

國立虎尾科技大學

機械設計工程系

# 協同機器人踢足球設計

## Web-based Football Scene Design

指導教授:嚴家銘老師

班 級:四設三乙

學 生:林立翔 (40723217)

陳柏成 (40923219)

黃亦銘 (40923225)

蔡淳宇 (40923241)

蕭仲佑 (40923246)

黃柏儒 (40971219)

傅柏皓 (40971220)

組 別:2b3-pj3bg1

中華民國一百一十二年六月

## 摘要

本專案旨在進行協同產品設計，改良並且優化多輪車，同時應用於機器人足球比賽中。而專案分為三個階段，階段三是對階段二的延伸。本團隊透過分組由7名成員組成，使用CAD軟體繪製場景和多輪車設計。利用ZmqRemoteAPI Python編程，開發控制程式以操控多輪車在足球場景中進行比賽。

階段三的目標是改進多輪車的移動和對戰效能。為了實現這個目標，團隊著重優化運動控制，以實現多輪車的靈活操控能力。同時，協同策略在階段三中發揮了關鍵作用，使多輪車能夠與團隊協調分工，在比賽中進行配合和合作。

在不斷優化控制系統的過程中，團隊進行數次甚至數十次的模擬和測試，並根據測試結果和出現的問題不斷進行改進和優化。在最後團隊將提供相關檔案的下載連結，並製作線上簡報和分組報告，以展示協同設計流程和成果。

透過本專案團隊獲得了實踐協同工作與設計創新的機會，同時提升了多輪車在足球比賽中的性能和效能。為未來的機器人技術和應用留下了歷史的一小頁。

# 目錄

第一章 緒論	1
第二章 動機與目的	2
第三章 方法與步驟圖表	3
3.1 機器人繪製	3
3.2 程式編程	5
3.3 測試過程	9
第四章 結論與建議	19
第五章 參考文獻	20
Pj3分組工作	21

# 第一章 緒論

本篇場景連線設計專案旨在探討如何在 CoppeliaSim 環境中建立連接並共享場景，以便多個使用者能夠同時觀看和參與仿真模擬。在這個專案中，我們將使用 ZMQ Remote API 和 IPv6 技術來實現場景連線。

在現代的工程和科學領域中，虛擬仿真已成為一種重要的工具，用於設計、測試和驗證各種系統。CoppeliaSim 提供了一個強大的仿真平台，並且通過 ZMQ Remote API，我們可以在不同的電腦之間建立連接，實現分散式的仿真環境。

在這個實驗中，我們使用 CoppeliaSim 的 ZMQ Remote API 和 IPv6 技術來建立場景連線。我們遵循了幾個關鍵步驟來解決連接問題。首先，我們確保所有電腦都連接到相同的局域網中，確保網路連通性。接下來，我們設定了 ZMQ Remote API 的 IP 位址和埠號，以便其他電腦可以連接到我們的 CoppeliaSim 場景。同時，我們確保防火牆設定允許外部訪問所使用的埠號，以避免任何阻礙連接的限制。最後，我們將我們的 IPv6 位址提供給其他人，這樣他們就可以輕鬆連接並觀看我們的 CoppeliaSim 場景。這個實驗提供了一個實用的解決方案，讓人們能夠在仿真模擬中共享場景並進行協作。

## 第二章 動機與目的

本次實驗的動機是探索在 CoppeliaSim 環境中建立場景連線的方法，以實現多用戶的協同觀看和參與。在現實世界中，協作和共享資源是實現創新和合作的關鍵。在仿真模擬領域，場景連線的實現可以讓不同的使用者在同一個虛擬環境中共同操作和觀察。這樣的功能可以被廣泛應用於教育、研究和工程等領域。

我們的目的是解決一個關鍵問題，即如何讓其他人輕鬆地連接到我們的 CoppeliaSim 場景。傳統上，場景連線可能需要複雜的設置和配置，導致使用者面臨技術困難和門檻。因此，我們希望通過本實驗，提供一個簡單且易於使用的方法，讓任何人都能夠在不同的電腦和網路環境下連接到我們的 CoppeliaSim 場景，進行協同觀看和參與。

