

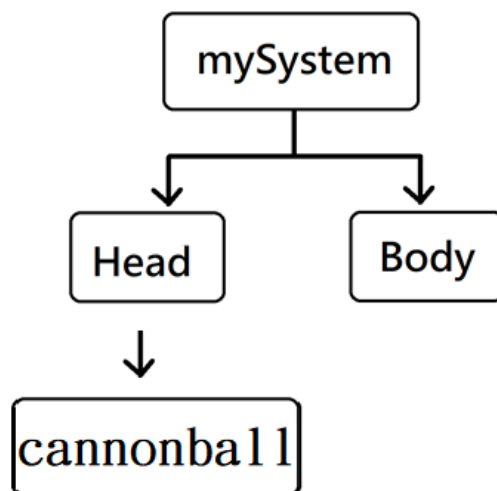
# 基本功能

- BB8組成

由一大一小的Sphere mesh所組成



簡易階層式架構



# 基本功能

- 前後移動(W、S)

按下W前進，S後退

並於放開時，根據慣性，繼續運動一小段時間

前進時身體往前滾動，後退時往後滾動

- 左右轉動(A、D)

按下A往左邊轉動(逆時鐘)

按下D往右邊轉動(順時鐘)

可於運動時按下WASD任意組合並做出對應的動作

# 基本功能

- 滑鼠控制

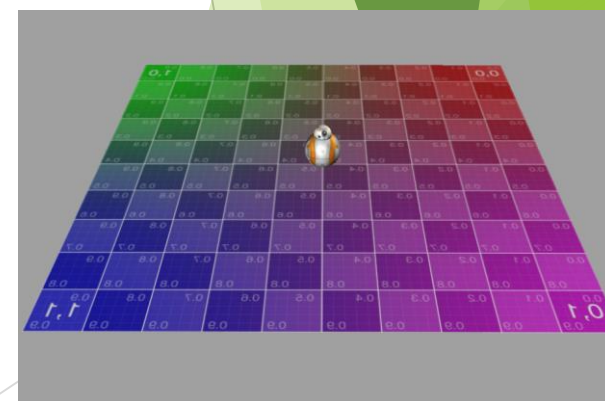
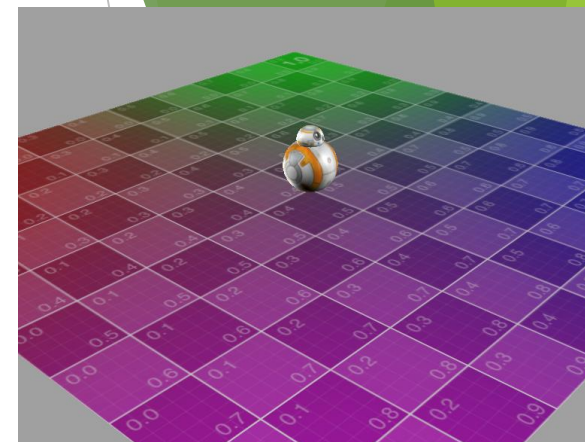
(左鍵按住拖曳控制視角、滾輪控制縮放)

script來源：

```
<script src="js/controls/OrbitControls.js"></script>
```

```
controls = new THREE.OrbitControls( camera, renderer.domElement );  
controls.enableKeys = false;
```

**Warning:** 使用時須把預設的enableKeys關掉  
在進行wasd操作時才不會有衝突

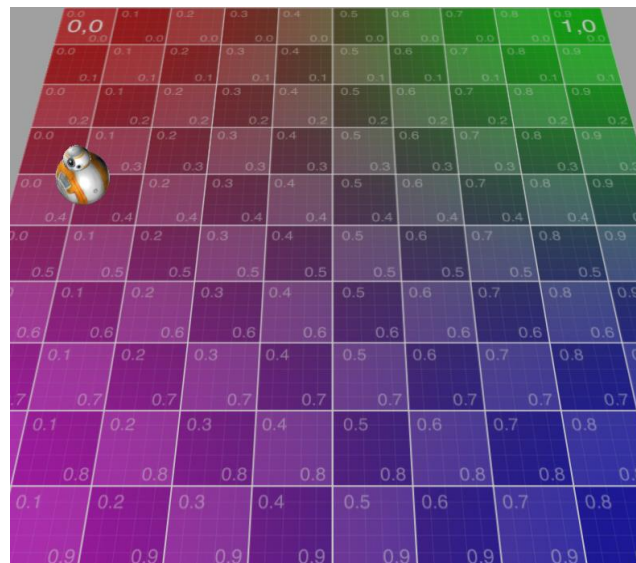
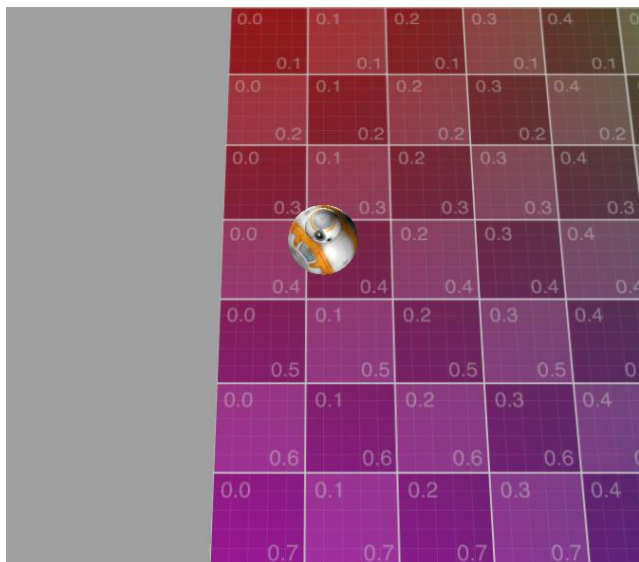


