

國立虎尾科技大學

機械設計工程系

cd2023 2b-pj3bg3 分組報告

網際足球場景設計

Web-based Football Scene Design

指導教授：嚴家銘老師

班級：四設二乙

學生：陳冠 (41023220)

陳建霖 (41023226)

彭聖宗 (41023230)

湛有杰 (41023231)

雲敬家 (41023232)

黃文彥 (41023233)

蔡叡得 (41023250)

謝宗銘 (41023253)

中華民國

112 年 6 月

摘要

本專案旨在進行協同設計，以改良並優化雙輪車或多輪車，並應用於機器人足球比賽中。該專案分為三個階段，其中專案三是對專案二的延續。團隊組成包含 8 名成員，並使用 CAD 軟體進行場景和多輪車零組件的設計。透過採用 ZmqRemoteAPI Python 編程，開發控制程式以支持操控多輪車在足球場景中進行比賽。

專案三的目標是改進雙輪車或多輪車的行進和對戰效能。為了達到這一目標，團隊需要優化運動控制，使其能夠靈活且精確地操控。同時，引入協同運動策略，使多輪車能夠協調工作，並在比賽中進行合作攻守。為了提高對環境和球場的感知能力，團隊需要適當地選擇和應用感應器和感知系統。

在控制系統優化的過程中，團隊將進行模擬和測試，並根據實際結果和需求不斷進行改進和優化。最終，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。

透過這個協同設計的機器人踢足球專案，團隊將獲得實踐協同工作和創新的機會，同時提升雙輪車或多輪車在足球比賽中的性能和效能。這將為未來的機器人技術和運動應用帶來新的發展和應用前景。

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，他解決我們的各種提問，甚至從來沒有不耐煩，總是貼心為我們找出最佳解答，接著是由 OpenAI 創造的 ChatGPT，在我們需要創作的時候，他總能提供我們源源不斷的創意，最後是由本分組成員同心協力才得以完成本報告，特此感謝

目 錄

摘要	i
誌謝	ii
第一章 division	v
第二章 更新團隊網站步驟	vi
2.1 詳細步驟說明	vi
第三章 pj3	vii
第四章 code	x
4.1 code_car	x
4.2 code_scoreboard	xii
4.3 code_scorer version	xiv
第五章 image	xix
5.1 image_Quadricycle First Edition	xix
5.2 Quadricycle First Edition_41023230 機器人	xix
5.3 Quadricycle First Edition_41023231 機器人	xix
5.4 Quadricycle First Edition_41023232 機器人	xix
5.5 Quadricycle First Edition_41023233 機器人	xix
5.6 Quadricycle First Edition_41023253 機器人	xix
5.7 image_Quadricycle Second Edition	xix
5.8 Quadricycle Second Edition_41023233 機器人	xix
5.9 Quadricycle Second Edition_41023250 機器人	xix
5.10 image_court	xix
5.11 court_ 球場本體	xix

5.12court_ 球門	xix
5.13court_ 球場球門組合	xix
5.14court_ 組合並加入車子	xix
5.15court_ Independence Stadium	xix
5.16image_scoreboard	xix
5.17image_race	xix
5.18court_ 得分球員板	xix
附錄	xix
作者簡介	xxiv
作者簡介 2	xxv

第一章 division

工作內容:

球場設計:41023253

四輪車設計:41023230 41023231 41023233 41023250 41023253

四輪車程式設計:41023226

記分板設計:41023232

記分板程式設計:41023226

得分球員版設計:41023226

得分球員版程式設計:41023226

pdf ppt 製作:41023226

整理網頁:41023220

第二章 更新團隊網站步驟

2.1 詳細步驟說明

1. 先開啟個人 USB 中的倉儲 ipv6
2. 在至個人 github 中的 fork 倉儲更新成最新版
3. 輸入 git pull 若失敗則有可能 PUTTY 跑掉重設定即可
4. 進行編輯
- 5.acp
6. 在至個人 fork 倉儲 Open pull request
7. 若無法 pull request 那就至個人 fork 在合併最新版本並除錯即可
8. 一路同意推送合併到底
9. 回到整組倉儲確認上傳已完成

第三章 pj3

專案三：接續專案二，各組需對雙輪車進行設計改良，以提升行進與對戰效能，各組需採 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，完成後各組需在分組網站中提供所有相關檔案下載連結，且提供線上分組簡報與分組 pdf 報告連結。