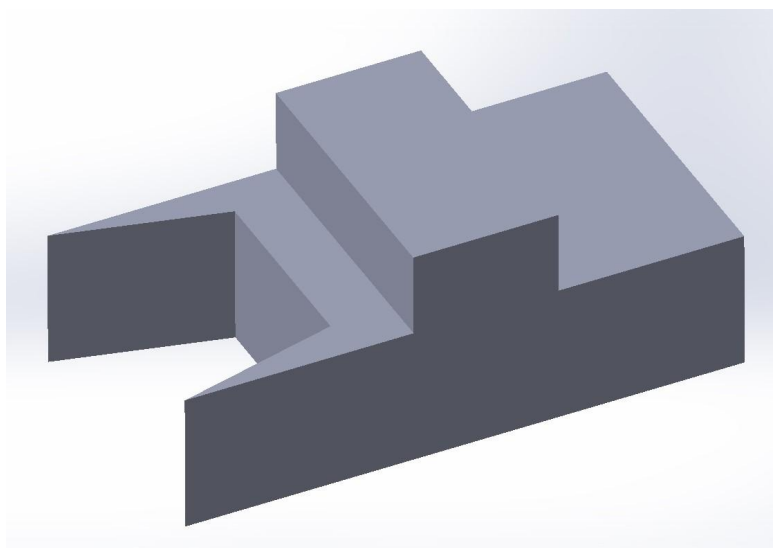
透過這次的實驗，我們將提供一個完整的解決方案，涵蓋了局域網連接、ZMQ Remote API 設定和防火牆配置等方面。我們的目標是讓人們能夠輕鬆地共享和參與 CoppeliaSim 的仿真模擬環境，促進知識交流、合作和創新。

## 第三章 方法與步驟圖表

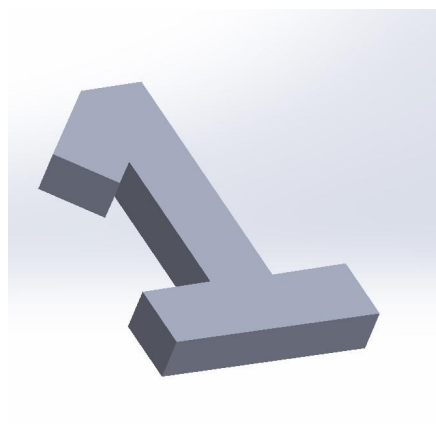
### 3.1 機器人繪製

為讓機器人與其背號號碼在CoppeliaSim裡能擁有不同的顏色，首先使用inventer繪製作為基本車子造型的機器人球員本體，再使用solidwork繪製機器人球員背號號碼的1到8號數字，再將機器人與數字分別匯入CoppeliaSim進行組裝。

