

國立虎尾科技大學

機械設計工程系

cd2023 2b3-pj3bg6 分組報告

BubbleRob 八人對戰

BubbleRob Eight-Player Battle

指導教授：嚴家銘老師
班級：四設二乙
學生：
莊雨薰 (41023203)
陳妤瑄 (41023204)
陳靚芸 (41023205)
陳澤瑜 (41023206)
楊舒涵 (41023207)
廖子儀 (41023208)
鄭翊均 (41023210)
徐佑寧 (41023255)

中華民國

112 年 6 月

摘 要

由於矩陣計算、自動求導技術、開源開發環境、多核 GPU 運算硬體等這四大發展趨勢，促使 AI 領域快速發展，藉由這樣的契機，將實體機電系統透過虛擬化訓練提高訓練效率，再將訓練完的模型應用到實體上。此專案是對雙輪車進行設計改良，以提升行進與對戰效能，採用 CAD 進行場景與多輪車零組件設計後，轉入足球場景中以鍵盤 arrow keys 與 wzas 等按鍵進行控制，對陣雙方每組將有四名輪車球員，且每兩人在同一台電腦上操作。

此專案是對雙輪車進行設計改良，以提升行進與對戰效能，採用 CAD 進行場景與多輪車零組件設計後，轉入足球場景中以鍵盤 arrow keys 與 wzas 等按鍵進行控制，對陣雙方每組將有四名輪車球員，且每兩人在同一台電腦上操作。

更多詳細內容可以到 <https://mdecd2023.github.io/2b3-pj3bg6> 了解。

Abstract

Due to the four major trends of matrix calculations, automatic differentiation techniques, open-source development environments, and multi-core GPU computing hardware, the field of AI has experienced rapid development. Taking advantage of this opportunity, physical mechatronic systems are being trained through virtualization to improve training efficiency, and the trained models are then applied to physical systems.

This project focuses on the design improvement of a two-wheeled vehicle, aiming to enhance its performance in movement and combat. CAD is used to design the scenes and components of the multi-wheeled vehicle. Afterward, the vehicle is placed in a soccer field scenario, and control is achieved using keyboard arrow keys and other buttons such as WASD. Each team will have four players controlling the vehicles, with each pair operating on the same computer.

目 錄

摘要	i
Abstract	ii
第一章 目標	viii
1.1 規格	viii
1.2 規則	viii
第二章 球員	ix
2.1 組裝	ix
2.2 背號	x
2.3 程式碼	xi
第三章 記分板	xviii
3.1 繪圖	xviii
3.2 組合	xix
3.3 組裝	xx
第四章 場景	xxix
4.1 匯入球場球門球員記分板	xxix
第五章 跨網路連線	xxx
5.1 設定連線步驟	xxx

5.2 程式碼.....**xxxi**

圖 目 錄

圖 1.1 足球場景	viii
圖 2.1 導入球員 STL 檔	ix
圖 2.2 爆炸分解	ix
圖 2.3 更改顏色	x
圖 2.4 更改物件名稱	x
圖 2.5 新增 Joint	x
圖 2.6 調整 Joint 大小	xi
圖 2.7 繞 X 軸旋轉，使 Joint 與物件平行 .	xi
圖 2.8 物件相互依附	xii
圖 2.9 調整物件傳動參數	xii
圖 2.10 調整物件質量參數	xii
圖 2.11 變更物件材質 (將數字貼在物件上)	xiii
圖 2.12 選擇要使用的圖案	xiii
圖 2.13 調整映像參數	xiii
圖 2.14 背號完成	xiv
圖 3.1 LED 記分板 1	xviii

圖 3.2 LED 記分板 2	xviii
圖 3.3 sketch	xviii
圖 3.4 Advancing Wheel gear	xix
圖 3.5 M4.5 Teeth20 gear	xix
圖 3.6 M4.5 Teeth8 gear	xix
圖 3.7 Number Wheel	xx
圖 3.8 Number Wheell1	xx
圖 3.9 board	xx
圖 3.10 Unit Digit Wheel	xxi
圖 3.11 Ten's Digit Wheel	xxi
圖 3.12 Mechanical Counter	xxii
圖 3.13 一次選取所有 STL 檔導入	xxiii
圖 3.14 將 wheel 物件爆炸分解	xxiii
圖 3.15 更改數字名稱	xxiv
圖 3.16 更改數字顏色	xxiv
圖 3.17 新增 Joint	xxv
圖 3.18 調整 Joint 方向及大小	xxv
圖 3.19 物件相互依附	xxvi
圖 3.20 調整 Joint 方向及大小	xxvii

圖 3.21 調整物件傳動參數	xxvii
圖 3.22 調整物件質量參數	xxviii
圖 3.23 完成記分板	xxviii
圖 4.1 Scene	xxix
圖 5.1 關閉防火牆	xxx
圖 5.2 點選進階設定	xxx
圖 5.3 設定輸入及輸出規則類型	xxx
圖 5.4 設定特定遠端連接埠	xxxi
圖 5.5 v 允許連線	xxxi
圖 5.6 設定名稱	xxxi
圖 5.7 設定規定的 ipv6	xxxii

第一章 目標

接續專案二，對雙輪車進行設計改良，以提升行進與對戰效能，採用 CAD 進行場景與多輪車零組件設計後，轉入足球場景中以鍵盤 arrow keys 與 wasd 等按鍵進行控制，對陣雙方每組將有四名輪車球員，且每兩人在同一台電腦上操作。球賽計分系統必須採.ttm 格式建立（099），使能通用於各類場景計數之用，並可擴增至三位數計分。除了採用 LED 顯示計分外，另外以建立機械轉盤傳動計分系統。

1.1 規格

足球規格: 白色，直徑 0.1m，重量 0.5kg

足球場地: 長 4m x 寬 2.5m

球門規格: 長 0.6m，高 0.3m，寬 0.1m

球員尺寸範圍: 長寬高各 0.2m，重量 5kg

1.2 規則

遊戲規則如下:

雙方球員各別控制，進球門得一分

在規定時間內得最多分者獲勝

進球後，球會重新從中間開始

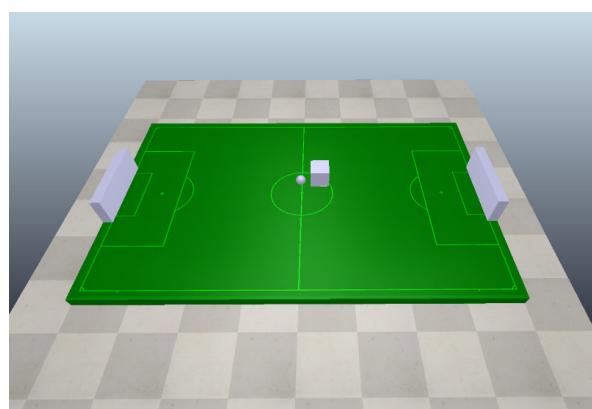


圖. 1.1: 足球場景

第二章 球員

2.1 組裝

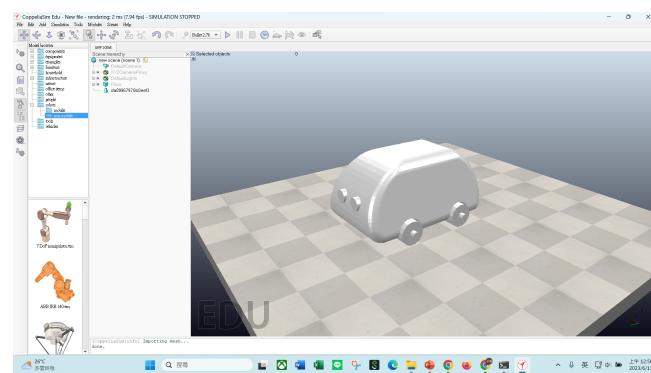


圖. 2.1: 導入球員 STL 檔

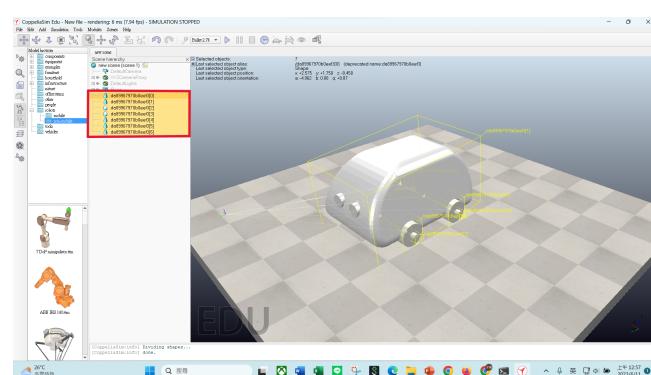


圖. 2.2: 爆炸分解

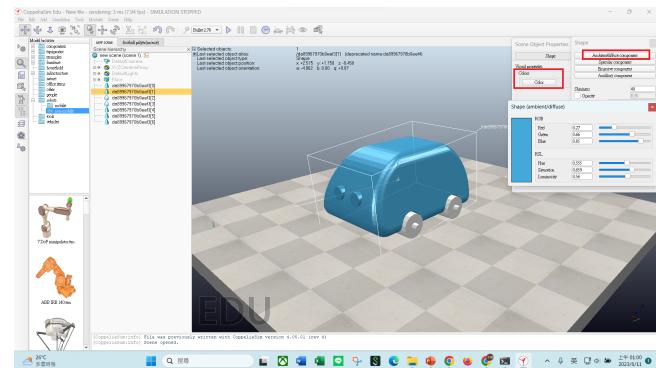


圖. 2.3: 更改顏色



圖. 2.4: 更改物件名稱

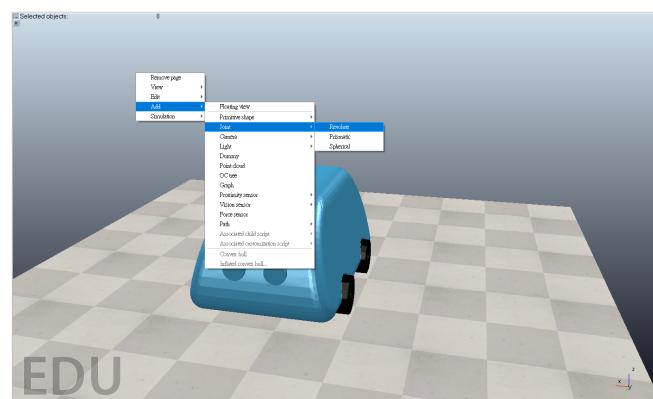


圖. 2.5: 新增 Joint

2.2 背號

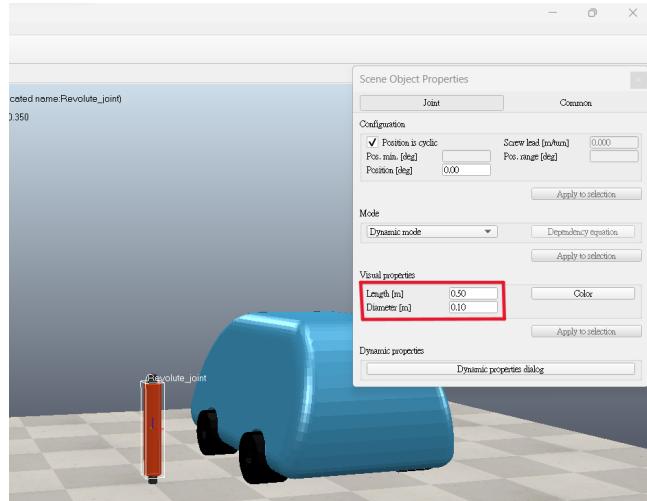


圖. 2.6: 調整 Joint 大小

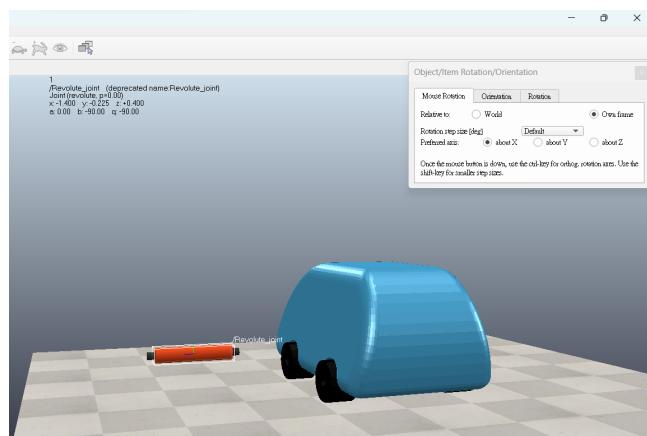


圖. 2.7: 繞 X 軸旋轉，使 Joint 與物件平行

2.3 程式碼

連線主機---使用遠端 API 來連接到位於本地主機 (localhost) 上執行，RemoteAPIClient 是用於建立與遠端應用程式之間的通訊連接。指定了要連接的主機名稱為 "localhost"，連接埠為 23000。使用 client.getObject("sim") 來獲取遠端應用程式中的一個名為 "sim" 的物件。"Program started" 表示程式開始執行，然後程式執行 client.getObject("sim") 這行程式碼後，輸出了 "Simulation started"，表示模擬 (simulation) 開始執行。

```

1 client = RemoteAPIClient('localhost', 23000)
2 print('Program started')
3 sim = client.getObject('sim')
4 #sim.startSimulation()