球賽計分系統必須採.ttm 格式建立 (0 99)，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，請另外以建立以機械轉盤傳動計分系統 (mechanical counter)，且採.ttm 格式建立。

協同產品設計規格：

足球規格 (ball)：白色，直徑 0.1m，重量 0.5kg

足球場地 (field)：長 4m x 寬 2.5m

球門規格 (goal[0] and goal[1])：長 0.6m，高 0.3m，寬 0.1m

球員尺寸範圍 (player[0]-player[7])：長 寬 高 各 0.2m，重量 5kg

應交付內容：

專案三場景與多輪車零組件設計（可使用各種 CAD 系統建立，但必須提供完整的檔案下載連結）

專案三控制程式（以 zmqRemoteAPI Python 製作）

專案三開會紀錄與逐字稿（可利用 jit.si 或 OBS 或其他線上開會系統）

專案三各組員任務分配與執行過程影片（可置於 Youtube 或 Onedrive）

專案三網站包括所有協同設計流程所衍生的檔案下載連結，各檔案必須設法壓縮在 30 MB 內並置於網站 downloads 目錄中。

專案三線上簡報檔案

專案三分組報告 pdf 檔案

直接利用 zmqRemoteAPI Python 程式建立場景物件:

```
1 # zmqRemoteApi_IPv6 為將 zmq 通訊協定修改為 IPv4 與 IPv6 相容
2 from zmqRemoteApi_IPv6 import RemoteAPIClient
3 import time
4 import math
5 import keyboard
6
7 # 利用 zmqRemoteAPI 以 23000 對場景伺服器進行連線
8 client = RemoteAPIClient('localhost', 23000)
9 # 以 getObject 方法取得場景物件
10 sim = client.getObject('sim')
11 box = sim.getObject('/box')
12
13 # 啟動模擬
14 sim.startSimulation()
15 # 建立尺寸數列，分別定義 x, y, z 方向尺寸
16 x = 0.2
17 y = 0.2
18 z = 0.1
19 size = [x, y, z]
20
21 # 利用 size 數列，建立圓柱物件，2 代表 cylinder
22 # 8 表示 respondable, 1 為質量
23 digit1_handle = sim.createPureShape(2, 8, size, 1, None)
24 # 將圓柱物件命名為 digit1，若用於機械計分可做為個位數轉盤
25 # 之後可再導入帶有數字組立的外型零件
26 sim.setObjectAlias(digit1_handle, 'digit1')
27 # 轉角單位為程度
28 sim.setObjectOrientation(digit1_handle, -1, [0, math.pi/2, 0])
29 # 起始物件中心位於 [0, 0, 0]，為了位於地板，往 z 提升一個半徑高度
30 sim.setObjectPosition(digit1_handle, -1, [0, 0, x/2])
31
32 # 建立 revolute joint 命名為 joint，且將 joint mode 設為 dynamic, control mode
33 # 設為 velocity
34 joint1_handle = sim.createJoint(sim.joint_revolute_subtype, sim.
            jointmode_dynamic, 0, None)
35 sim.setObjectInt32Param(joint1_handle, sim.jointintparam_dynctrlmode, sim.
            jointdynctrl_velocity)
```

```
35 sim.setObjectAlias(joint1_handle, 'joint1')
36
37 # 取得 cylinder 的位置座標
38 digit1_pos = sim.getObjectPosition(digit1_handle, -1)
39 joint1_pos = [digit1_pos[0], digit1_pos[1], digit1_pos[2]]
40
41 # 將 joint1 紮於 cylinder 中心
42 sim.setObjectPosition(joint1_handle, -1, joint1_pos)
43 # 取得 digit1_handle 的方位
44 digit1_ori = sim.getObjectOrientation(digit1_handle, -1)
45 # 將 joint1_handle 方位與 digit1 對齊
46 sim.setObjectOrientation(joint1_handle, -1, digit1_ori)
47
48 # 將 joint1 置於 box 上
49 sim.setObjectParent(joint1_handle, box, True)
50 # 將 cylinder 置於 joint1 上
51 sim.setObjectParent(digit1_handle, joint1_handle, True)
52
53 # 鎖定 joint1
54 sim.setJointForce(joint1_handle, float('inf'))
55
56 print("基本場景建立完成!")
57
58 # 設定主迴圈
59 while True:
60     # 設定 joint1 目標速度
61     sim.setJointTargetVelocity(joint1_handle, 10)
62     # 讓 coppeliasim 有時間按照設定讓 joint1 旋轉
63     time.sleep(0.01)
64
65     if keyboard.is_pressed('q'):
66         # 可以按下 q 鍵跳出重複執行迴圈
67         break
68
69 # 終止模擬
70 #sim.stopSimulation()
```

