B11132021　四資工二甲　蔣博元

圖學導論 Project03 – checkpoint 02

目次

TOP

API 文件

ArcBall.h

包含 class ArcBall，用來表示球面座標系。

DynamicHeightMap.h

包含 class DynamicHeightMap，用來動態模擬水面漣漪的 height map。

qtTextureCubeMap.h

包含 class qtTextureCubeMap，透過 QImage 來載入 6 張圖片形成 Cube Map。

qtTextureImage2D.h

包含 class qtTextureImage2D，透過 QImage 載入 2D 的圖片型成 Texture2D。

Shader.h

包含 class Shader，用來載入及編譯 shader。

UBO.h

包含 class UBO，包裝 Uniform Buffer Object。

VAO_Interface.h

包含 class VAO_Interface，定義 VAO 物件的基本 API。

Box_VAO.h

包含 class Box_VAO，可繪製一個以原點為中心的正方體，邊長為 2 * SIZE。

Plane_VAO.h

包含 class Plane_VAO，可在 2D 平面繪製(-1, -1) ~ (1, 1)的正方形，在 NDC 座標中，這範圍代表整個螢幕。

Wave_VAO.h

包含 class Wave_VAO，可繪製一個以原點為中心的平面正方形（落在 xz 平面），邊長為 SIZE。該平面被切成許多的三角形，以方便在 vertex shader 中實作水面的起伏。