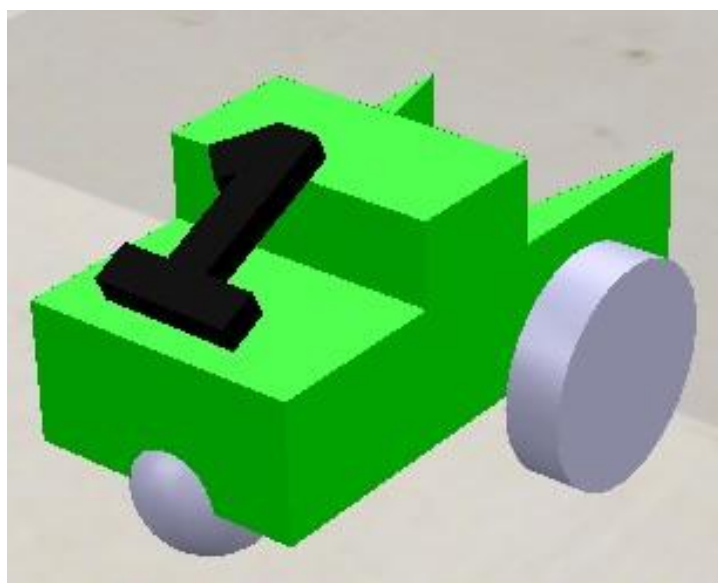
機器人球員本體



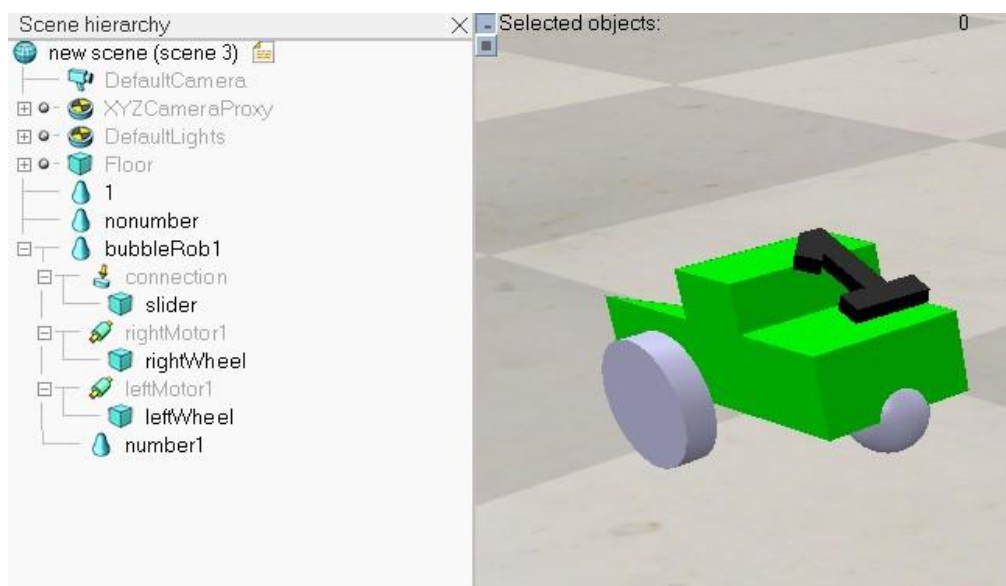
背號數字



於CoppeliaSim軟體進行組裝



在左右輪加入軸使其能夠旋轉，並使用連結器連結輔助輪與機器  
人



## 3.2 程式編程

CoppeliaSIM場景內程式編寫:

```
1 function sysCall_init()  
2     bubbleRob1 = sim.getObject('./bubbleRob1')  
3     bubbleRob2 = sim.getObject('./bubbleRob2')  
4     bubbleRob3 = sim.getObject('./bubbleRob3')  
5     bubbleRob4 = sim.getObject('./bubbleRob4')  
6     bubbleRob5 = sim.getObject('./bubbleRob5')  
7     bubbleRob6 = sim.getObject('./bubbleRob6')  
8     bubbleRob7 = sim.getObject('./bubbleRob7')  
9     bubbleRob8 = sim.getObject('./bubbleRob8')  
10    sensor = sim.getObject('./sensor')  
11    sensor2 = sim.getObject('./sensor2')  
12    ball = sim.getObject('./ball')  
13  
14    local math = require("math")  
15    joint1= sim.getObject('./joint1')  
16    joint2= sim.getObject('./joint2')  
17    joint3= sim.getObject('./joint3')  
18    joint4= sim.getObject('./joint4')  
19    sensor = sim.getObject('./sensor')
```

先將場景內的物件定好變數名稱

```
20    r1 = 360  
21    r2 = 360  
22    r3 = 360  
23    r4 = 360  
24    sim.setJointTargetPosition(joint1, math.rad(r1+180))  
25    sim.setJointTargetPosition(joint2, math.rad(r2+180))  
26    sim.setJointTargetPosition(joint3, math.rad(r3+180))  
27    sim.setJointTargetPosition(joint4, math.rad(r4+180))  
28
```

這邊則是為轉盤式計分做好準備



```

29     pos1 = {0.5, 1, 0.2}
30     pos2 = {1, 0.5, 0.2}
31     pos3 = {1, -0.5, 0.2}
32     pos4 = {0.5, -1, 0.2}
33     pos5 = {-0.5, -1, 0.2}
34     pos6 = {-1, -0.5, 0.2}
35     pos7 = {-1, 0.5, 0.2}
36     pos8 = {-0.5, 1, 0.2}
37     posS = {0, 0, 0.2}
38     ang1 = {1.57, 0, 0}
39     ang2 = {0, 0, 3.14}
40     ang3 = {0, -1.57, -1.57}
41     ang4 = {0, 0, 0}

```

定好個物件的初始位置，也方便進球後重制位置用

```

43     score1 = 0
44     score2 = 0
45     score3 = 0
46     score4 = 0
47     count = 60000
48     count1 = 0
49     score5 = 99
50     score6 = 0
51     score7 = 0
52     score8 = 1
53     s0={1, 1, 1, 1, 1, 0, 1}
54     s1={0, 0, 1, 1, 0, 0, 0}
55     s2={0, 1, 1, 0, 1, 1, 1}
56     s3={0, 0, 1, 1, 1, 1, 1}
57     s4={1, 0, 1, 1, 0, 1, 0}
58     s5={1, 0, 0, 1, 1, 1, 1}
59     s6={1, 1, 0, 1, 1, 1, 1}
60     s7={0, 0, 1, 1, 1, 0, 0}
61     s8={1, 1, 1, 1, 1, 1, 1}
62     s9={1, 0, 1, 1, 1, 1, 1}
63     s={s0, s1, s2, s3, s4, s5, s6, s7, s8, s9}

