國 立 虎 尾 科 技 大 學
機 械 設 計 工 程 系

cd2023 2a-pj1ag14 分組報告

網際足球泡泡機器人場景設計

Web-based bubbleRob Football Scene Design

指導教授： 嚴 家 銘 老 師
班 級： 四 設 二 甲
學 生： 李 學 淵 (41023125)
林 呈 恩 (41023128)

中華民國 112 年 4 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。

此專案是運用 w3 作業所做的泡泡機器人，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導入 Lua 及 python 程式，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。

Abstract

Due to the four major development trends of multidimensional arrays computing, automatic differentiation, open source development environment, and multi-core GPUs computing hardware. The rapid development of the AI field has been promoted. In view of this development, the physical mechatronic systems can gain machine learning efficiency through their simulated virtual system training process. And afterwards to apply the trained model into real mechatronic systems.

This project involves taking the bubbleRob created for the W3 assignment and importing it into the CoppeliaSim simulation environment with corresponding settings, simplifying its electromechanical system, and incorporating Lua and Python programming for testing algorithms in the CoppeliaSim simulation environment to determine their feasibility in practical application. Additionally, the project aims to stream CoppeliaSim images to a webpage for users to view or manipulate by setting up a server.

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，首先向嚴家銘老師致謝，不厭其煩的回答我們的提問，總是像燈塔一樣為我們指引出最正確的方向。特此感謝。

目　錄

圖　目　錄

表　目　錄

第一章　前言

1.1 規則說明

Pong game 的遊戲規則簡單，透過操控 bubbleRob 將球打入對方球門即得一分，時間內其中一方分數高獲勝。

遊戲規則如下：

1. 利用 wasd 操控移動方向
2. 將球打入敵方球框即得一分。
3. 時間內進球數多的一方獲勝。
4. 時間到即停止遊戲

第二章　製作歷程

2.1 移動方式

在原有的設計上, 將自動移動改成了使用鍵盤操控

w 向上,a 向左,s 向下,d 向右

觸發不同條件來改變兩 joint 的速度達到變動方向的效果

```
function sysCall_actuation()
    local message, auxiliaryData, auxiliaryData2 = sim.getSimulatorMessage()
    result=sim.readProximitySensor(noseSensor) -- Read the proximity sensor
    -- If we detected something, we set the backward mode:
    if (message == sim.message_keypress) then
        if auxiliaryData[1] == string.byte("w") then ----w
            print("w")
            sim.setJointTargetVelocity(leftMotor,speed)
            sim.setJointTargetVelocity(rightMotor,speed)
        elseif auxiliaryData[1] == string.byte("s") then
            print("s")
            sim.setJointTargetVelocity(leftMotor,-speed)
            sim.setJointTargetVelocity(rightMotor,-speed)
        elseif auxiliaryData[1] == string.byte("a") then
            print("a")
            sim.setJointTargetVelocity(leftMotor,speed/4)
            sim.setJointTargetVelocity(rightMotor,speed)
        elseif auxiliaryData[1] == string.byte("d") then
            print("d")
            sim.setJointTargetVelocity(leftMotor,speed)
            sim.setJointTargetVelocity(rightMotor,speed/4)
        else
            sim.setJointTargetVelocity(leftMotor,0)
            sim.setJointTargetVelocity(rightMotor,0)
        end
    end
```

圖. 2.1: lua move

2.2 重製位置

添加了只要觸發特定條件就能使 bubbleRob 雙輪車回到原位

先記錄初始位置

```
initialBubbleRobPosition = sim.getObjectPosition(bubbleRobBase, -1)
initialBubbleRobOrientation = sim.getObjectOrientation(bubbleRobBase, -1)
initialNoseSensorPosition = sim.getObjectPosition(noseSensor, -1)
initialNoseSensorOrientation = sim.getObjectOrientation(noseSensor, -1)
```

圖. 2.2: 紀錄初始位置

在觸發特定條件時使用來達成重製效果

```
sim.setObjectPosition(bubbleRobBase, -1, initialBubbleRobPosition)
sim.setObjectOrientation(bubbleRobBase, -1, initialBubbleRobOrientation)
sim.setObjectPosition(noseSensor, -1, initialNoseSensorPosition)
sim.setObjectOrientation(noseSensor, -1, initialNoseSensorOrientation)
```

