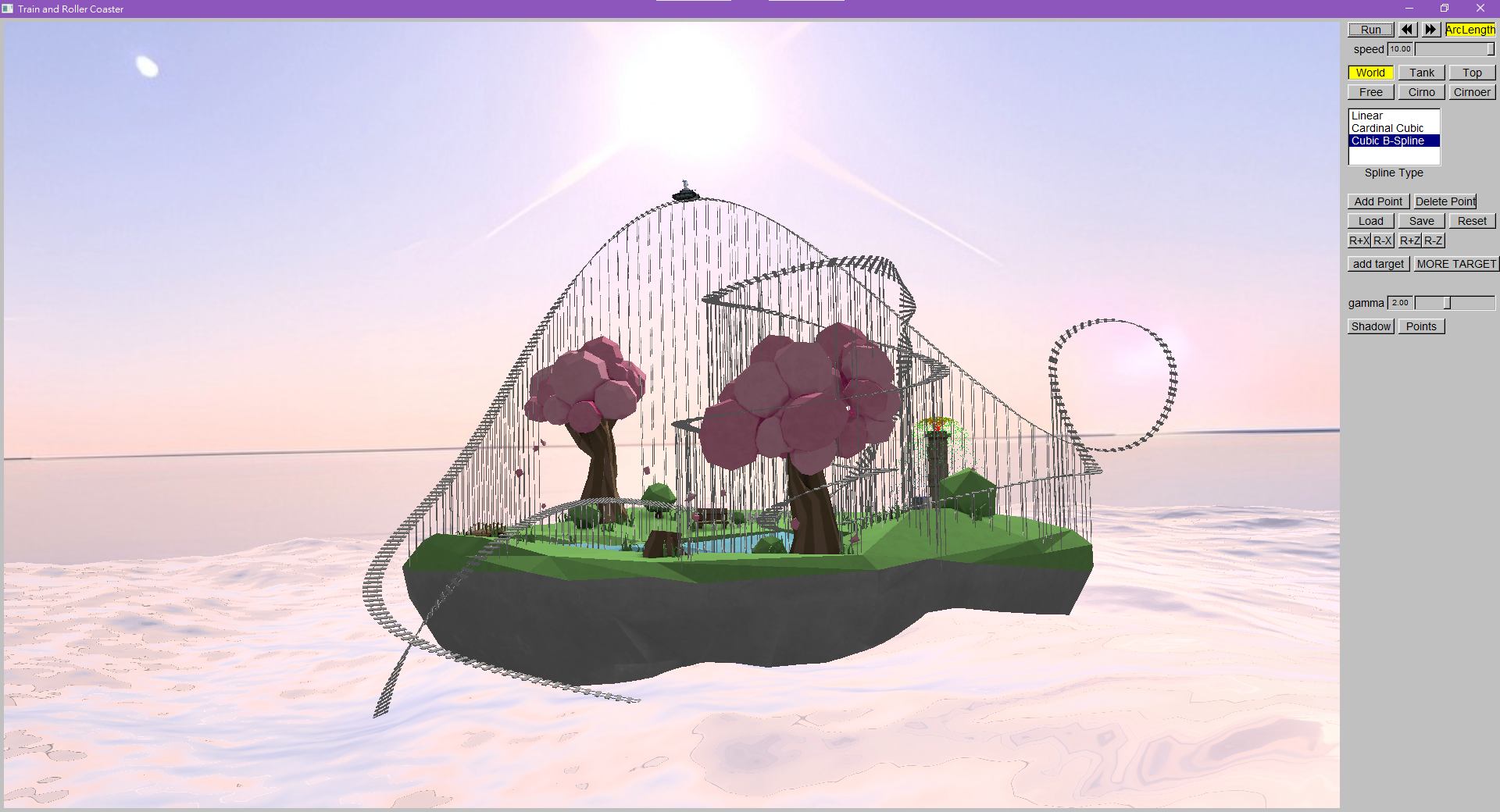
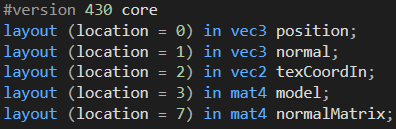
**圖學導論專案3 技術文件**

