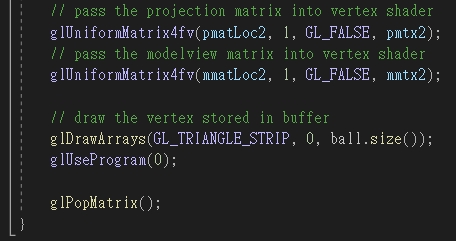
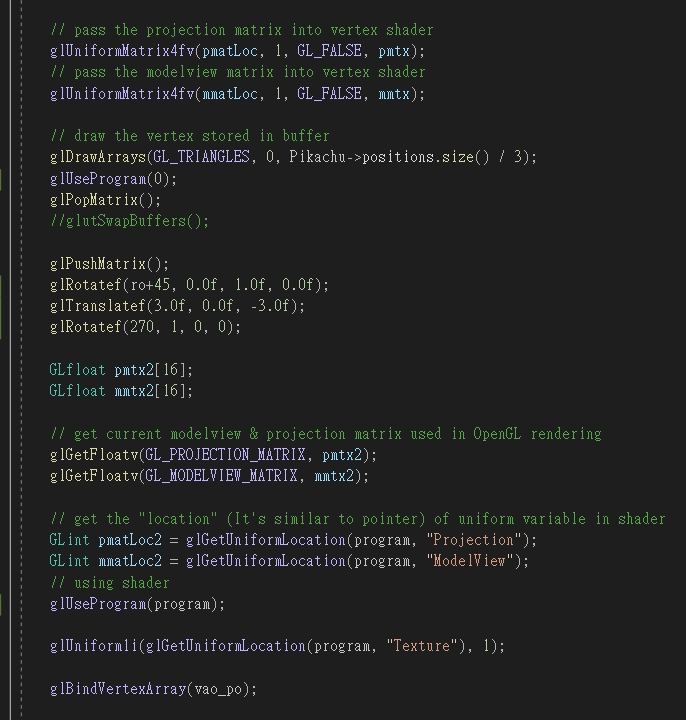


DrawModel:

ro為按鍵s的旋轉角度(1 degree / per frame)。接下來是先畫出皮卡丘，把旋轉、移動、放大依照pdf給的數值弄好，然後創建並連結modelview & projection matrix後(glGetFloatv)，用glGetUniformLocation來get the "location" of uniform variable in shader，再把牠們放進shader(glUniformMatrix4fv)，Texture則是用glUniform1i放進shader。之後便bind vao\_pi，使glDrawArrays(GL\_TRIANGLES, 0, Pikachu->positions.size() / 3)可以draw the vertex stored in buffer，使用GL\_TRIANGLES是因為GL\_TRIANGLE\_STRIP會不斷用之前兩點加上新的一點來畫三角形，造成皮卡丘的突出部分之間會有絲，Pikachu->positions.size() / 3則是因為position每個單位不是存一點，而是一點的x or y or z。



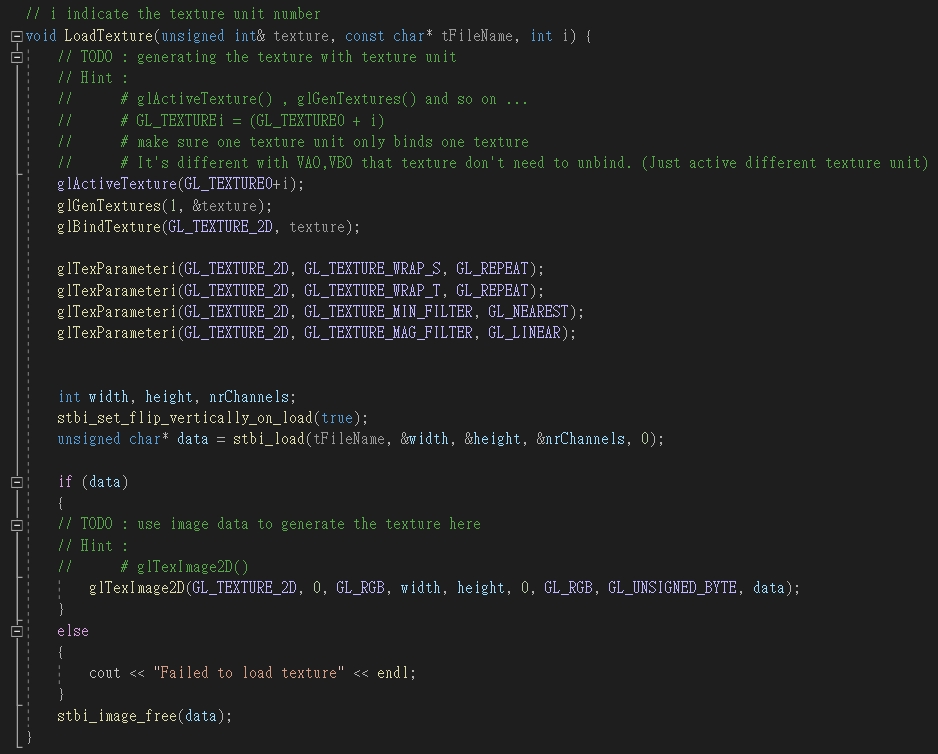
球的部分跟皮卡丘幾乎相同，但需要注意的是球因為之前DrawSphere掃u和v的時候沒有注意方向，所以需要另外加上glRotatef(270, 1, 0, 0)轉到正面。還有glDrawArrays是使用GL\_TRIANGLE\_STRIP，原因正好跟皮卡丘相反，若是使用GL\_TRIANGLES 則反而會造成球面會有洞。