圖. 2.3: 重製位置

2.3 記分板與計時

增加倒數計時與分數的面板, 遊戲開始後開始倒數計時, 時間到則結束遊戲

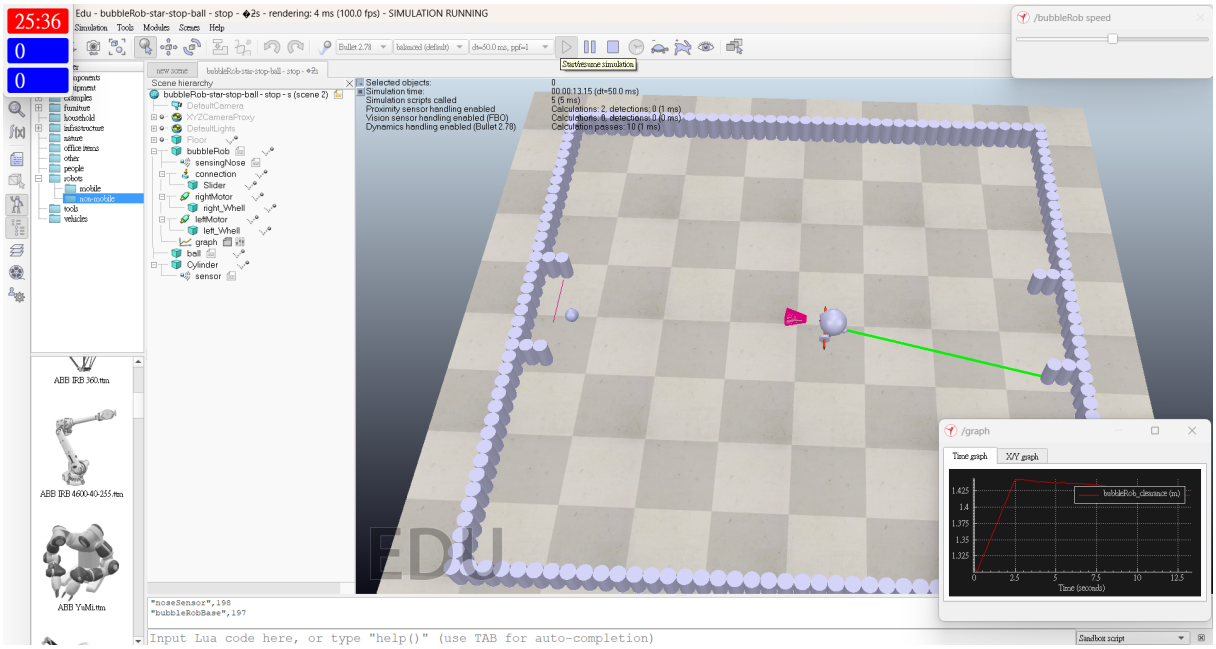

圖. 2.4: 記分板與計時

使用了 xml 進行編寫, 再利用 simUI.create 顯示出來

設定在不同的條件下，更改顯示的數字

倒數結束時則利用 sim.stopSimulation() 來停止模擬

```lua
function sysCall_init()
    count = 1800  -- ?230???????????
    score1 = 0  -- ??????
    score2 = 0
    xml = [[
        <ui closeable="false" resizeable="false" activate="false">
            <label text="30:00" style="* {background-color: #F00; color: #FFF; font-size: 32px; font-weight: bold; padding: 4px; border-radius: 4px;}
            <label text="0" style="* {background-color: #00F; color: #FFF; font-size: 32px; font-weight: bold; padding: 4px; border-radius: 4px;}" id
            <label text="0" style="* {background-color: #00F; color: #FFF; font-size: 32px; font-weight: bold; padding: 4px; border-radius: 4px;}" id
        </ui>
    ]]
    ui = simUI.create(xml)
    simUI.setPosition(ui, 0,0, true)
end

function sysCall_actuation()
    if count > 0 then
        count = count - 1
        local minutes = math.floor(count / 60)
        local seconds = count % 60
        local timeStr = string.format("%d:%02d", minutes, seconds)
        simUI.setLabelText(ui, 10, timeStr)
    else
        sim.stopSimulation()
    end
    --if count == 1700 then
        --sim.callScriptFunction(206, sim.scripttype_childscript, "updateScore", {1,2})
    --end
end
```

