

國立虎尾科技大學  
機械設計工程系  
**cd2023 2a-pj1ag2 分組報告**

網際足球機器人

**Web-based Football Scene Design**

指導教授： 嚴 家 銘 老 師  
班 級： 四 設 二 甲  
學 生： 第 一 位 (41023146)  
第 二 位 (41023148)

中華民國 112 年 3 月

國立虎尾科技大學 機械設計工程系  
分組報告製作合格認可證明

分組報告製作修習學生： 四設二甲 41023146 第一位  
四設二甲 41023148 第二位

分組報告題目：網際手足球場景設計

經評量合格，特此證明

評 審 委 員： \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

指 導 老 師： \_\_\_\_\_

中 華 民 國 一 一 二 年 三 月 三 十 一 日

## 摘 要

本課程將在設計簡單的移動機器人 BubbleRob 的同時，介紹相當多的 CoppeliaSim 功能。本教程相關的 CoppeliaSim 場景文件位於 `scenes/tutorials/BubbleRob`。

此專題是運用足球機器人，將其導入 CoppeliaSim 模擬環境並給予對應設置，將其機電系統簡化並運用 AI 進行訓練，找到適合此系統的演算法後，再到 CoppeliaSim 模擬環境中進行測試演算法在實際運用上的可行性。並嘗試透過架設伺服器將 CoppeliaSim 影像串流到網頁供使用者觀看或操控。

關鍵字: 類神經網路、強化學習、caht gpt、CoppeliaSim、OpenAI Gym

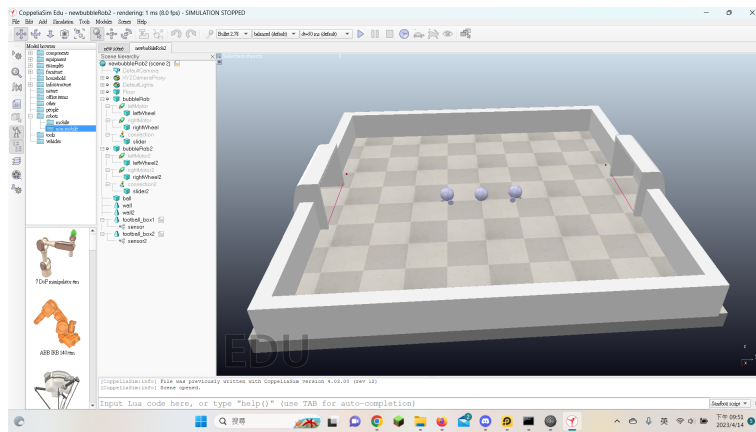
## 誌謝

在此鄭重感謝製作以及協助本分組報告完成的所有人員，首先感謝學長範本，讓我們比較好改。

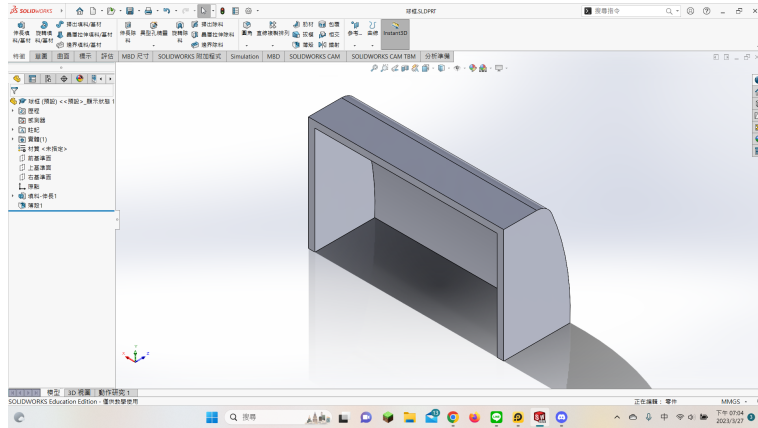
# 第一章 前言

## 1.1 研究動機

機器學習與各領域結合的應用越來越廣泛，在機電系統採用強化學習是為了讓機電系統的控制達到最佳化。本專題以實體的足球機之機電系統作為訓練模型，將實體機器轉移到虛擬環境進行模擬，為了找到適合的訓練參數，因此將模型簡化後再進行測試各種參數的優劣，透過不斷的訓練來得到一個優化過的對打系統，以下是成品圖。