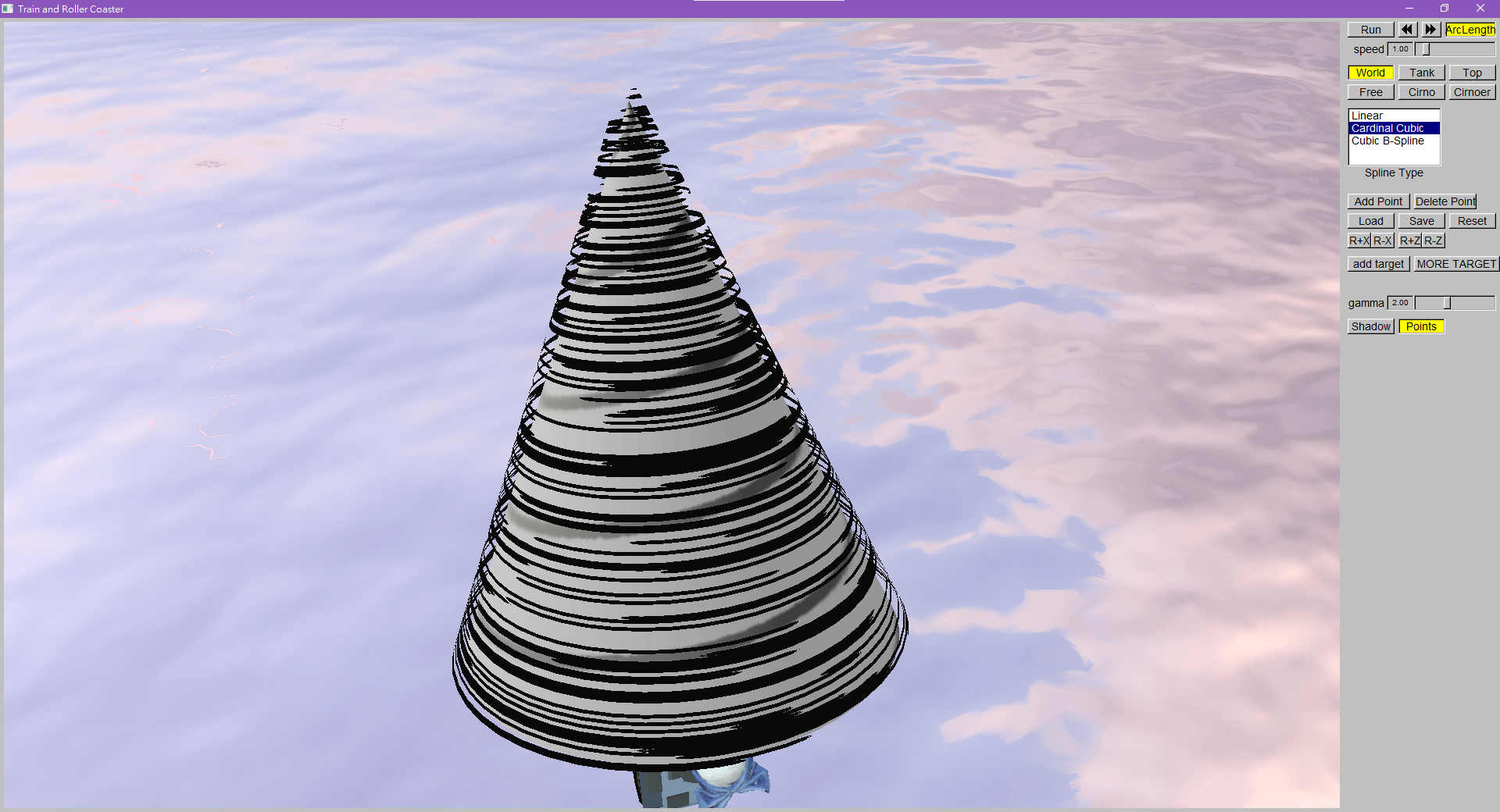
B11215036 張維元 B11215059 翁世承



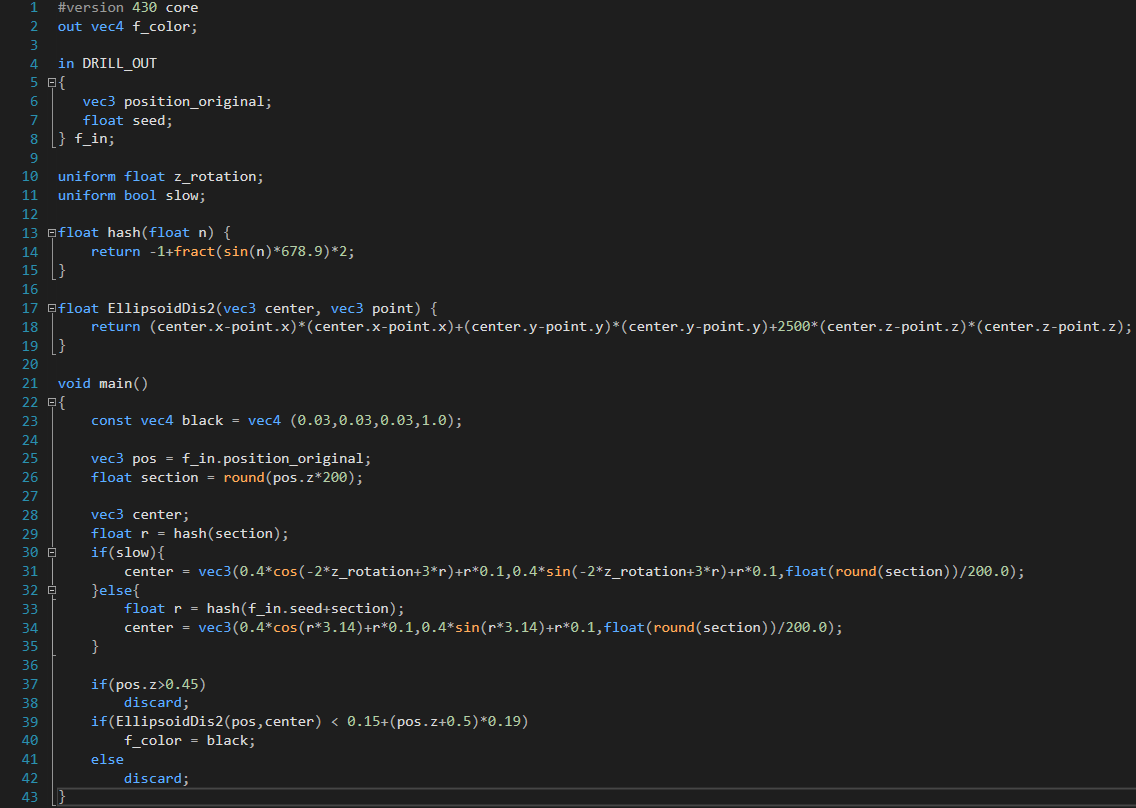
* 模型導入：
* 水面動態：
* 粒子特效：
* Draw by instance：對於大量重複出現的簡單物件──例如鐵軌、枕木、砲彈，以及標靶，我們使用了glDrawElementsInstanced，先做好一個單位實例(例如1\*1\*1的立方體)，再傳入各物件的模型矩陣給shader，讓他們進行伸展、旋轉……。如此便能利用GPU平行運算物件的世界座標，加快運算效率