```

接下來用到的變數：

score1~4: 記分板顯示的數字

count、score5: 計時用

score6~8、count1:顯示在計時器上的數字

S陣列:七段顯示的設定值s0~s9即為0~9的七段顯示

```
64 | for j = 0,6,1 do
65 |   a = sim.getObject('/_G[\"..j..\"]')
66 |   if (s[1][j+1]==1) then
67 |     sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
68 |   else
69 |     sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
70 |   end
71 | end
```

初始化並分配記分、計時板，這架構有8個，初始設定也就此結束。

```
129 | function sysCall_actuation()
130 |   result = sim.readProximitySensor(sensor) |
131 |   if(r2>0 and result>0)then
132 |     sim.setObjectPosition(ball, -1, posS)
133 |     sim.setObjectOrientation(ball, -1, ang4)
134 |     sim.setObjectPosition(bubbleRob1, -1, pos1)
135 |     sim.setObjectPosition(bubbleRob2, -1, pos2)
136 |     sim.setObjectPosition(bubbleRob3, -1, pos3)
137 |     sim.setObjectPosition(bubbleRob4, -1, pos4)
138 |     sim.setObjectPosition(bubbleRob5, -1, pos5)
139 |     sim.setObjectPosition(bubbleRob6, -1, pos6)
140 |     sim.setObjectPosition(bubbleRob7, -1, pos7)
141 |     sim.setObjectPosition(bubbleRob8, -1, pos8)
142 |     sim.setObjectOrientation(bubbleRob1, -1, ang3)
143 |     sim.setObjectOrientation(bubbleRob2, -1, ang1)
144 |     sim.setObjectOrientation(bubbleRob3, -1, ang1)
145 |     sim.setObjectOrientation(bubbleRob4, -1, ang4)
146 |     sim.setObjectOrientation(bubbleRob5, -1, ang2)
147 |     sim.setObjectOrientation(bubbleRob6, -1, ang2)
148 |     sim.setObjectOrientation(bubbleRob7, -1, ang2)
149 |     sim.setObjectOrientation(bubbleRob8, -1, ang2)
150 |     score1 = score1 + 1
151 |     r1=r1-36
152 |     sim.setJointTargetPosition(joint1, math.rad(r1+180))
153 |     if(r1==0) then
154 |       r1=360
155 |       r2=r2-36
156 |       sim.setJointTargetPosition(joint1, math.rad(r1+180))
157 |       sim.setJointTargetPosition(joint2, math.rad(r2+180))
158 |     end
```

```

159   if (score1 <= 9) then
160     i = score1 +1
161     for j = 0, 6, 1 do
162       a = sim.getObject('/_R['..j..'']')
163       if (s[i][j+1] == 1) then
164         sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
165       else
166         sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
167       end
168     end
169   elseif (score1 > 9 and score2<=9) then
170     score1 = 0
171     for j = 0, 6, 1 do
172       a = sim.getObject('/_R['..j..'']')
173       if (s[1][j+1] == 1) then
174         sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
175       else
176         sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
177       end
178     end
179     score2 = score2 + 1
180     i = score2 +1
181     for j = 0, 6, 1 do
182       b = sim.getObject('/_R['..j..'']')
183       if (s[i][j+1] == 1) then
184         sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
185       else
186         sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
187       end
188     end
189   else
190     sim.pauseSimulation()
191   end
192 end