```

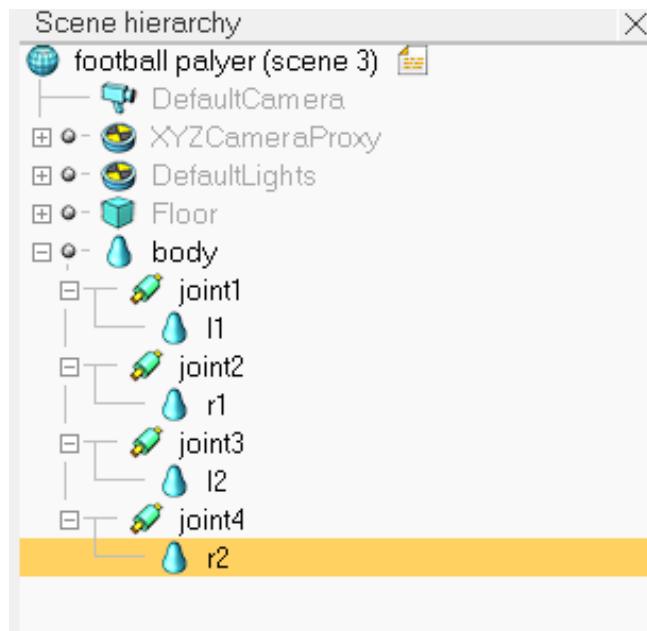


圖. 2.8: 物件相互依附

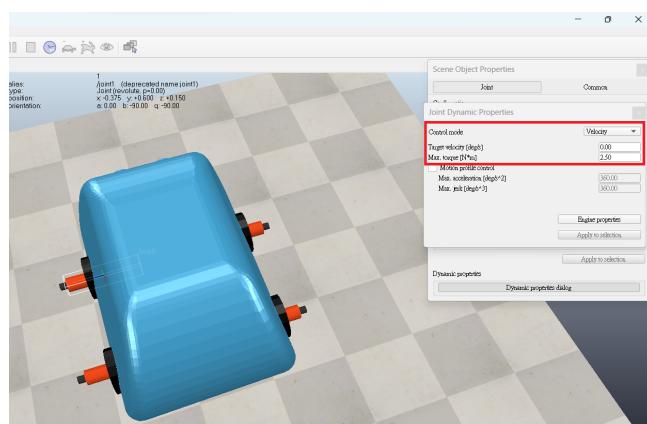


圖. 2.9: 調整物件傳動參數

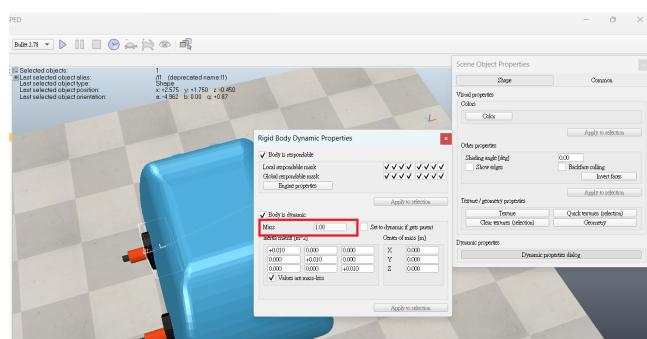


圖. 2.10: 調整物件質量參數

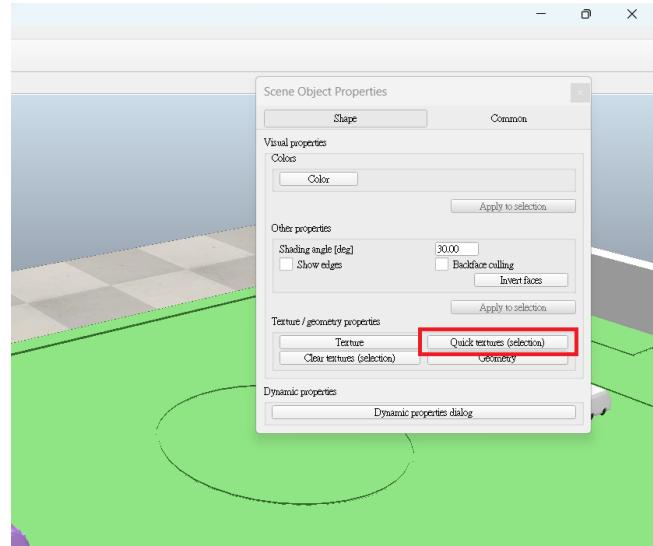


圖. 2.11: 變更物件材質 (將數字貼在物件上)

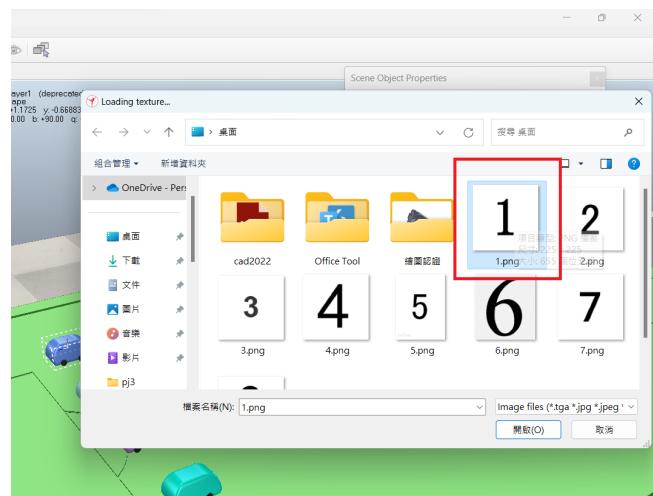


圖. 2.12: 選擇要使用的圖案

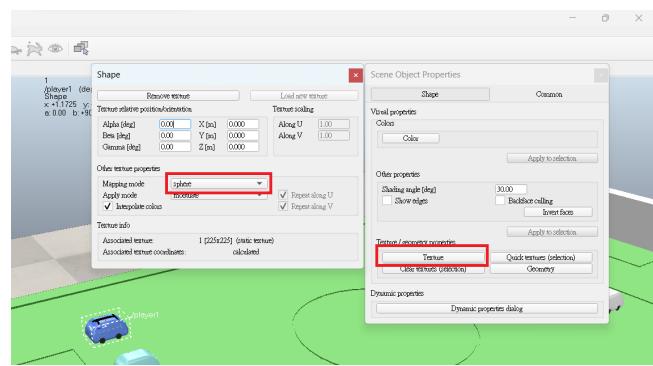


圖. 2.13: 調整映像參數



圖. 2.14: 背號完成

```
5 | print('Simulation started')
```

輪子速度---定義一個 setBubbleRobVelocity 的函式，用於設定 BubbleRob 模型的輪子速度。首先使用 sim.getObject 函式獲取四個物件 leftMotor1、rightMotor1、leftMotor2、rightMotor2。用 sim.setJointTargetVelocity 函式 >>> 設定輪子的目標速度，setJointTargetVelocity 函式 >>> 設定 joint 的目標速度，因此可以用來控制輪子的運動。函式中的四個參數 leftWheelVelocity1、rightWheelVelocity1、leftWheelVelocity2、rightWheelVelocity2 分別代表四個輪子的速度。這些參數將會被傳遞給對應的 setJointTargetVelocity 函式來設定輪子的目標速度。

```
1 | def setBubbleRobVelocity(leftWheelVelocity1, rightWheelVelocity1
2 |     ,leftWheelVelocity2, rightWheelVelocity2):
3 |     leftMotor1 = sim.getObject('/lM1')
4 |     rightMotor1 = sim.getObject('/rM1')
5 |     leftMotor2 = sim.getObject('/lM2')
6 |     rightMotor2 = sim.getObject('/rM2')
7 |     sim.setJointTargetVelocity(leftMotor1, leftWheelVelocity1)
8 |     sim.setJointTargetVelocity(rightMotor1, rightWheelVelocity1)
9 |     sim.setJointTargetVelocity(leftMotor2, leftWheelVelocity2)
10 |    sim.setJointTargetVelocity(rightMotor2, rightWheelVelocity2)
11 |    #輸入四個變數分別給四個軸速度
```

輪子方向---定義名為 setBubbleRobangel 的函式。目的是根據輸入的參數 a，改變 BubbleRob 模型的前輪方向。首先使用 sim.getObject 函式獲取物件 bR，它代表 BubbleRob 模型的身體 (body)。接著程式計算出一個 angel 列表。angel 列表包含三個

元素，第一個元素為 -90 度轉換為絆度的值，第二個元素為 a 度轉換為絆度的值，第三個元素為 0。然後用 sim.getObject 函式分別獲取物件 leftMotor 和 rightMotor，它們分別代表 BubbleRob 模型的左輪和右輪。最後使用 sim.setObjectOrientation 函式來設定物件的方向。該函式用於設定物件相對於參考物件的方向。在這裡將 leftMotor 和 rightMotor 的方向設定為相對於 bR 物件的方向，方向由 angel 列表指定。

```
1 def setBubbleRobangel(a):  
2     bR = sim.getObject('/bR')  
3     angel = [-90*math.pi/180, a*math.pi/180, 0]  
4     leftMotor = sim.getObject('/lM1')  
5     rightMotor = sim.getObject('/rM1')  
6     sim.setObjectOrientation(leftMotor, bR, angel)  
7     sim.setObjectOrientation(rightMotor, bR, angel)  
8     #輸入一個變數改變前輪方向
```