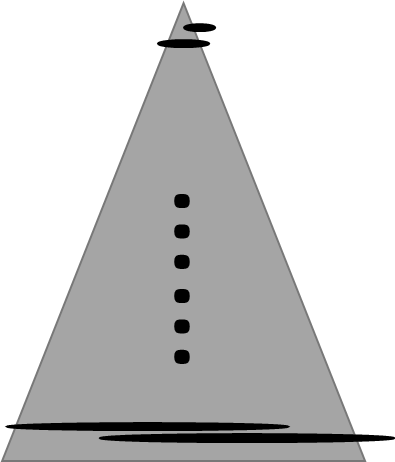
* 旋轉的鑽頭：

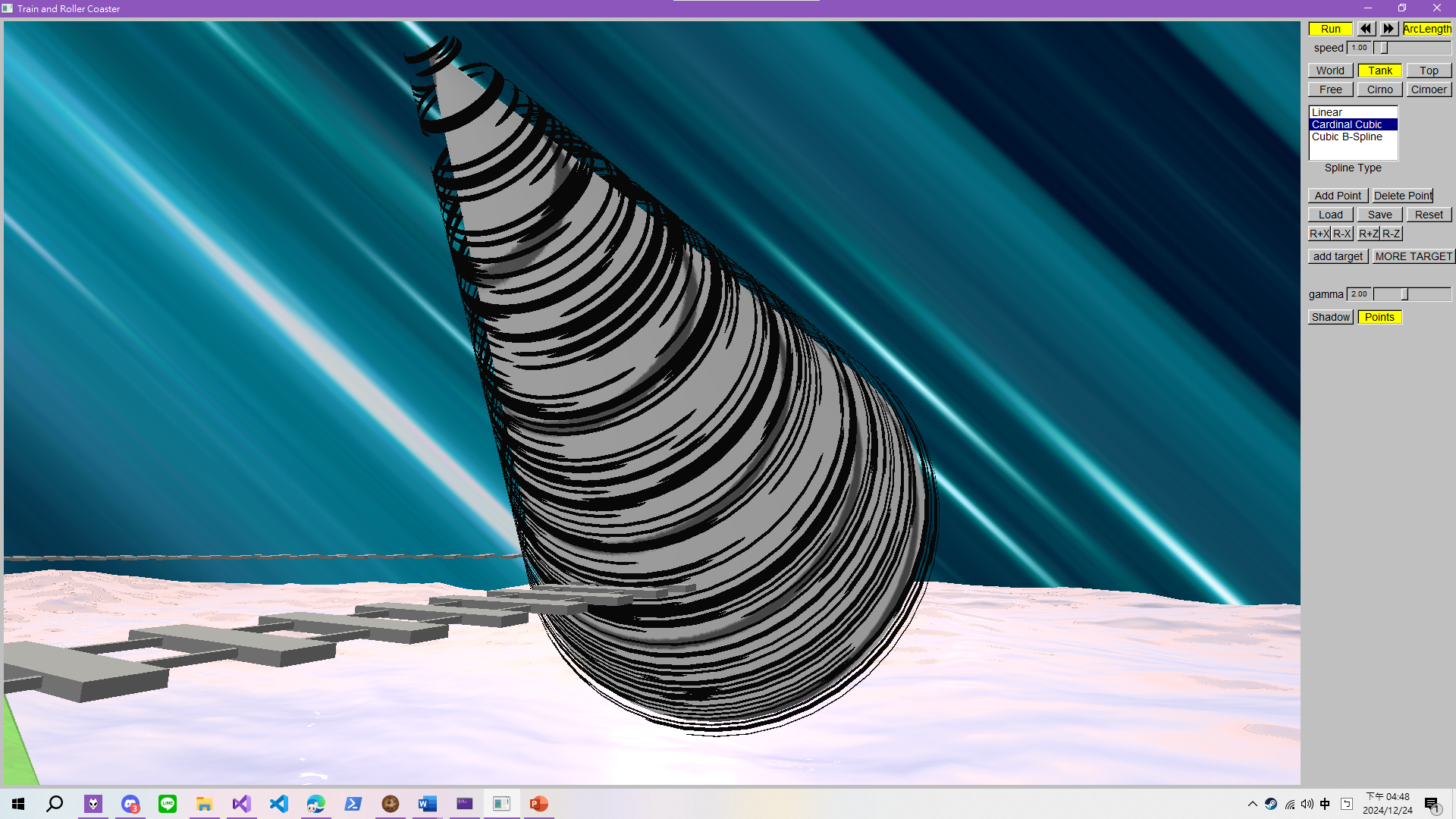


包圍在外的黑色線條，其實就是一個大一號的黑色圓錐，只不過我們在片段著色器上動了點手腳

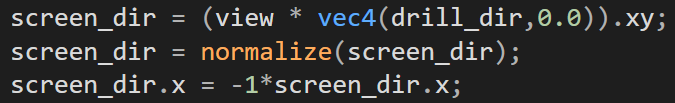
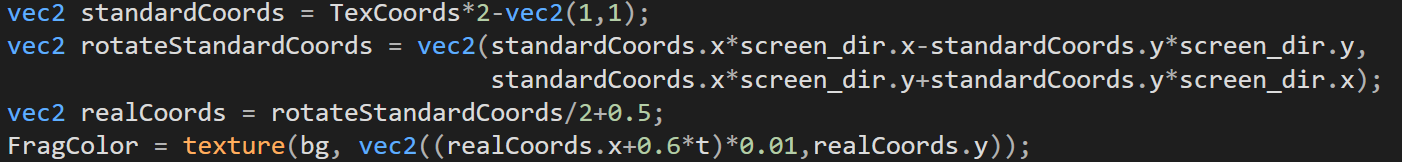


其核心原理大概是：首先把圓錐從頭到腳切成200段，對於每一個區段，都有一個隨機的圓心，從那個圓心畫出一個扁球面(因為是錐形，半徑也會跟著區段有所不同)，只要在球面內的部分就塗上黑色，否則discard。畫成平面側視圖的話大概就是下面這樣的感覺：



* 速度線：  
  把一張圖片拉得很長，再由左至右掃略而過，就能做出這樣的感覺。而在shader中，只要調整texture採樣的位置，再將超界處理設為鏡像即可。

速度線的方向也會跟隨鑽頭的指向，這點只要把火車頭的向量(w=0)換算成實際在螢幕上的指向，再取其xy值即可。

順帶一提原本的圖片是這張，超級機器人大戰X在Steam的預設背景，買來還沒通關的遊戲能先在圖學專案派上用場真是太好了呢。

* 子彈時間效果



這又能大致拆成兩個部分：

其一、簡單顏色處理，總之我們疊了各種濾鏡，讓它看起來盡可能酷。  
(就算仔細想來那或許和減慢時間沒什麼關聯)

* + 降低飽和度：將原先的顏色與灰階後的結果進行混合
  + 黑色邊框：根據螢幕座標位置與黑色進行不同程度的混合
  + 暈影效果(可以關注上圖中的藍色粒子)：往螢幕中間的方向採樣並混合，就能達到殘影向外擴散的效果。

其二、標靶雷達效果

我們將標靶的z軸畫成深度圖，並在後處理時畫出該深度圖的邊緣(其值變化的交界處)。由於深度圖的精度有限，標靶並不會整個變成白色，而是在深度圖最小精度的邊緣上被畫上白線。這其實是偶然力下的結果，但我們都覺得它看起來真的很酷。