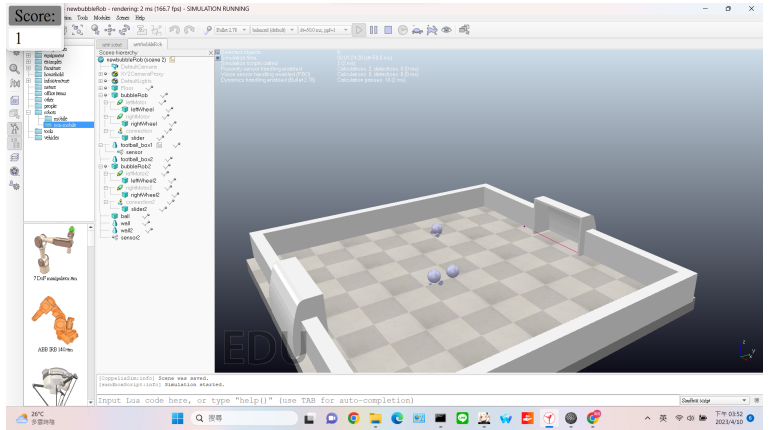
## 1.2 製作過程



首先繪製球框。

```
1 function sysCall_init()
2     right_wheel= sim.getObjectHandle('joint1')
3     left_wheel= sim.getObjectHandle('joint2')
4     right_velocity=0
5     left_velocity=0
6     speed=5
7     sim.setJointTargetVelocity(right_wheel,0)
8     sim.setJointTargetVelocity(left_wheel,0)
9
10 end
11
12 function sysCall_actuation()
13     message,auxiliaryData=sim.getSimulatorMessage()
14     while message~-1 do
15         if (message==sim.message_keypress) then
16
17             if (auxiliaryData[1]==32) then
18
19                 right_velocity=0
20                 left_velocity=0
21                 sim.setJointMaxForce(right_wheel, 0)
22                 sim.setJointMaxForce(left_wheel, 0)
23                 break
24             else
25                 sim.setJointMaxForce(right_wheel, 10)
26                 sim.setJointMaxForce(left_wheel, 10)
27
28                 if (auxiliaryData[1]==2007) then -- up key
29                     sim.setJointTargetVelocity(right_wheel,speed)
30                     sim.setJointTargetVelocity(left_wheel,speed)
31                 end
32                 if (auxiliaryData[1]==2008) then -- down key
33                     sim.setJointTargetVelocity(right_wheel,-speed/2)
34                     sim.setJointTargetVelocity(left_wheel,-speed/2)
35                 end
36                 if (auxiliaryData[1]==2009) then -- left key
37                     sim.setJointTargetVelocity(right_wheel,speed)
38                     sim.setJointTargetVelocity(left_wheel,speed/2)
39                 end
40                 if (auxiliaryData[1]==2010) then -- right key
41                     sim.setJointTargetVelocity(right_wheel,speed/2)
42                     sim.setJointTargetVelocity(left_wheel,speed)
43                 end
44             end
45         end
46         message,auxiliaryData=sim.getSimulatorMessage()
47     end
48 end
```

執行程式碼。



加入球框感測器和記分板。

感測器 lua 腳本，詳情可見

<https://mdec2023.github.io/football-apj1/content/ag2.html>

```

1  function sysCall_init()
2      score1 = 0
3
4      sensor = sim.getObject('/./sensor')
5      xml = [[
6          <ui title="Scoreboard" closeable="false" resizable="false" style="plastique">
7              <label text="Score:" style="* {background-color: #808080; color: #000000; font-size: 40px; font-weight: bold;
8                  <label text="0" style="* {background-color: #FFF; color: #000000; font-size: 40px; font-weight: bold; padding:
9
10             </ui>
11         ]]
12         ui = simUI.create(xml)
13         simUI.setPosition(ui, 0,0, true)
14         bubbleRob = sim.getObject('/bubbleRob')
15         ball = sim.getObject('/ball')
16         bubbleRob2 = sim.getObject('/bubbleRob2')
17         initialPosition = sim.getObjectPosition(bubbleRob, -1)
18         initialOrientation = sim.getObjectOrientation(bubbleRob, -1)
19         initialPosition2 = sim.getObjectPosition(bubbleRob2, -1)
20         initialOrientation2 = sim.getObjectOrientation(bubbleRob2, -1)
21         initialballPosition = sim.getObjectPosition(ball, -1)
22         initialballOrientation = sim.getObjectOrientation(ball, -1)
23     end
24
25
26
27 function sysCall_actuation()
28     --simUI.setLabelText(ui, 30, tostring(sim.getFloatSignal("myVariable")))
29     result=sim.readProximitySensor(sensor)
30     if(score1<5)then
31         if(result>0)then
32             score2 = score1+1
33             simUI.setLabelText(ui, 30, tostring(score2))
34
35             sim.setObjectPosition(bubbleRob, -1, initialPosition)
36             sim.setObjectOrientation(bubbleRob, -1, initialOrientation)
37             sim.setObjectPosition(bubbleRob2, -1, initialPosition2)
38             sim.setObjectOrientation(bubbleRob2, -1, initialOrientation2)
39             sim.setObjectPosition(ball, -1, initialballPosition)
40             sim.setObjectOrientation(ball, -1, initialballOrientation)
41             score1=score2
42         end
43     else
44         sim.pauseSimulation()
45     end
46 end