控制---這段程式碼是一個無窮迴圈，它會持續監聽鍵盤的按鍵輸入，並根據接下的按鍵來控制 BubbleRob 模型的運動。目的是根據鍵盤輸入來控制 BubbleRob 模型的運動，並提供了前進、後退、旋轉和停止等功能。

```
1 while True:
2
3     if keyboard.is_pressed('w'):
4
5         setBubbleRobVelocity(4, 4, 4, 4)
6
7         if keyboard.is_pressed('a'):
8
9             setBubbleRobangel(-40)
10
11        elif keyboard.is_pressed('d'):
12
13            setBubbleRobangel(40)
14
15        else:
16
17            setBubbleRobangel(0)
18
19        elif keyboard.is_pressed('s'):
20
21            setBubbleRobVelocity(-4, -4, -4, -4)
22
23            if keyboard.is_pressed('a'):
24
25                setBubbleRobangel(-40)
26
27            elif keyboard.is_pressed('d'):
28
29                setBubbleRobangel(40)
30
31            else:
32
33                setBubbleRobangel(0)
34
35        elif keyboard.is_pressed('a'):
36
37            setBubbleRobVelocity(-4, 4, -4, 4)
38
39        elif keyboard.is_pressed('d'):
40
41            setBubbleRobVelocity(4, -4, 4, -4)
42
43        elif keyboard.is_pressed('q'):
44
45            % stop simulation
46
47            sim.stopSimulation()
48
49        else:
50
51            setBubbleRobVelocity(0, 0, 0, 0)
52
53            setBubbleRobangel(0)
54
55 #程式碼中使用 \text{keyboard.is_pressed} 函式來判斷指定的按鍵是否被按下。
56
57 #若按下 'w' 鍵，則呼叫 'setBubbleRobVelocity(4, 4, 4, 4)' 來設定 BubbleRob 模型的速度為正向，並根據 'a' 和 'd' 鍵的狀態設定前
```

輪的方向。

- 30 #若按下 's' 鍵，則呼叫 'setBubbleRobVelocity(-4, -4, -4, -4)' 來設定 BubbleRob 模型的速度為負向，並根據 'a' 和 'd' 鍵的狀態設定前輪的方向。
- 31 #若按下 'a' 鍵，則呼叫 'setBubbleRobVelocity(-4, 4, -4, 4)' 來設定 BubbleRob 模型的速度使其向左旋轉。
- 32 #若按下 'd' 鍵，則呼叫 \textup{'setBubbleRobVelocity(4, -4, 4, -4)'} 來設定 BubbleRob 模型的速度使其向右旋轉。
- 33 #若按下 'q' 鍵，則停止模擬 (*simulation*)。
- 34 #若沒有按下任何按鍵，則呼叫 'setBubbleRobVelocity(0, 0, 0, 0)' 和 'setBubbleRobangel(0)'，將 BubbleRob 模型的速度和前輪方向設定為零，即停止移動。