圖. 2.5: 記分板與計時程式

2.4 歡迎與恭喜

加入了歡迎與得分的 ui 提升觀賞性, 一樣使用 xml 編寫即可使用同一個 id, 只是更改顯示的內容, 開始時隱藏, 暫停時顯現

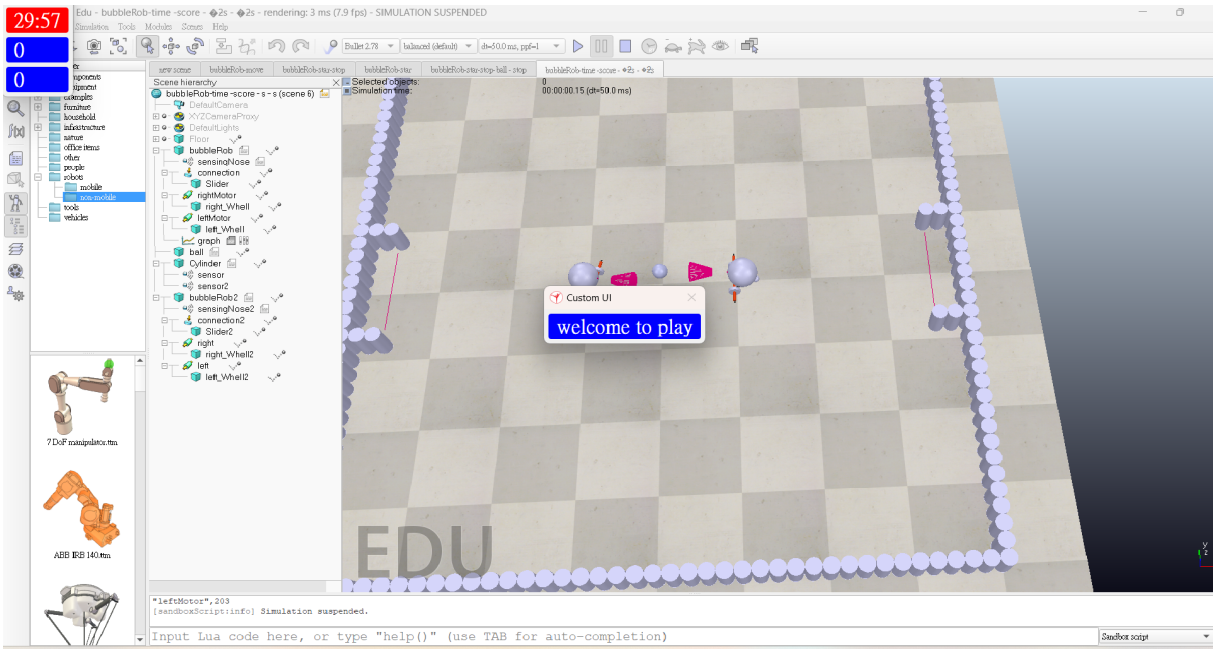

圖. 2.6: 歡迎

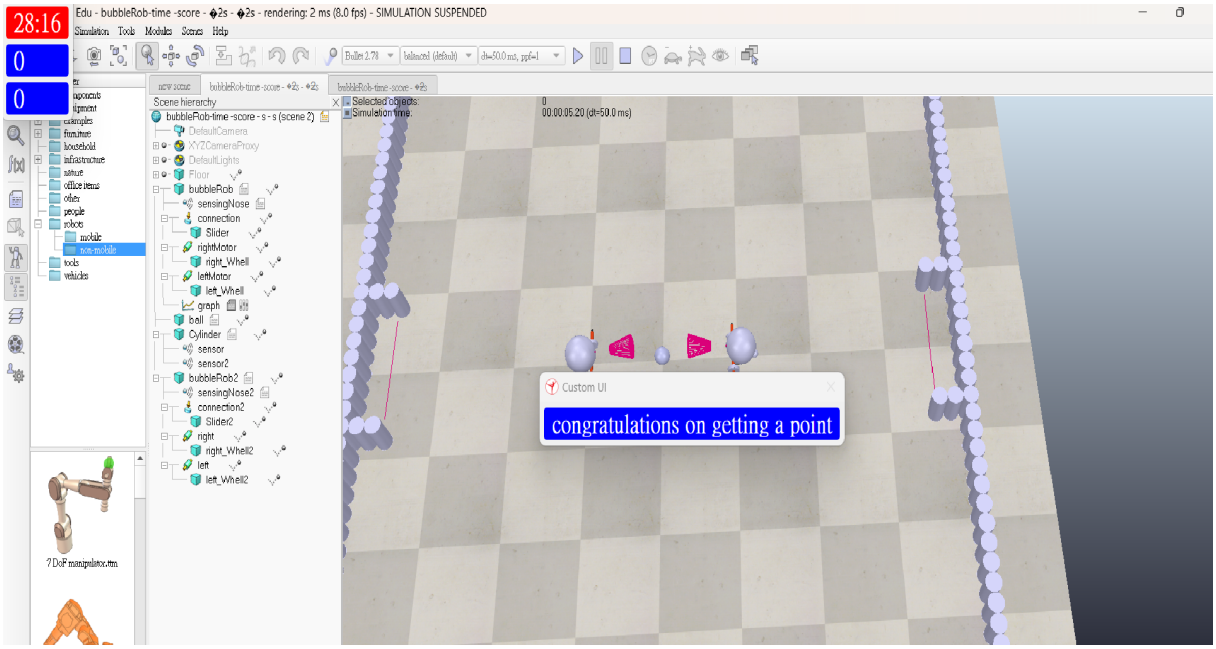

圖. 2.7: 恭喜

```
function sysCall_init()
    xml1 = [[
        <ui closeable="false" resizeable="false" activate="false">
            <label text="welcome to play" style="* {background-color: #000F; color: #FFF; font-size: 32px; font-weight: bold; padding: 4px; border-radiu
        </ui>
    ]]
    ui1 = simUI.create(xml1)
    add = true
    -- Pause simulation on the first run
    sim.pauseSimulation(true)
end

function sysCall_actuation()
    simUI.hide(ui1)


end

function sysCall_suspend()
    simUI.show(ui1)
    --simUI.setLabelText(ui1, 1, "good game")
    if add == false then
        simUI.setLabelText(ui1, 1, "congratulations on getting a point")
    end
    add = false
end
```

圖. 2.8: 歡迎與恭喜 lua

2.5 RemoteApi

開啟 19998 與 23020 埠號來提供玩家使用 RemoteApi 連線

將前面使用 lua 編寫的控制移動程式轉為使用 python 編寫, 並使用 RemoteApi 進行連線在模擬未開啟前, 可以使用 19997 默認埠號進行連線, 須注意連線時 coppeliasim 畫面必須開啟

```
function sysCall_init()
-- 場景模擬開始時開啟19998與23020埠號
simRemoteApi.start(19998)
simRemoteApi.start(23020)
end
```

圖. 2.9: 開啟埠號

```python
#bubbleRob.py
import sim
import sys, math
import keyboard
import time

with open('my_ip.txt', 'r') as f:
    my_ip1 = f.readlines()
    for line in my_ip1:
        if 'game' in line: #game #myip #local
            ip = line.split(':')[1].strip()
            print(ip)

# 連接到 CoppeliaSim simulation
sim.simxFinish(-1)
clientID = sim.simxStart(ip, 19997, True, True, 5000, 5)
sim.simxStartSimulation(clientID, sim.simx_opmode_oneshot_wait)