```

循環開始，球門會持續檢測進球，並執行：

進球後，球和球員回到初始位置

轉盤角度調整後檢查並執行進位

分數加一

檢查進位並將分數顯示到七段顯示器上

另一方的球門也有一樣的檢測系統，兩者都有分數超過99即暫停模擬的機制。

```

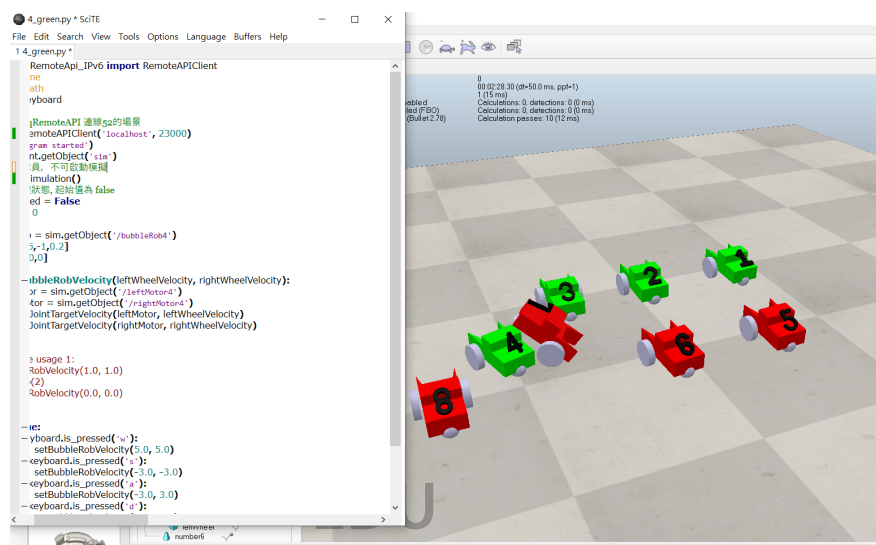
256 if (count > 0) then
257   score5 = score5 - 1
258   if (score5 < 1) then
259     score5 = 99
260     count1 = count1 - 1
261   end
262   if (count1 >= 0) then
263     for i = 0, 9, 1 do
264       if (count1 == i) then
265         for j = 0, 6, 1 do
266           local e = sim.getObject('/_/_G['..j..'']')
267           if (s[i+1][j+1] == 1) then
268             sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
269           else
270             sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
271           end
272         end
273       end
274     end
275   end
276   elseif (score6 > 0 and count1 < 1) then
277     count1 = 9
278     score6 = score6 - 1
279     for i = 0, 9, 1 do
280       for j = 0, 6, 1 do
281         local e = sim.getObject('/_/_G['..j..'']')
282         if (s[i+1][j+1] == 1) then
283           sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
284         else
285           sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
286         end
287       end
288     end
289     if (score6 == i) then
290       for j = 0, 6, 1 do
291         local f = sim.getObject('/_/_C['..j..'']')
292         if (s[i+1][j+1] == 1) then
293           sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
294         else
295           sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
296         end
297       end
298     end
299   end
300 end

```

最後則是計時，由於是倒計時，進位方法與得分有些許不同，但整體大同小異，值得一提的是計時的方法：設定一值後每次循環都讓他減一，是為百分之一秒，以此再做換算便能倒計時十分鐘。

## 測試過程：

利用SciTe控制各機器人球員的python執行檔啟動CoppeliaSim場景，進行測試。





上圖為我們設計機器人在場景里移動的程式，包括設定左右輪速度及重生時回到開始點等。

## 完整程式碼

1. `function sysCall_init()`
2. `bubbleRob1 = sim.getObject('/bubbleRob1')`
3. `bubbleRob2 = sim.getObject('/bubbleRob2')`
4. `bubbleRob3 = sim.getObject('/bubbleRob3')`
5. `bubbleRob4 = sim.getObject('/bubbleRob4')`
6. `bubbleRob5 = sim.getObject('/bubbleRob5')`

```

7. bubbleRob6 = sim.getObject('/bubbleRob6')
8. bubbleRob7 = sim.getObject('/bubbleRob7')
9. bubbleRob8 = sim.getObject('/bubbleRob8')
10. sensor = sim.getObject('/sensor')
11. sensor2 = sim.getObject('/sensor2')
12. ball = sim.getObject('/ball')
13.
14. local math = require("math")
15. joint1= sim.getObject('/joint1')
16. joint2= sim.getObject('/joint2')
17. joint3= sim.getObject('/joint3')
18. joint4= sim.getObject('/joint4')
19. sensor = sim.getObject('/sensor')
20. r1 = 360
21. r2 = 360
22. r3 = 360
23. r4 = 360
24. sim.setJointTargetPosition(joint1, math.rad(r1+180))
25. sim.setJointTargetPosition(joint2, math.rad(r2+180))
26. sim.setJointTargetPosition(joint3, math.rad(r3+180))
27. sim.setJointTargetPosition(joint4, math.rad(r4+180))
28. --你好
29. pos1 = {0.5,1,0.2}
30. pos2 = {1,0.5,0.2}
31. pos3 = {1,-0.5,0.2}
32. pos4 = {0.5,-1,0.2}
33. pos5 = {-0.5,-1,0.2}
34. pos6 = {-1,-0.5,0.2}
35. pos7 = {-1,0.5,0.2}
36. pos8 = {-0.5,1,0.2}
37. posS = {0,0,0.2}
38. ang1 = {1.57,0,0}
39. ang2 = {0,0,3.14}
40. ang3 = {0,-1.57,-1.57}
41. ang4 = {0,0,0}
42. bubbleRobBase = 4
43.
44. score1 = 0
45. score2 = 0
46. score3 = 0
47. score4 = 0
48. count = 60000
49. count1 = 0
50. score5 = 99
51. score6 = 0
52. score7 = 0
53. score8 = 1