# 基本功能

- Follow me功能

勾選時，跟隨BB8且置於畫面正中央

取消時，相機回到初始位置



# 介面

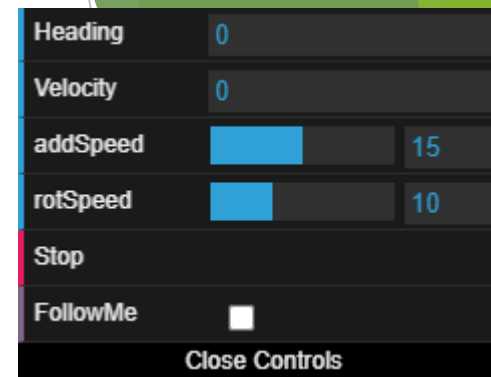
- 顯示：

## 車輛方向(Heading)

轉動時，往順時鐘方向遞增，逆時鐘方向遞減  
範圍為 0 ~ 360

## 目前速度(Velocity)

持續按壓W加速，持續按壓S減速，  
速度為正時，表示往前移動；為負時，表示往後移動  
在放開時，逐漸回歸至0  
根據調整的加速度，有不同的速度上限



# 介面

- 可調整：

## 車輛移動(addSpeed)

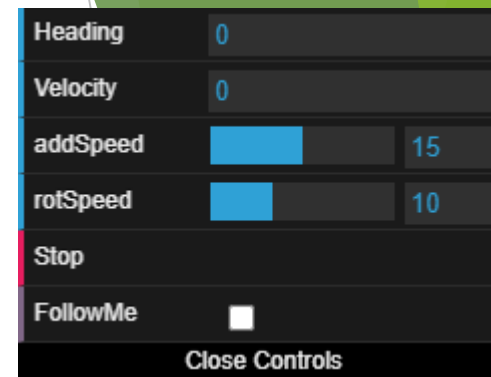
數值越高，運動時的加速度越快

數值越低，運動時的加速度越慢

## 轉動速度(rotSpeed)

數值越高，自轉速度越快

數值越低，自轉速度越慢



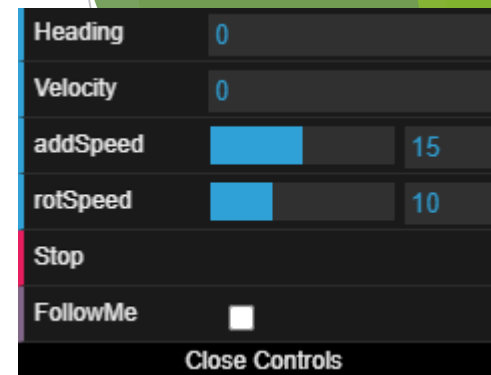
# 介面

- Stop

速度設為0，運動停止

- FollowMe

勾選時，跟隨BB8且置於畫面正中央  
取消時，回到相機回到初始位置



# 運動設計

- 運動範圍

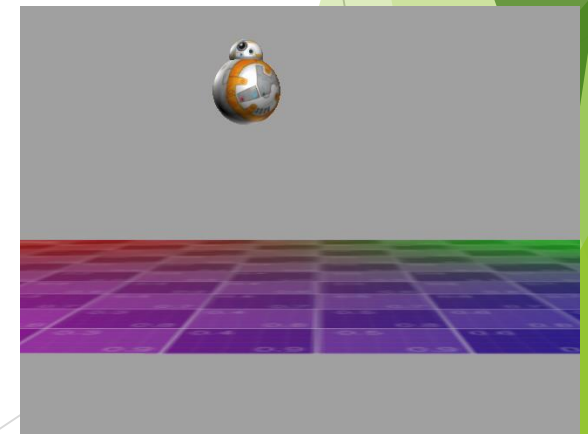
在走到地板邊緣時，生成比原本較大的地板

- 跳躍

按下1次空白鍵彈跳一次，

且可持續於空中進行運動(WASD)，

最多彈跳2次，且需等BB8降至地面時  
才可繼續跳躍動作





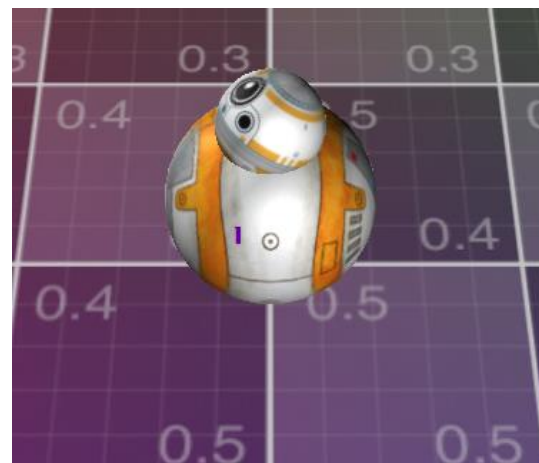
# 額外功能

- 頭部額外轉動(方向鍵 ↑ ↓ ← →)

上、下鍵控制頭部垂直轉動

左、右鍵控制頭部水平轉動

並在運動時維持其方向



# 額外功能

- 發射砲彈(enter鍵)

砲彈會從眼睛旁射出，並符合物理特性(協拋運動)

水平軸：

等速度運動： $v=v_0+at$

垂直軸：

等加速度運動(自由落體)： $s=v_0t+(at^2)/2$

由於使用requestAnimationFrame

只需計算每個frame的位移量即可