if clientID != -1:
    print("已連線到遠端 CoppeliaSim 伺服器")
else:
    print('連線失敗')
    sys.exit('無法連線到 CoppeliaSim 伺服器')

# 取得馬達與感測器的 handles
errorCode, leftMotor = sim.simxGetObjectHandle(clientID, 'leftMotor', sim.simx_opmode_oneshot_wait)
errorCode, rightMotor = sim.simxGetObjectHandle(clientID, 'rightMotor', sim.simx_opmode_oneshot_wait)
errorCode, sensingNose = sim.simxGetObjectHandle(clientID, 'sensingNose', sim.simx_opmode_oneshot_wait)

# 設定一些參數
deg = math.pi/180
paused = False
if errorCode == -1:
    print('找不到左右馬達')
    sys.exit()

def jointspeed(left,right):
    errorCode4=sim.simxSetJointTargetVelocity(clientID,leftMotor,left, sim.simx_opmode_oneshot)
    errorCode5=sim.simxSetJointTargetVelocity(clientID,rightMotor,right, sim.simx_opmode_oneshot_wait)
errorCode, number2 = sim.simxLoadModel(clientID, 'number2.ttm', 0, sim.simx_opmode_oneshot_wait)
while sim.simxGetConnectionId(clientID) != -1:
    event = keyboard.read_event()
    if event.event_type == 'down':
        print('The "' + event.name + '" key was pressed.')
    if event.name == 'a' :
        jointspeed(-3,5)
    elif event.name == 'w' :
        jointspeed(5,5)
    elif event.name == 's' :
        jointspeed(-5,-5)
    elif event.name == 'd' :
        jointspeed(5,-3)
    if event.name == 'p':
        if not paused:
            print('Paused')
            sim.simxPauseSimulation(clientID, sim.simx_opmode_oneshot_wait)
            paused = True
            time.sleep(0.1)
        else:
            print('Resumed')
            sim.simxStartSimulation(clientID, sim.simx_opmode_oneshot_wait)
            paused = False
            time.sleep(0.1)
```