```

```

54. s0={1,1,1,1,1,0,1}
55. s1={0,0,1,1,0,0,0}
56. s2={0,1,1,0,1,1,1}
57. s3={0,0,1,1,1,1,1}
58. s4={1,0,1,1,0,1,0}
59. s5={1,0,0,1,1,1,1}
60. s6={1,1,0,1,1,1,1}
61. s7={0,0,1,1,1,0,0}
62. s8={1,1,1,1,1,1,1}
63. s9={1,0,1,1,1,1,1}
64. s={s0,s1,s2,s3,s4,s5,s6,s7,s8,s9}
65. for j = 0,6,1 do
66. a = sim.getObject('/_G['..j..']')
67. if (s[1][j+1]==1) then
68. sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
69. else
70. sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
71. end
72. end
73. for j = 0,6,1 do
74. b = sim.getObject('/_G['..j..']')
75. if (s[1][j+1]==1) then
76. sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
77. else
78. sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
79. end
80. end
81. for j = 0,6,1 do
82. d = sim.getObject('/_R['..j..']')
83. if (s[1][j+1]==1) then
84. sim.setShapeColor(d, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
85. else
86. sim.setShapeColor(d, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
87. end
88. end
89. for j = 0,6,1 do
90. c = sim.getObject('/_R['..j..']')
91. if (s[1][j+1]==1) then
92. sim.setShapeColor(c, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
93. else
94. sim.setShapeColor(c, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
95. end
96. end
97. for j = 0,6,1 do
98. local g = sim.getObject('/_C['..j..']')
99. if (s[1][j+1]==1) then
100. sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})

```

```

101.else
102.sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
103.end
104.end
105.for j = 0,6,1 do
106.local h = sim.getObject('./C_['..j..']')
107.if (s[1][j+1]==1) then
108.sim.setShapeColor(h, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
109.else
110.sim.setShapeColor(h, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
111. end
112.end
113.for j = 0,6,1 do
114.local e = sim.getObject('./__C_['..j..']')
115.if (s[1][j+1]==1) then
116.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
117.else
118.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
119.end
120.end
121.for j = 0,6,1 do
122.local f = sim.getObject('./_C_['..j..']')
123.if (s[1][j+1]==1) then
124.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
125.else
126.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
127.end
128.end
129.end
130.function sysCall_actuation()
131.result = sim.readProximitySensor(sensor)
132.--print(s[i])
133.if(r2>0 and result>0)then
134.sim.setObjectPosition(ball, -1, posS)
135.sim.setObjectOrientation(ball, -1, ang4)
136.sim.setObjectPosition(bubbleRob1, -1, pos1)
137.sim.setObjectPosition(bubbleRob2, -1, pos2)
138.sim.setObjectPosition(bubbleRob3, -1, pos3)
139.sim.setObjectPosition(bubbleRob4, -1, pos4)
140.sim.setObjectPosition(bubbleRob5, -1, pos5)
141.sim.setObjectPosition(bubbleRob6, -1, pos6)
142.sim.setObjectPosition(bubbleRob7, -1, pos7)
143.sim.setObjectPosition(bubbleRob8, -1, pos8)
144.sim.setObjectOrientation(bubbleRob1, -1, ang3)
145.sim.setObjectOrientation(bubbleRob2, -1, ang1)
146.sim.setObjectOrientation(bubbleRob3, -1, ang1)
147.sim.setObjectOrientation(bubbleRob4, -1, ang4)