LoadTexture:

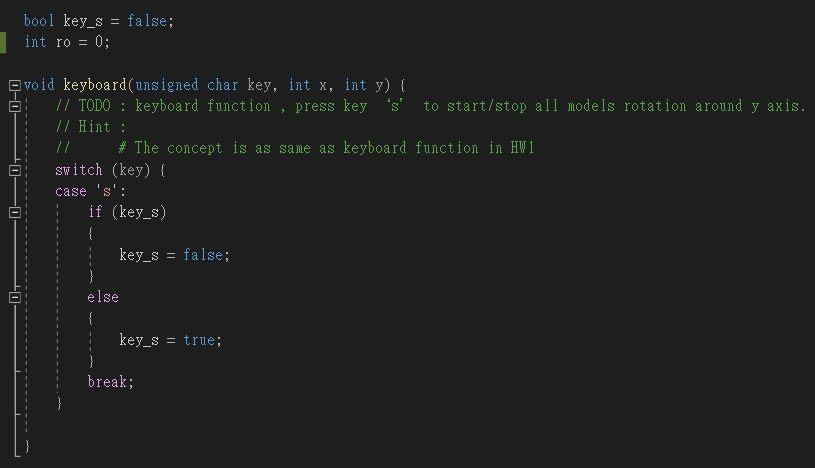
用glActiveTexture去selects texture unit。然後glGenTextures去Takes as input one textures we want to generate and stores them in a unsigned int array(texture)，

再bind它。 glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT); glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT); glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST); glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);則是為了Texture wrapping和Texture filtering。最後用glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGB, width, height, 0, GL\_RGB, GL\_UNSIGNED\_BYTE, data);來使用圖像數據生成texture。

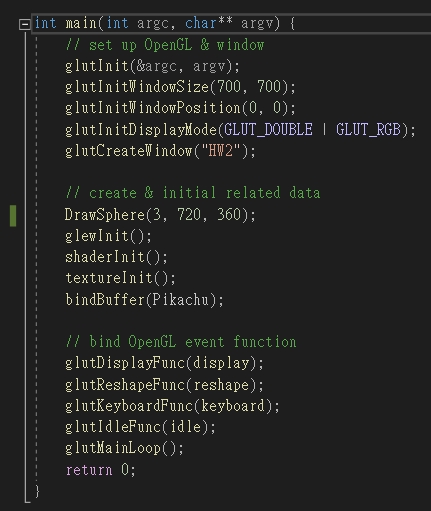


Keyboard:

設定鍵盤按鍵s，每按一下會開始/暫停旋轉。

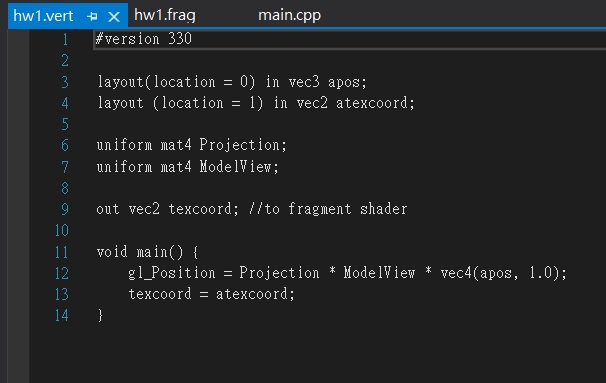


之前說過xy\_step和z\_step如果太大的話球會跑出鋸齒狀，因此在這邊DrawSphere加大了slice和stack的數字，使得xy\_step和z\_step變小。



Vertex shader:

把vertex的position和texcoord讀進來，把apos從vec3變成vec4再乘上Projection和ModelView傳給gl\_Position。以及把texcoord傳給fragment shader。



Fragment shader:

輸入texture和textcoord，再用texture2D來找出texture再textcoord位置的顏色後輸出。