```

# 參考文獻

[1] <https://www.coppeliarobotics.com/helpFiles/index.html>

## 附錄

### LaTeX

LaTeX 為一種程式語言，支援標準庫 (Standard Libraries) 和外部程式庫 (External Libraries)，不過與一般程式語言不同的是，它可以直接表述 Tex 排版結構，類似於 PHP 之於 HTML 的概念。但是直接撰寫 LaTeX 仍較複雜，因此可以藉由 Markdown 這種輕量的標註式語言先行完成文章，再交由 LaTeX 排版。此專題報告採用編輯軟體為 LaTeX，綜合對比 Word 編輯方法，LaTeX 較為精準正確、更改、製作公式等，以便符合規範、製作。

表. 1: 文字編輯軟體比較表

	相容性	直觀性	文件排版	數學公式	微調細部
LaTeX	✓		✓	✓	✓
Word		✓			✓

- 特點:

1. 相容性：以 Word 為例會有版本差異，使用較高版本編輯的文件可能無法以較低的版本開啟，且不同作業系統也有些許差異；相比 LaTeX 可以利用不同編譯器進行編譯，且為免費軟體也可移植至可攜系統內，可以搭配 Github 協同編譯。
2. 文件排版：許多規範都會要求使用特定版型，使用文字編譯環境較能準確符合規定之版型，且能夠大範圍的自定義排定所需格式，並能不受之後更改而整體格式變形。
3. 數學公式呈現：LaTeX 可以直接利用本身多元的模組套件加入、編輯數學公式，在數學推導過程能夠快速的輸入自己需要的內容即可。



4. 細部調整：在大型論文、報告中有多項文字、圖片、表格，需要調整細部時，要在好幾頁中找尋，而 LaTeX 可以分段章節進行編譯，再進行合併處理大章節。

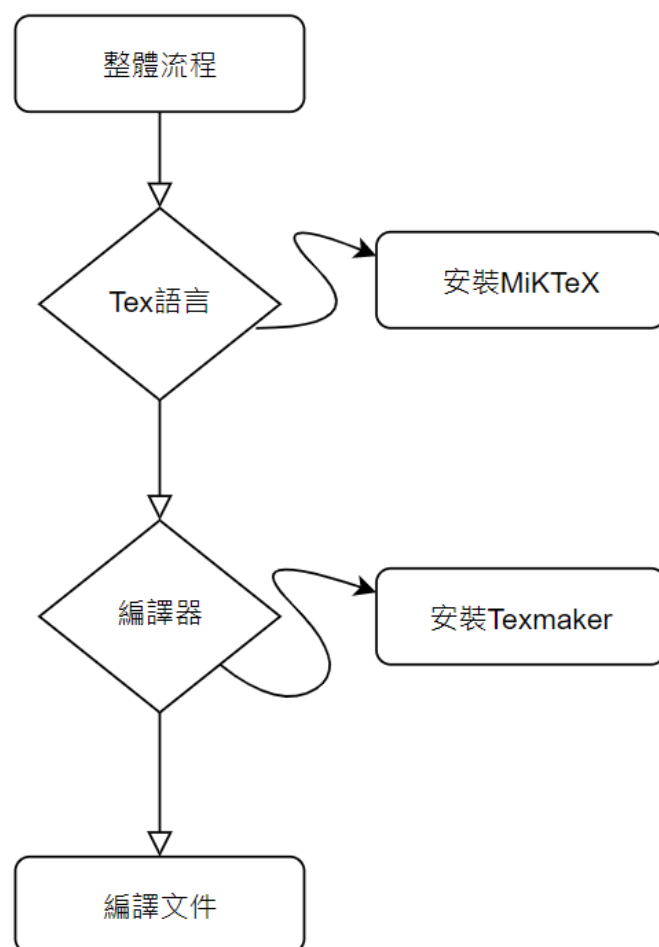


圖. 1: 編譯流程

## FFmpeg

FFmpeg 是一個開放原始碼的自由軟體，可以對音訊和視訊進行多種格式的錄影、轉檔、串流功能。在專題訓練過程中透過 FFmpeg 的視訊錄製的功能記錄對打影像來了解實際訓練狀況。