

資工 3B 王嘉羽 00957116

● 介紹:

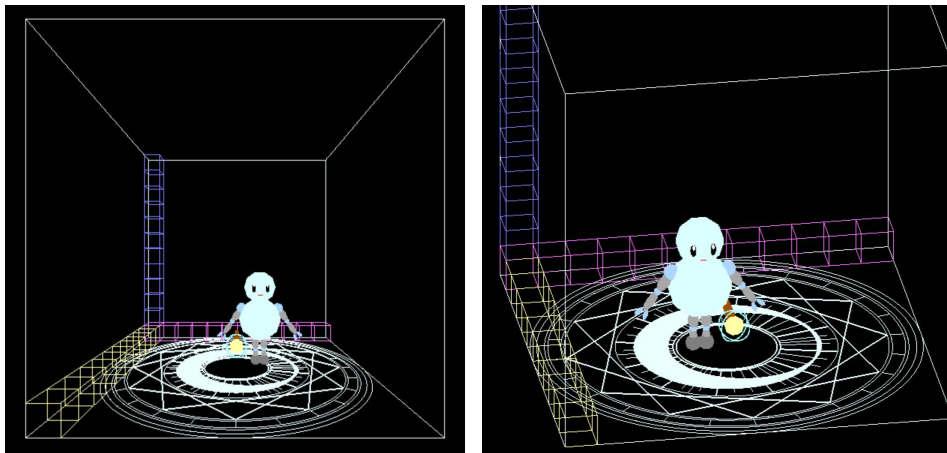
這次專案主題是在第二次的作品上加入相機，可以看整個場景。這次的作業要求大致可以分為 4 種:

1. 畫出世界坐標系的 xyz 軸
2. 對鏡頭(視野)做 6 個自由度運動
3. 畫出 view volume
4. 用 key 切換投影 method

● 設計理念:

我想了很久，要如何讓上面那 4 點融合進我的作品，且一點也不刻意。但我都沒有想出來...尤其是第一點..我原本是一個魔法陣要自然加入 3 軸很難想耶，我原本有想到我可以召喚出一個類似指南針的魔法書，但作業要求要有原點，所以我又沒有想法了..

最後決定按按鈕生成一個空間，然後在原點顯示出 3 軸。



然後關於鏡頭的部分，我做了一個小精靈~



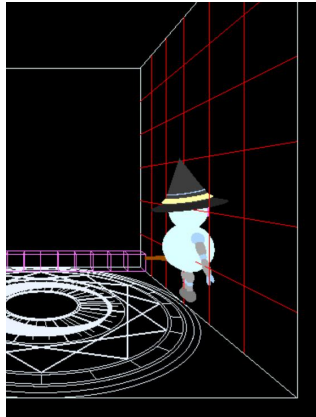
→有利於加分的因素，用紅字標(雖然這次好像沒有 fancy idea)。

- 操作說明:

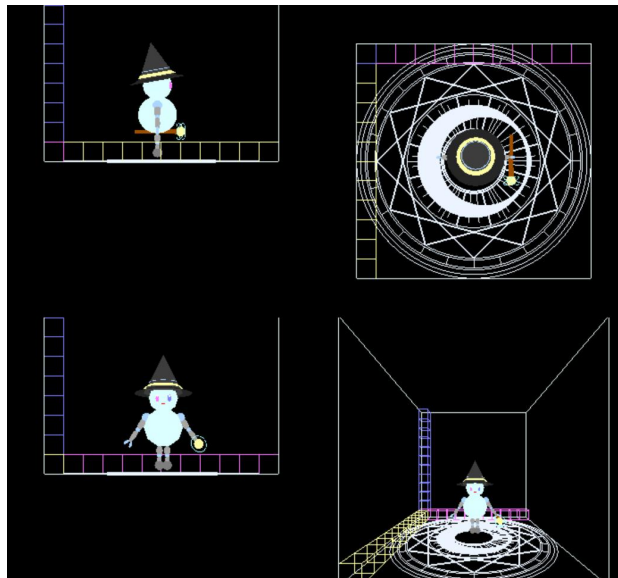
- 顯示 xyz 軸(X/x)

- ◆ 特別效果:

動畫效果出現 3 軸，碰到邊邊會有警示效果燈(一閃一閃)



- 切換 5 projection methods (Y/y)



|        |        |
|--------|--------|
| 平行 x 軸 | 平行 y 軸 |
| 平行 z 軸 | 透視投影   |

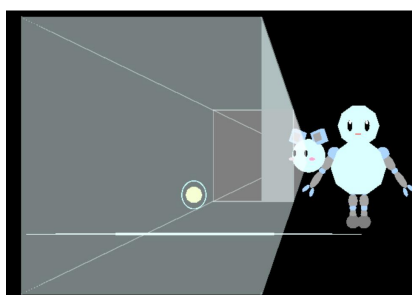
每次按都會切換成不同的 x -> y -> z -> 透視 -> 4 格 -> 跟著機器人

\*如果是在草原的場景只有 透視 和 跟著機器人 (因為 4 格很卡...)