第四章 code

4.1 code_car

第一版程式，使用兩邊的速度差來控制方向，且可以在前後的同時控制方向

```
1 from zmqRemoteApi_IPv6 import RemoteAPIClient
2 import keyboard
3
4 client = RemoteAPIClient('localhost', 23000)
5
6 print('Program started')
7 sim = client.getObject('sim')
8 sim.startSimulation()
9 print('Simulation started')
10
11 def setWheelMotion(leftSpeed, rightSpeed):
12     # Set target velocity for each wheel
13     frontLeftWheel = sim.getObject('/frontLeftJoint')
14     frontRightWheel = sim.getObject('/frontRightJoint')
15     rearLeftWheel = sim.getObject('/rearLeftJoint')
16     rearRightWheel = sim.getObject('/rearRightJoint')
17     sim.setJointTargetVelocity(frontLeftWheel, leftSpeed)
18     sim.setJointTargetVelocity(frontRightWheel, rightSpeed)
19     sim.setJointTargetVelocity(rearLeftWheel, leftSpeed)
20     sim.setJointTargetVelocity(rearRightWheel, rightSpeed)
21
22 # Initialize motion variables
23 leftSpeed = 0
24 rightSpeed = 0
25
26 # Main loop
27 while True:
28     # Check keyboard input
29     if keyboard.is_pressed('w'):
30         leftSpeed = -10    # Forward motion
31         rightSpeed = -10   # Forward motion
32     elif keyboard.is_pressed('s'):
33         leftSpeed = 10    # Backward motion
34         rightSpeed = 10    # Backward motion
```

```

35     else:
36         leftSpeed = 0
37         rightSpeed = 0
38
39     if keyboard.is_pressed('a'):
40         leftSpeed += 5 # Left turn
41         rightSpeed -= 5 # Left turn
42     elif keyboard.is_pressed('d'):
43         leftSpeed -= 5 # Right turn
44         rightSpeed += 5 # Right turn
45
46     if keyboard.is_pressed('q'):
47         break # Quit
48
49     # Set motion for all wheels
50     setWheelMotion(leftSpeed, rightSpeed)
51
52 # Stop the simulation
53 sim.stopSimulation()

```

第二版程式，前輪各多加一個joint，使其運動更加合理

```

1 from zmqRemoteApi_IPv6 import RemoteAPIClient
2 import keyboard
3
4 client = RemoteAPIClient('localhost', 23000)
5
6 sim = client.getObject('sim')
7
8 sim.startSimulation()
9
10 frontLeftSteeringJoint = sim.getObject('/frontLeftJoint1')
11 frontRightSteeringJoint = sim.getObject('/frontRightJoint1')
12 frontLeftWheel = sim.getObject('/frontLeftJoint2')
13 frontRightWheel = sim.getObject('/frontRightJoint2')
14 rearLeftWheel = sim.getObject('/rearLeftJoint')
15 rearRightWheel = sim.getObject('/rearRightJoint')
16
17 def setFrontWheelSteeringAngle(steeringAngle):
18     sim.setJointTargetPosition(frontLeftSteeringJoint, steeringAngle)
19     sim.setJointTargetPosition(frontRightSteeringJoint, steeringAngle)
20

```

```

21 def setAllWheelSpeed(speed):
22     sim.setJointTargetVelocity(frontLeftWheel, speed)
23     sim.setJointTargetVelocity(frontRightWheel, speed)
24     sim.setJointTargetVelocity(rearLeftWheel, speed)
25     sim.setJointTargetVelocity(rearRightWheel, speed)
26
27 steeringAngle = 0
28 speed = 0
29
30 SPEED_FORWARD = -20
31 SPEED_BACKWARD = 20
32 STEERING_ANGLE_LEFT = 0.3
33 STEERING_ANGLE_RIGHT = -0.3
34
35 while True:
36     if keyboard.is_pressed('w'):
37         speed = SPEED_FORWARD
38     elif keyboard.is_pressed('s'):
39         speed = SPEED_BACKWARD
40     else:
41         speed = 0
42
43     if keyboard.is_pressed('a'):
44         steeringAngle = STEERING_ANGLE_LEFT
45     elif keyboard.is_pressed('d'):
46         steeringAngle = STEERING_ANGLE_RIGHT
47     else:
48         steeringAngle = 0
49
50     if keyboard.is_pressed('q'):
51         break
52
53     setFrontWheelSteeringAngle(steeringAngle)
54
55     setAllWheelSpeed(speed)
56
57 sim.stopSimulation()