```



```

148.sim.setObjectOrientation(bubbleRob5, -1, ang2)
149.sim.setObjectOrientation(bubbleRob6, -1, ang2)
150.sim.setObjectOrientation(bubbleRob7, -1, ang2)
151.sim.setObjectOrientation(bubbleRob8, -1, ang2)
152.score1 = score1 + 1
153.r1=r1-36
154.sim.setJointTargetPosition(joint1, math.rad(r1+180))
155.if(r1==0) then
156.r1=360
157.r2=r2-36
158.sim.setJointTargetPosition(joint1, math.rad(r1+180))
159.sim.setJointTargetPosition(joint2, math.rad(r2+180))
160.end
161.if (score1 <= 9) then
162.i = score1 +1
163.for j = 0, 6, 1 do
164.a = sim.getObject('/_R['..j..']')
165.if (s[i][j+1] == 1) then
166.sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
167.else
168.sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
169.end
170.end
171.elseif (score1 > 9 and score2<=9) then
172.score1 = 0
173.for j = 0, 6, 1 do
174.a = sim.getObject('/_R['..j..']')
175.if (s[1][j+1] == 1) then
176.sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
177.else
178.sim.setShapeColor(a, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
179.end
180.end
181.score2 = score2 + 1
182.i = score2 +1
183.for j = 0, 6, 1 do
184.b = sim.getObject('/R_['..j..']')
185.if (s[i][j+1] == 1) then
186.sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {0, 1, 0})
187.else
188.sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
189.end
190.end
191.else
192.sim.pauseSimulation()
193.end
194.end

```

```

195.result2 = sim.readProximitySensor(sensor2)
196.if(r4>0 and result2>0)then
197.sim.setObjectPosition(ball, -1, posS)
198.sim.setObjectOrientation(ball, -1, ang4)
199.sim.setObjectPosition(bubbleRob1, -1, pos1)
200.sim.setObjectPosition(bubbleRob2, -1, pos2)
201.sim.setObjectPosition(bubbleRob3, -1, pos3)
202.sim.setObjectPosition(bubbleRob4, -1, pos4)
203.sim.setObjectPosition(bubbleRob5, -1, pos5)
204.sim.setObjectPosition(bubbleRob6, -1, pos6)
205.sim.setObjectPosition(bubbleRob7, -1, pos7)
206.sim.setObjectPosition(bubbleRob8, -1, pos8)
207.sim.setObjectOrientation(bubbleRob1, -1, ang3)
208.sim.setObjectOrientation(bubbleRob2, -1, ang1)
209.sim.setObjectOrientation(bubbleRob3, -1, ang1)
210.sim.setObjectOrientation(bubbleRob4, -1, ang4)
211.sim.setObjectOrientation(bubbleRob5, -1, ang2)
212.sim.setObjectOrientation(bubbleRob6, -1, ang2)
213.sim.setObjectOrientation(bubbleRob7, -1, ang2)
214.sim.setObjectOrientation(bubbleRob8, -1, ang2)
215.score3 = score3 + 1
216.r3=r3-36
217.sim.setJointTargetPosition(joint3, math.rad(r3+180))
218.if(r3==0) then
219.r3=360
220.r4=r4-36
221.sim.setJointTargetPosition(joint3, math.rad(r3+180))
222.sim.setJointTargetPosition(joint4, math.rad(r4+180))
223.end
224.if (score3 <= 9) then
225.i = score3 +1
226.for j = 0, 6, 1 do
227.d = sim.getObject('._G[.j..]')
228.if (s[i][j+1] == 1) then
229.sim.setShapeColor(d, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
230.else
231.sim.setShapeColor(d, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
232.end
233.end
234.elseif (score3 > 9 and score4<=9) then
235.score3 = 0
236.for j = 0, 6, 1 do
237.c = sim.getObject('._G[.j..]')
238.if (s[1][j+1] == 1) then
239.sim.setShapeColor(c, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
240.else
241.sim.setShapeColor(c, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})

```

```

242.end
243.end
244.score4 = score4 + 1
245.i = score4 +1
246.for j = 0, 6, 1 do
247.b = sim.getObject('/G_['..j..']')
248.if (s[i][j]+1) == 1) then
249.sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 0, 0})
250.else
251.sim.setShapeColor(b, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
252.end
253.end
254.else
255.sim.pauseSimulation()
256.end
257.end
258.if (count > 0) then
259.score5 = score5 - 1
260.if(score5<1)then
261.score5 = 99
262.count1 = count1-1
263.end
264.if (count1 >= 0) then
265.
266.for i = 0, 9, 1 do
267.if (count1 == i) then
268.for j = 0, 6, 1 do
269.local e = sim.getObject('/_C_['..j..']')
270.if (s[i+1][j]+1) == 1) then
271.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
272.else
273.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
274.end
275.end
276.end
277.end
278.elseif (score6 > 0 and count1 <1) then
279.count1 = 9
280.score6 = score6 - 1
281.for i = 0 , 9, 1 do
282.for j = 0, 6, 1 do
283.local e = sim.getObject('/_C_['..j..']')
284.if (s[i+1][j]+1) == 1) then
285.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
286.else
287.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
288.end