TOP

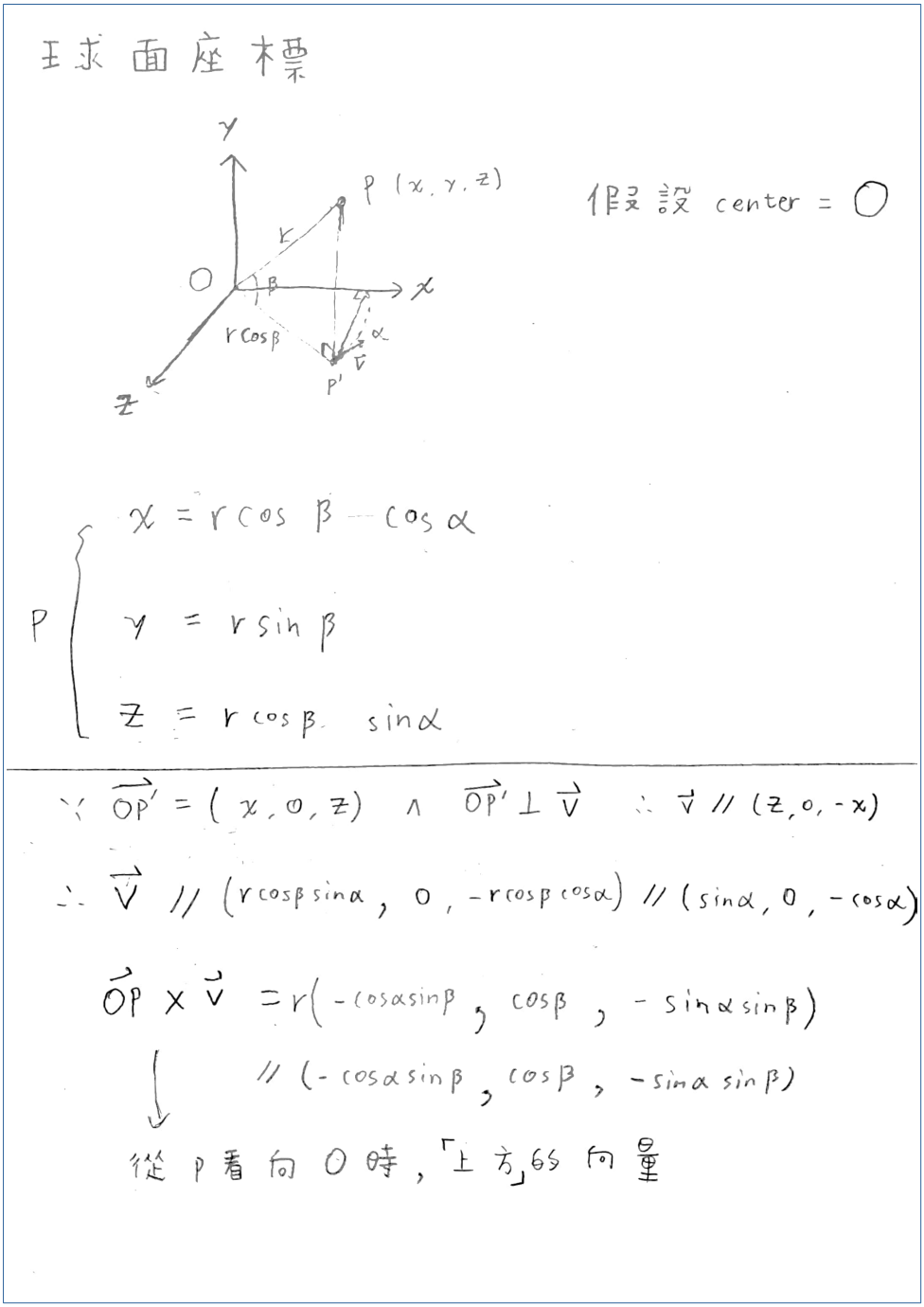
演算法

Arc Ball

我用了兩個角度 alpha、beta、球的半徑 r、以及球心 center 來控制點（眼睛）的位置。

角度 alpha 是方位角，而角度 beta 則是仰角。

在計算從眼睛看向 center 的 view matrix 時會需要指示上方在哪的 up 向量。所以在下圖中我先算了 v 向量（指示了右邊的方向），接著計算 OP 向量和 v 向量的外積得到了需要的 up 向量。

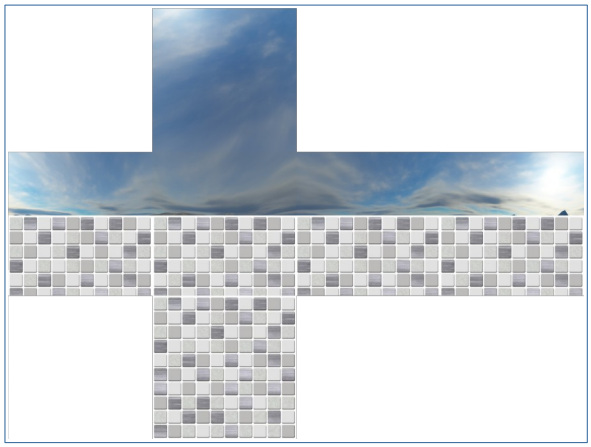


TOP

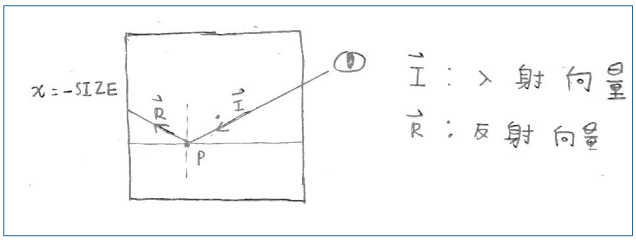
B11132021　四資工二甲　蔣博元

水波的反射、折射

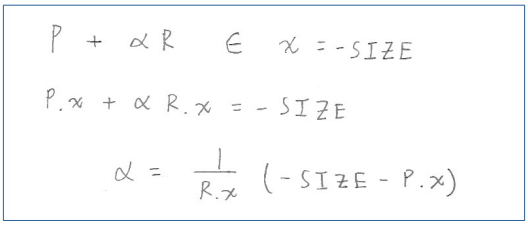
我事先做了一個以原點為中心的 cube map，包含了天空和瓷磚。



然後再用類似 ray tracing 的方式看反射（或折射）的光線會打到 cube map 的哪裡。



上圖中，射向 P 點的反射光很顯然地會落在 cube map 中 x = -SIZE 的那一面，所以我們可以假設反射光和該面的交點為 P + alpha * R，並用下圖的方式求出 alpha。



因為我的水池和 cube map 都是以原點為中心，所以算出的交點可以直接當作材質座標從 cube map 中取出 P 點要顯示的顏色。

在實作時需要考慮 P 沿著 R 向量前進時可能會碰到哪些平面並計算交點，再從這些交點中取和 P 最近的。

透過 R 的 xyz 的正負號可以判斷可能會碰到哪些平面，當 R 的(x, y, z)為(+, -, +)時那麼 P 沿著 R 前進就可能碰到 x = SIZE、y = -SIZE、z = SIZE 這些平面。

TOP

B11132021　四資工二甲　蔣博元

水面的法向量

我使用 geometry shader 來計算水面的法向量。geometry shader 會傳入水面上某個三角形的三個頂點，可以用這三個點求出兩個向量並外積得到法向量。

TOP