- 顯示相機(小精靈) (Ctrl + p)

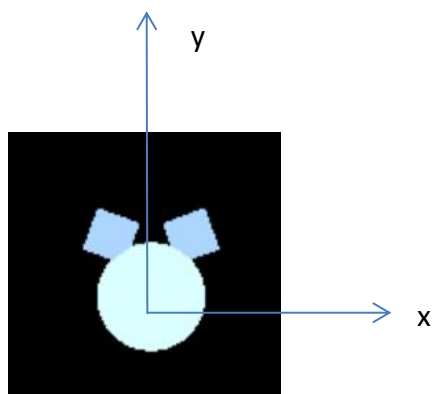
- 將相機和 view volume 初始化為原本的地方 (Ctrl + backspace)

- 顯示 view volume (Ctrl + o)



## ■ 移動小精靈

小精靈背面  
(面向的方向定義為 +z)  
(右手定義為 +x)  
(頭頂定義為 +y)

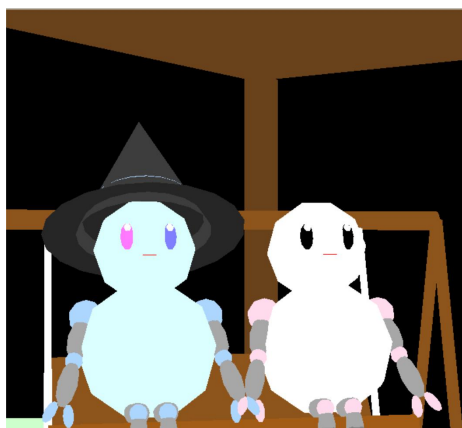


|          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
| +x 平移    | -x 平移    | +y 平移    | -y 平移    | +z 平移    | -z 平移    |
| Ctrl + d | Ctrl + a | Ctrl + w | Ctrl + s | Ctrl + q | Ctrl + e |

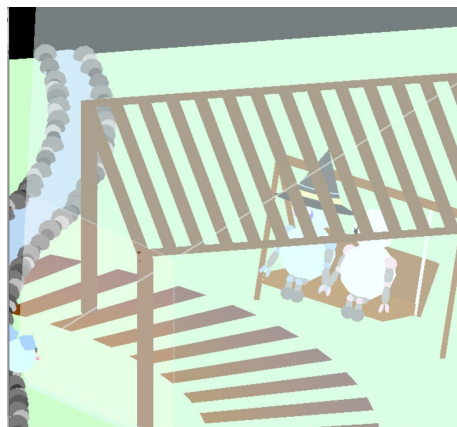
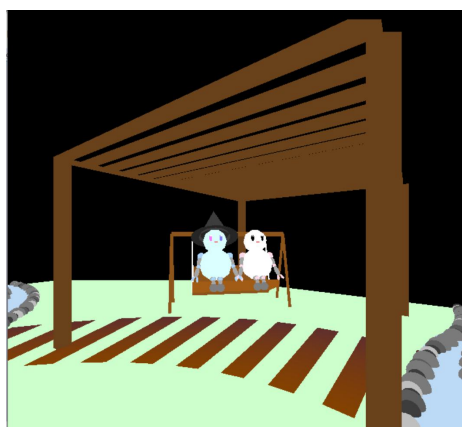
|                 |                |                |
|-----------------|----------------|----------------|
| x 軸旋轉(pitching) | y 軸旋轉(heading) | z 軸旋轉(rolling) |
| Ctrl + x        | Ctrl + y       | Ctrl + z       |

|          |          |          |          |
|----------|----------|----------|----------|
| zoom in  | Zoom out | 看得更遠     | 不看那麼遠    |
| Ctrl + c | Ctrl + v | Ctrl + b | Ctrl + n |