```

```

289.end
290.if (score6 == i) then
291.for j = 0, 6, 1 do
292.local f = sim.getObject('/C_['..j..']')
293.if (s[i+1][j+1] == 1) then
294.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
295.else
296.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
297.end
298.end
299.end
300.end
301.elseif (score7 > 0 and score6 < 1) then
302.score5 = 9
303.score6 = 6
304.score7 = score7 - 1
305.for i = 0, 9, 1 do
306.for j = 0, 6, 1 do
307.local e = sim.getObject('/_C_['..j..']')
308.local f = sim.getObject('/C_['..j..']')
309.if (s[i+1][j+1] == 1) then
310.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
311.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
312.else
313.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
314.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
315.end
316.end
317.if (score7 == i) then
318.for j = 0, 6, 1 do
319.local g = sim.getObject('/__C_['..j..']')
320.if (s[i+1][j+1] == 1) then
321.sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
322.else
323.sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
324.end
325.end
326.end
327.end
328.elseif (score8 > 0 and score7 < 1) then
329.score5 = 9
330.score6 = 6
331.score7 = 9
332.score8 = score8 - 1
333.for i = 0, 9, 1 do
334.for j = 0, 6, 1 do
335.local e = sim.getObject('/_C_['..j..']')

```

```

336.local f = sim.getObject('./C_['..j..']')
337.local g = sim.getObject('./_C_['..j..']')
338.if (s[i+1][j+1] == 1) then
339.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
340.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
341.sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
342.else
343.sim.setShapeColor(e, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
344.sim.setShapeColor(f, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
345.sim.setShapeColor(g, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
346.end
347.end
348.if (score8 == i) then
349.for j = 0, 6, 1 do
350.local h = sim.getObject('./_C_['..j..']')
351.if (s[i+1][j+1] == 1) then
352.sim.setShapeColor(h, nil, sim.colorcomponent_ambient_diffuse, {0, 0, 0})
353.else
354.sim.setShapeColor(h, nil, sim.colorcomponent_ambient_diffuse, {1, 1, 1})
355.end
356.end
357.end
358.end
359.else
360.
361.sim.stopSimulation()
362.end
363.else
364.sim.stopSimulation()
365.end
366.end

```

## 第四章 結論與建議

在本次實驗中，我們成功地探索了在CoppeliaSim環境中建立場景連線的方法。通過設置ZMQ Remote API和調整防火牆設定，我們能夠讓不同的使用者從不同的電腦和網路環境下連接到共享主機的CoppeliaSim場景，實現了協同觀看和參與的目標。我們透過IPv6的支援，擴大了連接的範圍，讓使用者能夠更加靈活且方便地連接到CoppeliaSim環境。

然而，這次實驗也揭示出了一些挑戰和問題。例如，在機器人踢球遊戲中，當球進入球門時，我們面臨了確定最後得分的機器人的問題。這涉及到場景中多個機器人的識別和計分機制，需要更深入的設計和開發。

此外，在場景連線中，客戶端能夠使用指令去更改模擬的結果，這可能導致一些不正當的操作和類似開外掛的行為。這需要我們思考如何實現權限控制和遠程操作的安全性，以確保場景的公平性和合法性。

基於上述問題，我們建議進一步探討如何實現多機器人識別和計分機制，以及如何確保場景連線中的安全性和合法性。這些議題可以成為未來研究的方向，並為CoppeliaSim環境中的協同觀看和參與帶來更完善的解決方案。

透過這次實驗的探索和討論，我們為場景連線的實現奠定了基礎，並提出了一些值得進一步研究和改進的問題。我們相信這將為仿真模擬領域的協作和共享帶來更大的可能性，並促進知識交流和創新的發展。

## 第五章 參考文獻

[如何透過 zmq 以鍵盤控制 BubbleRob 多輪車](#)

[BubbleRob 多輪車程式參考](#)

[場地模板與轉軸式記分版參考](#)

[深入了解協同產品設計與如何在多人協同環境中執行產品開發](#)

## Pj3分組工作

40723217 林立翔:建設連線環境, 整合設定場景物件

40923219 陳柏成:七段顯示器計分板、計時器建模及程式編寫

40923225 黃亦銘:場景建模、統整各組員資料、簡報撰寫

40923241 蔡淳宇:場景建模、球員設計及建模

40923246 蕭仲佑:轉軸式計分板建模及程式編寫, 建設連線環境

40971219 黃柏儒:球員設計

40923220 傅柏皓:scite程式編寫, 報告撰寫