第三章 記分板

3.1 繪圖

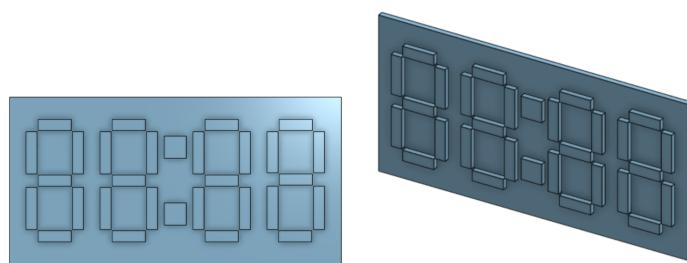


圖. 3.1: LED 記分板 1

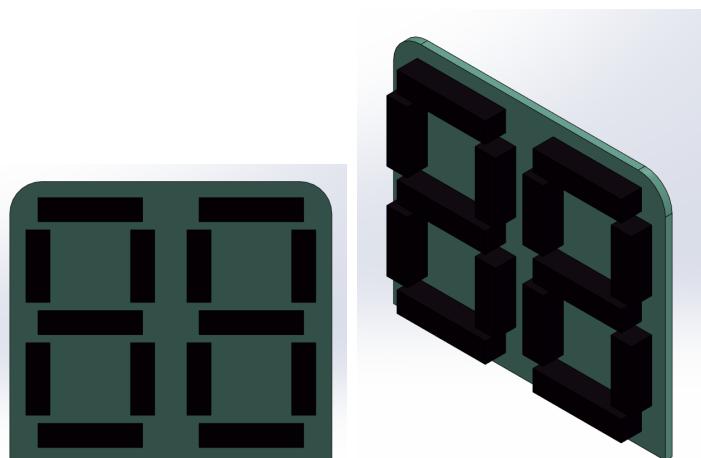


圖. 3.2: LED 記分板 2

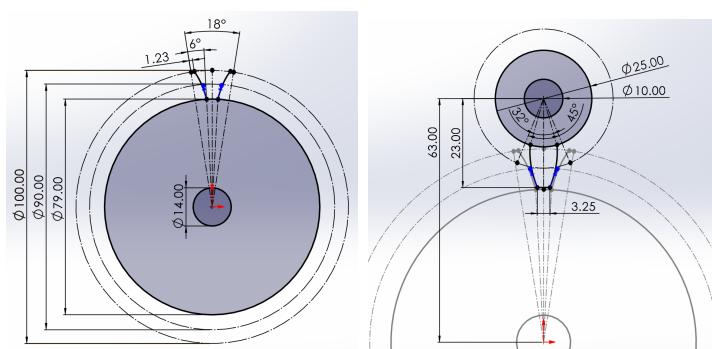


圖. 3.3: sketch

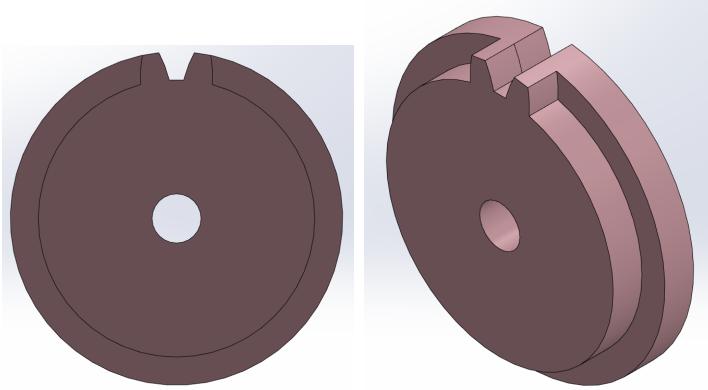


圖. 3.4: Advancing Wheel gear

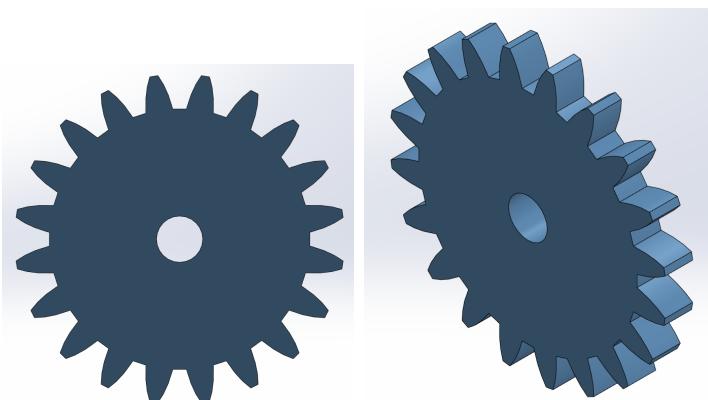


圖. 3.5: M4.5 Teeth20 gear

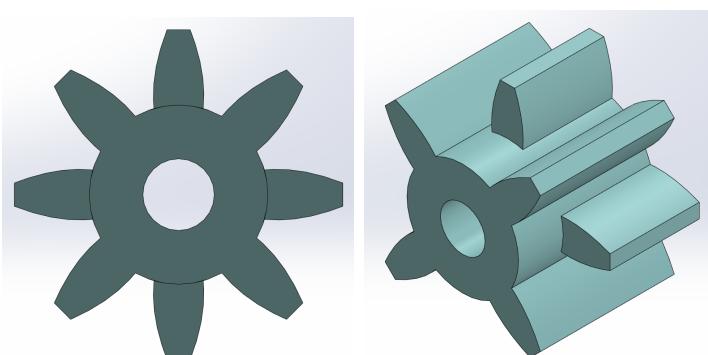


圖. 3.6: M4.5 Teeth8 gear

3.2 組合

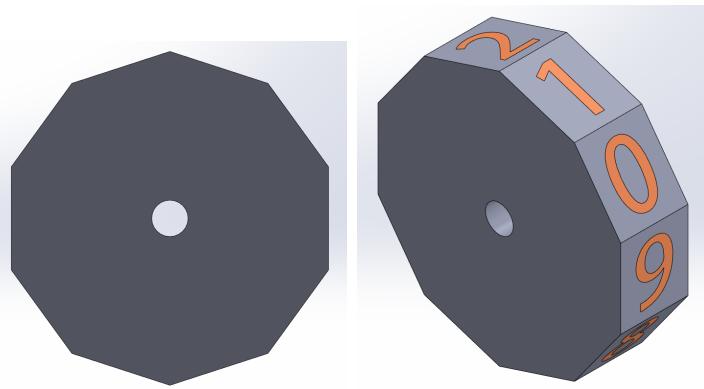


圖. 3.7: Number Wheel

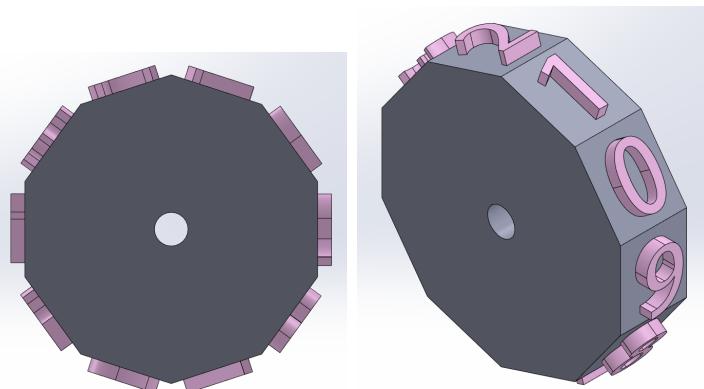


圖. 3.8: Number Wheel1

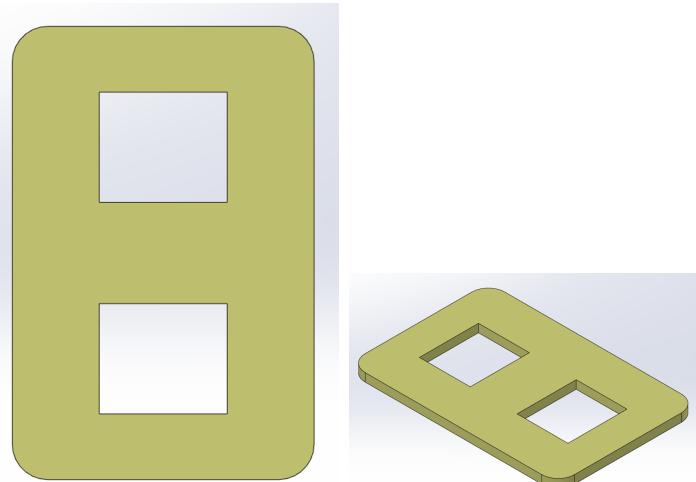


圖. 3.9: board

3.3 組裝

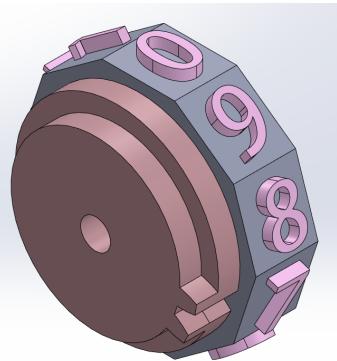
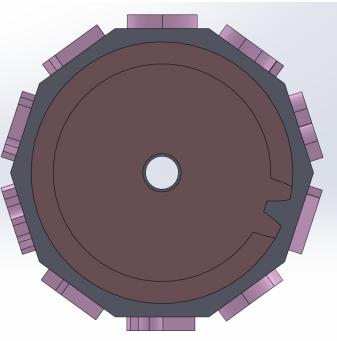


圖. 3.10: Unit Digit Wheel

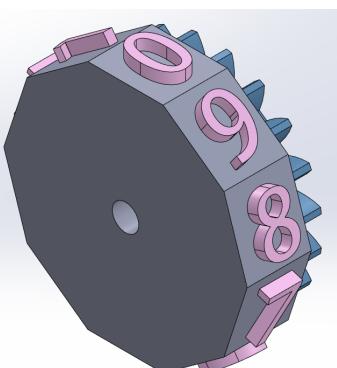
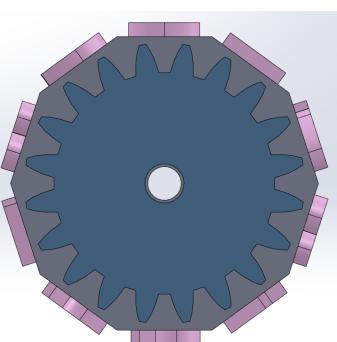


圖. 3.11: Ten's Digit Wheel

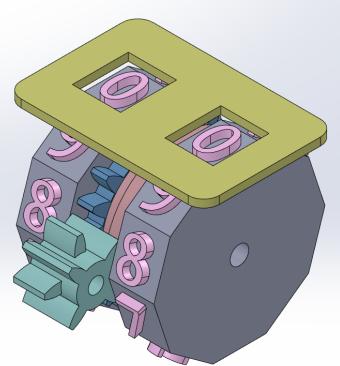
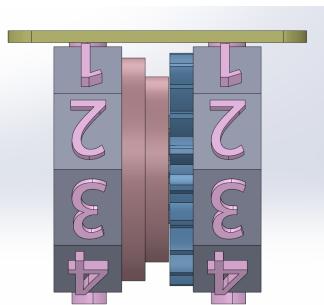
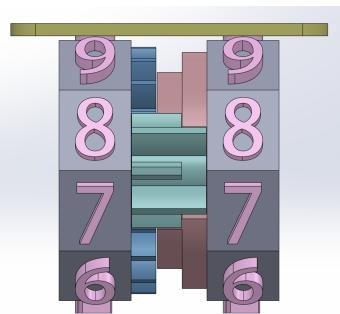
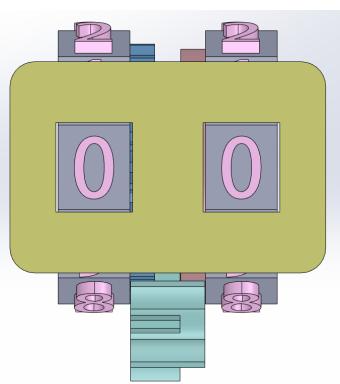


圖. 3.12: Mechanical Counter

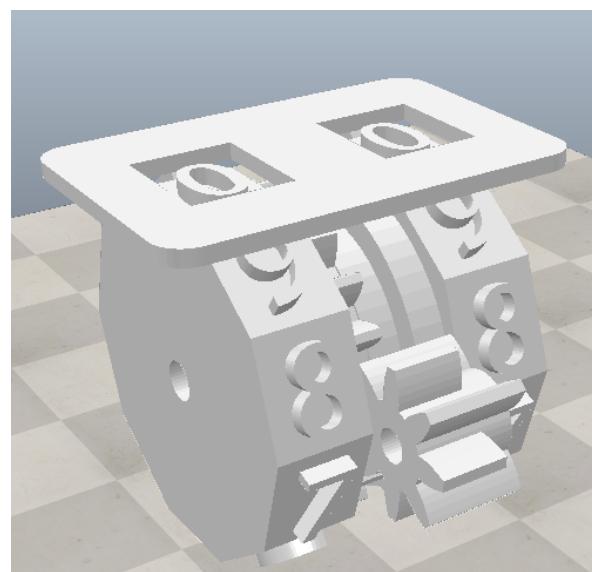
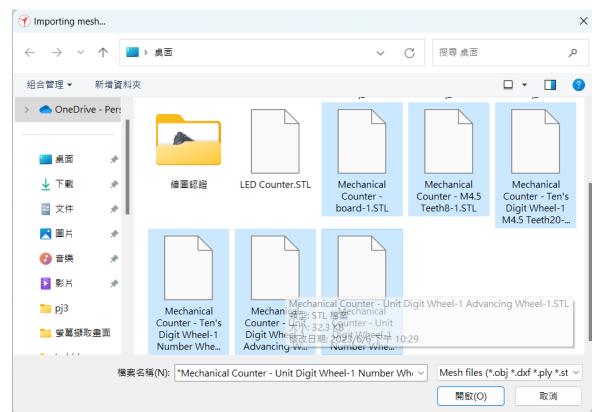