圖. 2.10: python 操控

參考文獻

[1] https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[2] https://towardsdatascience.com/derivative-of-the-sigmoid-function-536880cf918e

[3] http://www.incompleteideas.net/book/RLbook2020.pdf

[4] https://medium.com/change-the-world-with-technology/policy-gradient-181d43a24cf5

[5] https://livebook.manning.com/book/grokking-deep-reinforcement-learning/chapter-11/v-11/38

[6] http://ukko.life.nctu.edu.tw/ u0517047/usage.html

[7] https://lilianweng.github.io/lil-log/2018/04/08/policy-gradient-algorithms.html

[8] https://uupgrade.medium.com/python-那些年我們一起玩過的遊戲-三-打磚塊-d89b648896ca

[9] https://cvfiasd.pixnet.net/blog/post/275774124-深度學習激勵函數介紹

[10] https://www.coppeliarobotics.com/helpFiles/

[11] https://hackernoon.com/the-reason-behind-moving-in-the-direction-opposite-to-the-gradient-f9566b95370b

[12] https://ruder.io/optimizing-gradient-descent/

[13] https://zh.wikipedia.org/wiki/HSL 和 HSV 色彩空間

[14] https://gist.github.com/karpathy/a4166c7fe253700972fcbc77e4ea32c5#file-pg-pong-py

[15] https://github.com/schinger/pong_ actor-critic/blob/master/pg-pong-ac.py

[16] https://gist.github.com/etienne87/6803a65653975114e6c6f08bb25e1522

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。

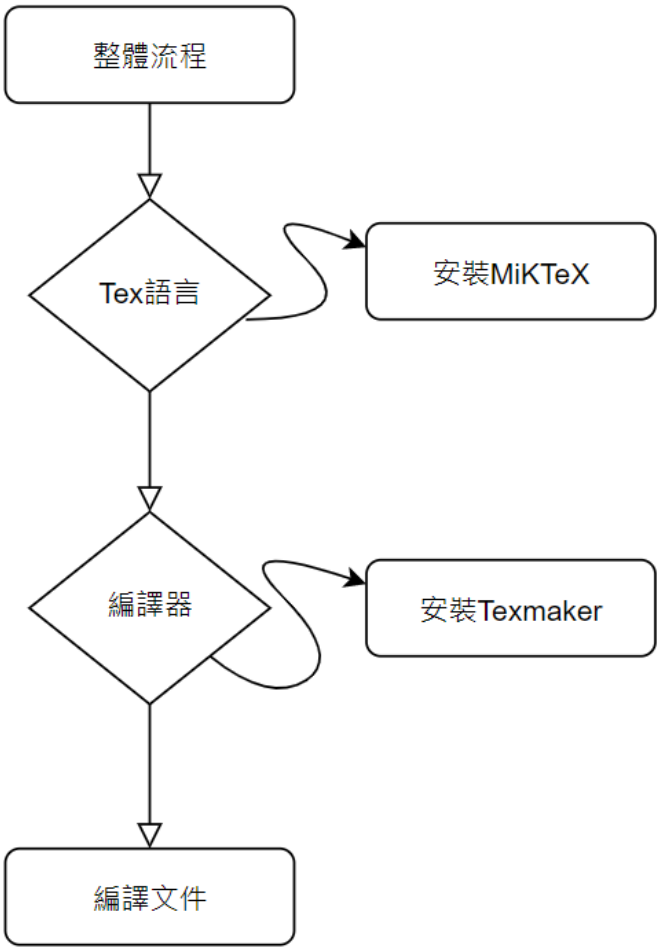

圖. 11: 編譯流程

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。