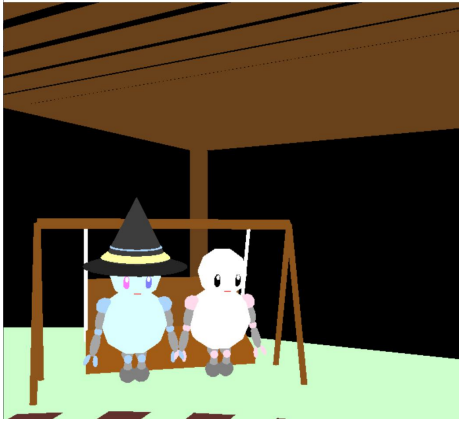
## ◆ Zoom in



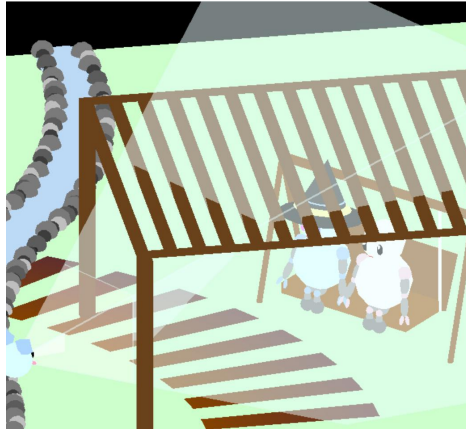
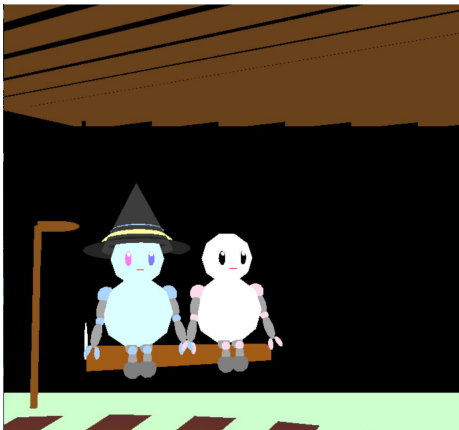
## ◆ Zoom out



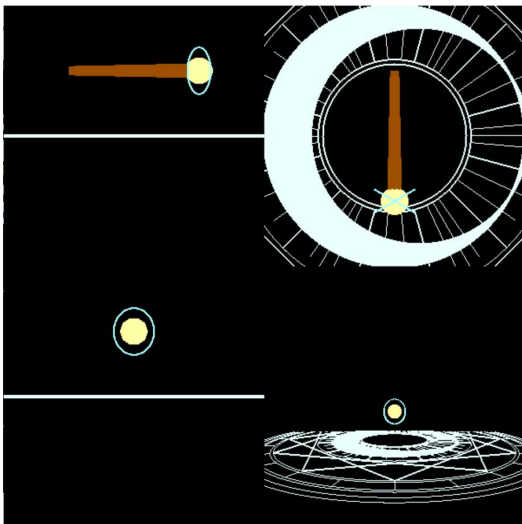
◆ 看得遠



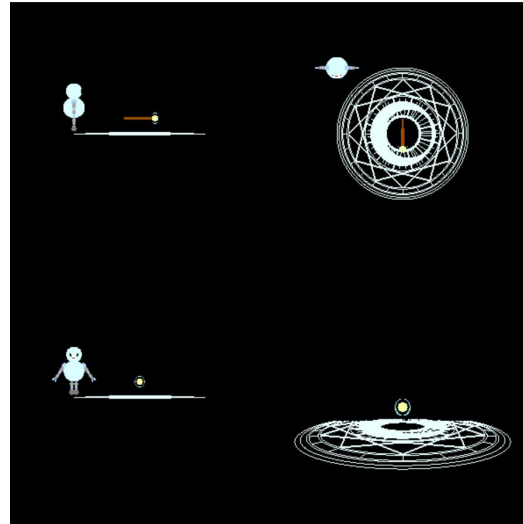
◆ 看得不那麼遠



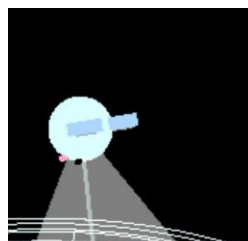
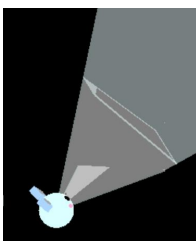
◆ 平行投影



Zoom in



Zoom out



我的小精靈會跟著轉歐!

- 心得:

這次的東西好難，那個 **view volume** 我研究了好久，一開始我想說和攝影機畫在一起，然後對整個做旋轉 **xyz** 和平移，之後發現大錯特錯.....因為矩陣不具有交換律，所以先轉 **10x** 再轉 **20y** 再轉 **30x** 和先轉 **40x** 再轉 **20y** 不一樣.....之後我就放棄了這個寫法..然後轉而直接在 **wcs** 直接畫...然後又遇到很多理解上的錯誤，我不知道要怎麼旋轉..恩不過呢在研究之後我還是成功了，我把座標\***u** 矩陣，就是 **transformation** 的矩陣這樣我的平移就會自帶旋轉效果，就可以和眼睛一樣慢慢轉，不會一次轉 **xyz**。

但很不幸的是，我在寫完 **code** 後發現，原來 **opengl** 有函數可以把 **u** 矩陣直接乘進去.....嗚嗚我白推了，不過也算是有意外的收穫吧，之前因為只會用函式所以完全不懂數學概念，才會導致這次遇到的問題，但現在我會了!!讚啦!!以往作業都是讓我更了解 **opengl** 這一次我更了解了線性代數了，雖然過程很痛苦~

最後為甚麼這次沒有 **fancy idea** 的加分啦嗚嗚 qq