圖. 3.13: 一次選取所有 STL 檔導入

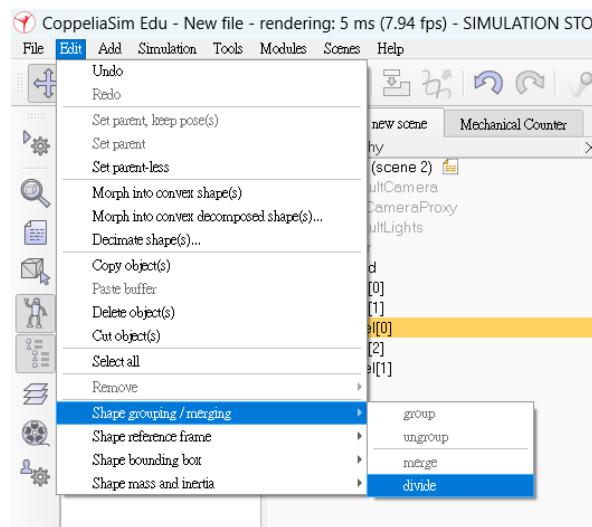


圖. 3.14: 將 wheel 物件爆炸分解

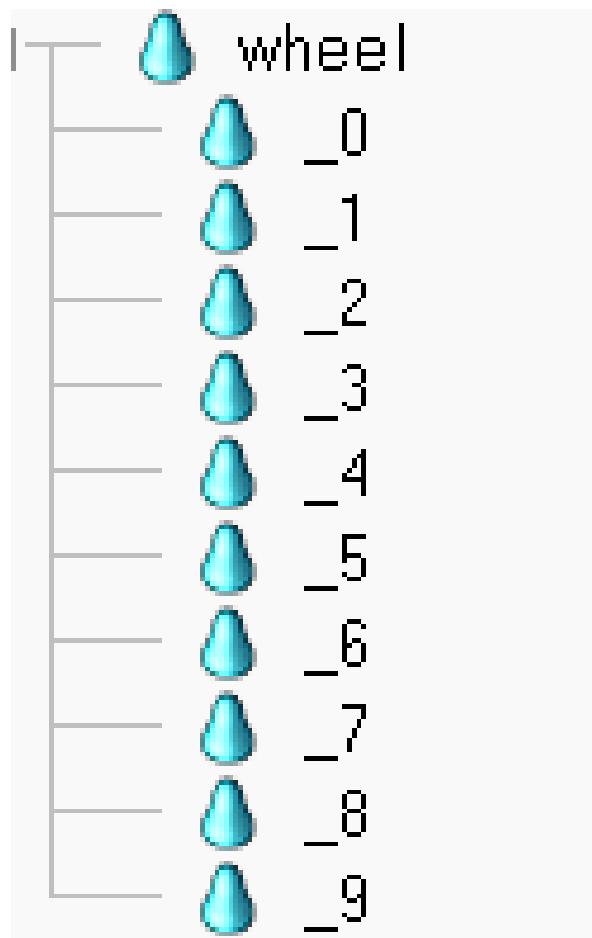


圖. 3.15: 更改數字名稱

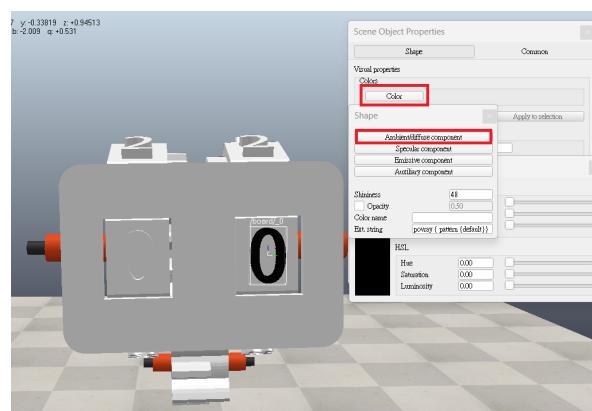


圖. 3.16: 更改數字顏色

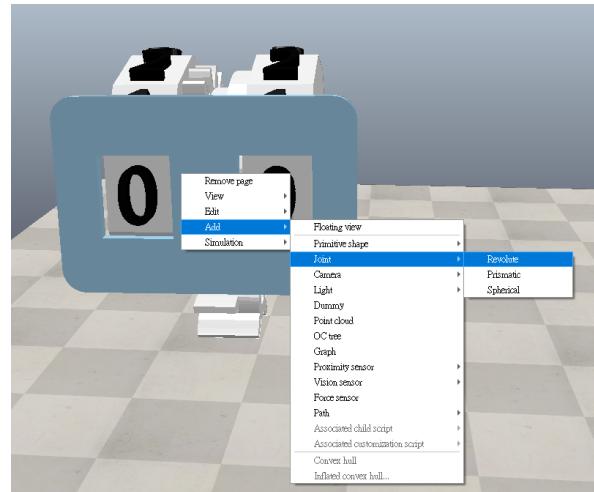


圖. 3.17: 新增 Joint

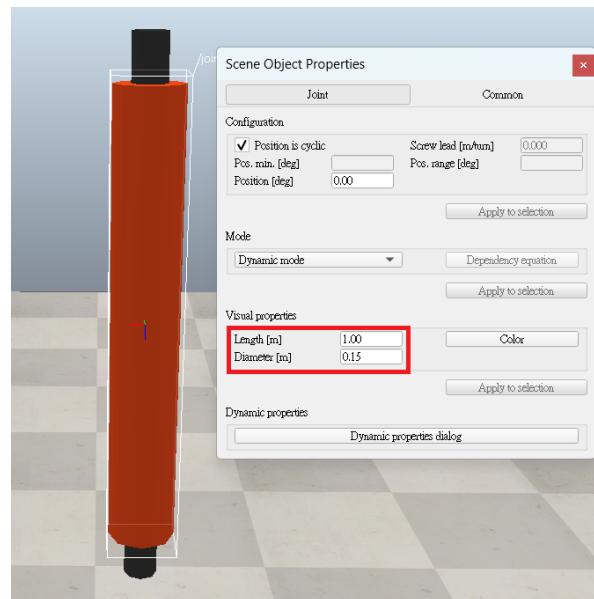


圖. 3.18: 調整 Joint 方向及大小

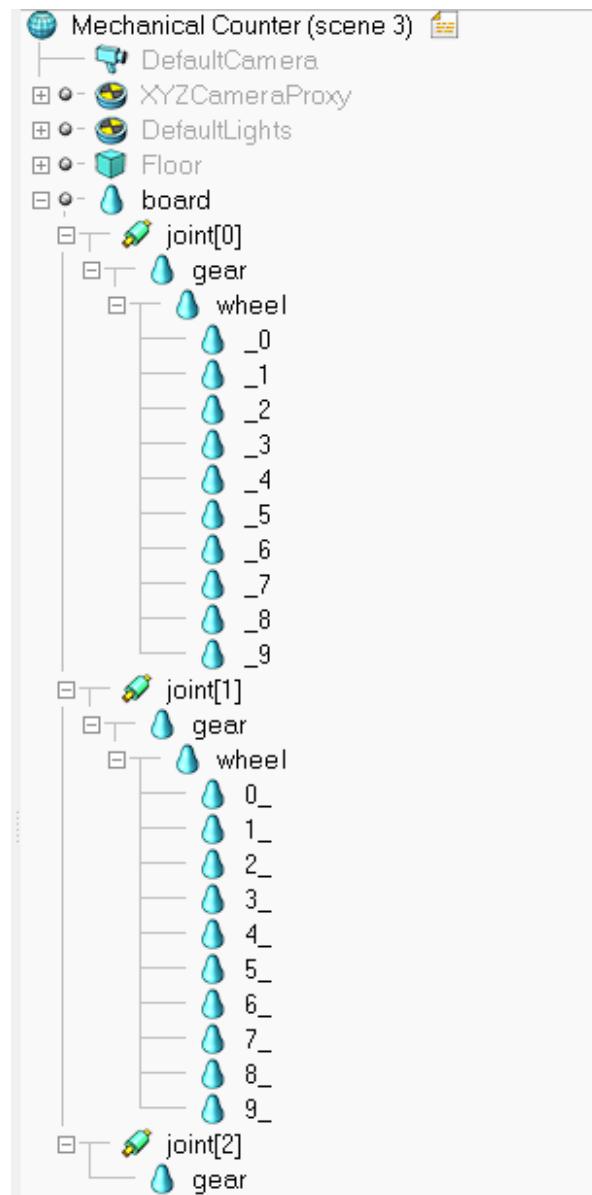


圖. 3.19: 物件相互依附

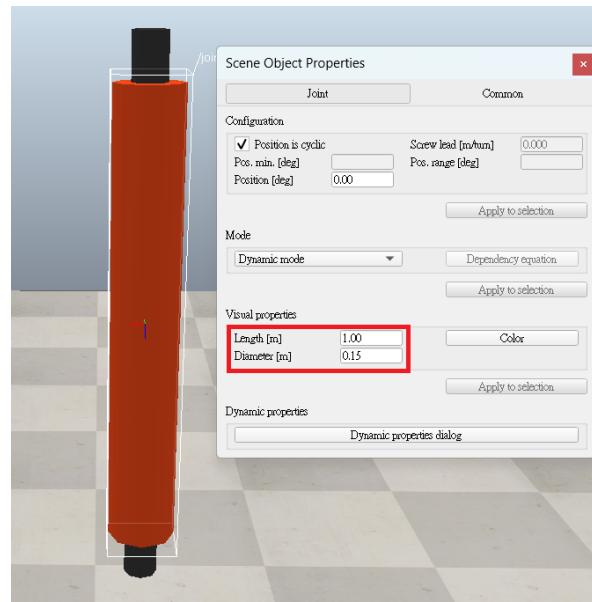


圖. 3.20: 調整 Joint 方向及大小

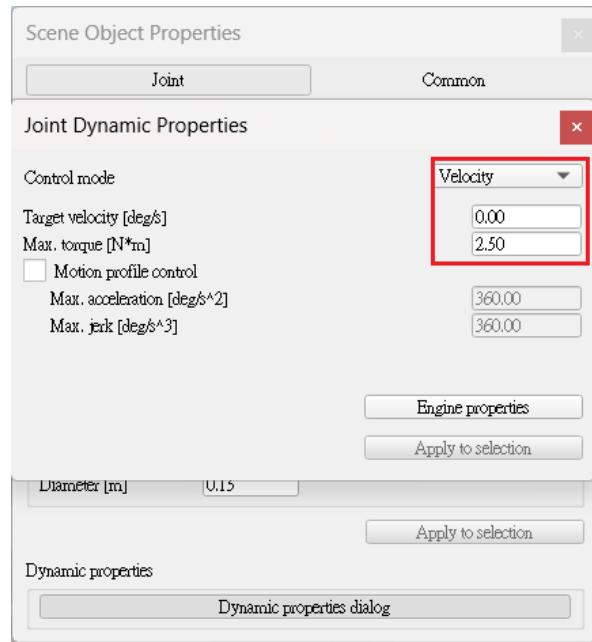


圖. 3.21: 調整物件傳動參數

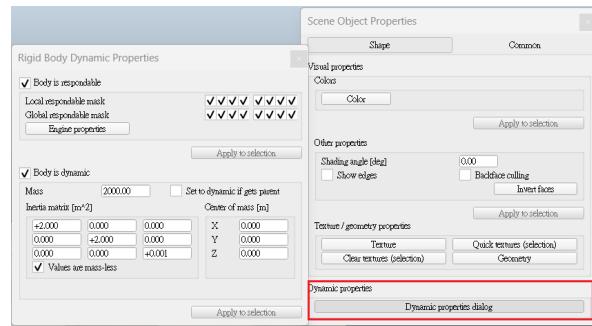


圖. 3.22: 調整物件質量參數

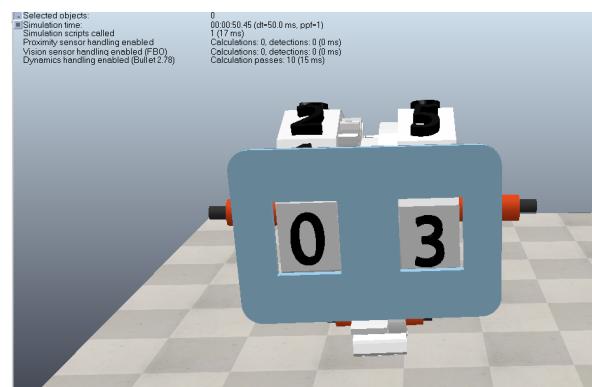


圖. 3.23: 完成記分板

第四章 場景

4.1 匯入球場球門球員記分板

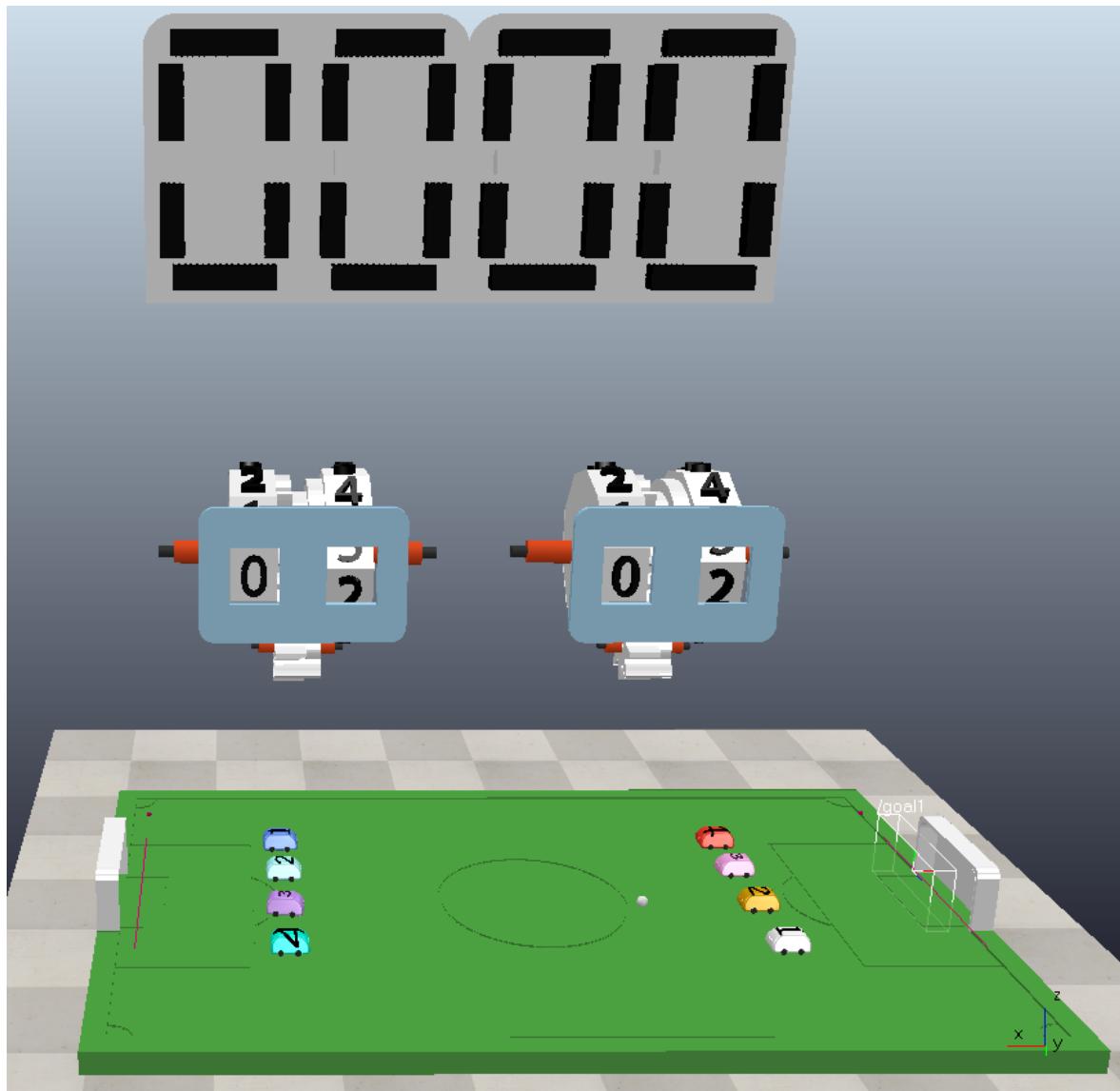


圖. 4.1: Scene

第五章 跨網路連線

5.1 設定連線步驟



圖. 5.1: 關閉防火牆

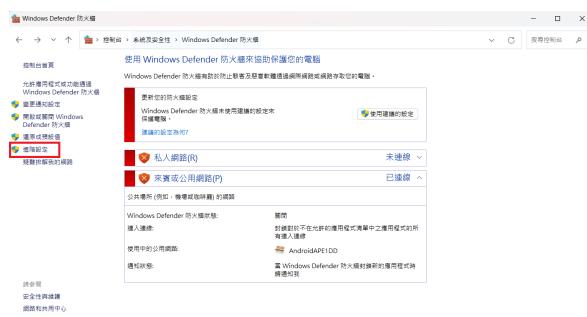


圖. 5.2: 點選進階設定

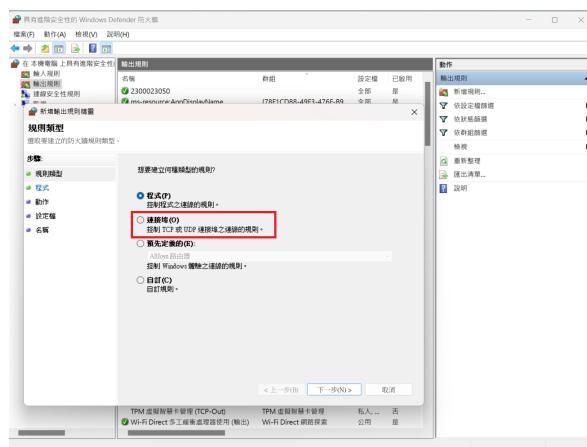


圖. 5.3: 設定輸入及輸出規則類型

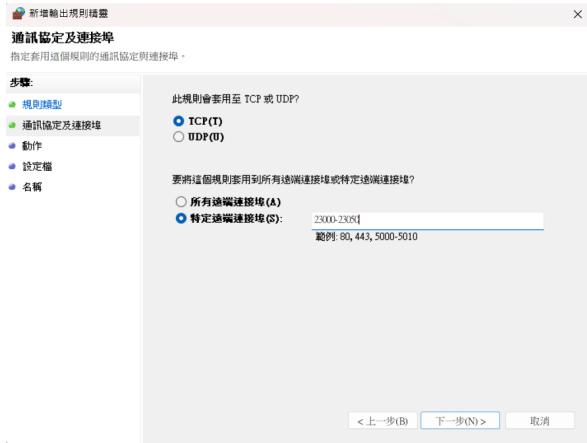


圖. 5.4: 設定特定遠端連接埠

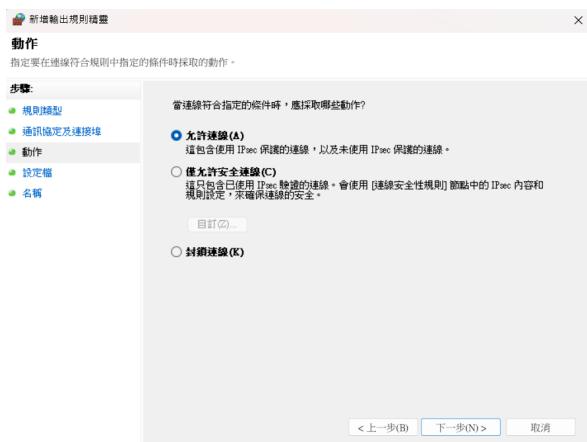


圖. 5.5: v 允許連線

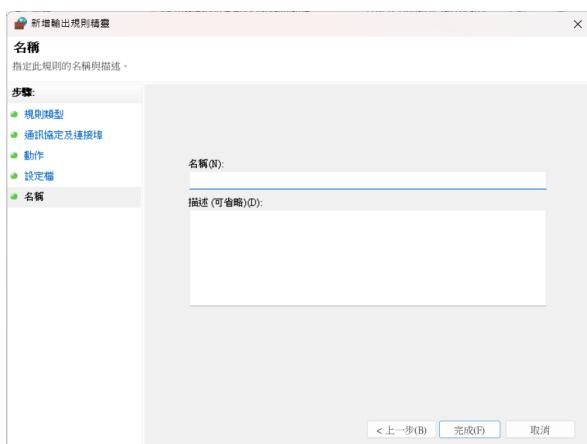


圖. 5.6: 設定名稱

5.2 程式碼

```
1 | # pip install pyzmq cbor keyboard
```

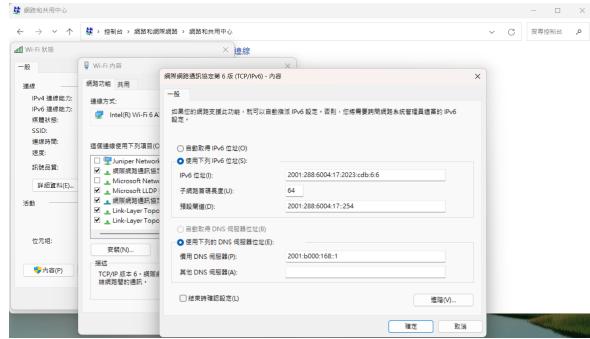


圖. 5.7: 設定規定的 ipv6

```

2 from zmqRemoteApi import RemoteAPIClient
3
4
5 client = RemoteAPIClient('localhost', 23000)
6
7
8 print('Program started')
9
10 sim = client.getObject('sim')
11
12
13 def setBubbleRobVelocity(leftWheelVelocity, rightWheelVelocity):
14     leftMotor = sim.getObject('/leftMotor')
15     rightMotor = sim.getObject('/rightMotor')
16     sim.setJointTargetVelocity(leftMotor, leftWheelVelocity)
17     sim.setJointTargetVelocity(rightMotor, rightWheelVelocity)
18
19 '''
20 # Example usage 1:
21 setBubbleRobVelocity(1.0, 1.0)
22 time.sleep(2)
23 setBubbleRobVelocity(0.0, 0.0)
24 '''
25 # use keyboard to move BubbleRob
26

```

```
27 while True:  
28     if keyboard.is_pressed('w'):  
29         setBubbleRobVelocity(1.0, 1.0)  
30     elif keyboard.is_pressed('s'):  
31         setBubbleRobVelocity(-1.0, -1.0)  
32     elif keyboard.is_pressed('a'):  
33         setBubbleRobVelocity(-1.0, 1.0)  
34     elif keyboard.is_pressed('d'):  
35         setBubbleRobVelocity(1.0, -1.0)  
36     elif keyboard.is_pressed('q'):  
37         # stop simulation  
38         sim.stopSimulation()  
39     else:  
40         setBubbleRobVelocity(0.0, 0.0)
```