```

4.2 code_scoreboard

```

1 function sysCall_init()
2     score = 0
3     wheelJoint = sim.getObjectHandle('/joint1g')

```

```

4     robot = {
5         sim.getObjectHandle('/car1'),
6         sim.getObjectHandle('/car2'),
7         sim.getObjectHandle('/car3'),
8         sim.getObjectHandle('/car4'),
9         sim.getObjectHandle('/car5'),
10        sim.getObjectHandle('/car6'),
11        sim.getObjectHandle('/car7'),
12        sim.getObjectHandle('/car8')
13    }
14
15    initialPos = {
16        {-1.050, -0.77134, 0.21},
17        {-1.050, -0.27134, 0.21},
18        {-1.050, 0.22867, 0.21},
19        {-1.050, 0.62866, 0.21},
20        {1.175, -0.77134, 0.21},
21        {1.175, -0.27134, 0.21},
22        {1.175, 0.22867, 0.21},
23        {1.175, 0.62866, 0.21}
24    }
25
26    initialOri = {
27        {0, 90, 0},
28        {0, 90, 0},
29        {0, 90, 0},
30        {0, 90, 0},
31        {0, -90, 180},
32        {0, -90, 180},
33        {0, -90, 180},
34        {0, -90, 180}
35    }
36 end
37
38 sensor = sim.getObject('./sensor')
39 initialPosBall = sim.getObjectPosition(sensor, -1)
40 ball = sim.getObject('/ball')
41
42 function sysCall_actuation()
43     result = sim.readProximitySensor(sensor)
44     if (result > 0) then
45         score = score + 1
46         sim.setObjectPosition(ball, -1, {0,0,0.25})

```

```

46     -- Rotate the wheel joint by 36 degrees
47     local currentAngle = sim.getJointPosition(wheelJoint)
48     local targetAngle = currentAngle + math.rad(-36)
49     sim.setJointTargetPosition(wheelJoint, targetAngle)
50
51     for i = 1, 8 do
52         sim.setObjectPosition(robot[i], -1, initialPos[i])
53         sim.setObjectOrientation(robot[i], -1, initialOri[i])
54     end
55 end
56 end

```

這個程式控制了球進入球門後，記分板轉動 36 度，以及車子與球回到初始位置，目前發生未知原因使的 5 到 8 的車子在球進入球門後，位置與方向與我所設置的不相符

4.3 code_scorer version

將進球後顯示 goal 加入記分板程式中

```

1 function sysCall_init()
2     score = 0
3     wheelJoint = sim.getObjectHandle('/joint1g')
4     robot = {
5         sim.getObjectHandle('/car1'),
6         sim.getObjectHandle('/car2'),
7         sim.getObjectHandle('/car3'),
8         sim.getObjectHandle('/car4'),
9         sim.getObjectHandle('/car5'),
10        sim.getObjectHandle('/car6'),
11        sim.getObjectHandle('/car7'),
12        sim.getObjectHandle('/car8')
13    }
14    initialPos = {
15        {-1.050, -0.77134, 0.21},
16        {-1.050, -0.27134, 0.21},
17        {-1.050, 0.22867, 0.21},
18        {-1.050, 0.62866, 0.21},
19        {1.175, -0.77134, 0.21},
20        {1.175, -0.27134, 0.21},
21        {1.175, 0.22867, 0.21},
22        {1.175, 0.62866, 0.21}
23    }

```

```

24     initialOri = {
25         {0, 90, 0},
26         {0, 90, 0},
27         {0, 90, 0},
28         {0, 90, 0},
29         {0, -90, 180},
30         {0, -90, 180},
31         {0, -90, 180},
32         {0, -90, 180}
33     }
34
35     displayHandles = {
36         sim.getObjectHandle('/display1'),
37         sim.getObjectHandle('/display2'),
38         sim.getObjectHandle('/display3'),
39         sim.getObjectHandle('/display4'),
40         sim.getObjectHandle('/display5'),
41         sim.getObjectHandle('/display6'),
42         sim.getObjectHandle('/display7')
43     }
44
45     displayParameters = {
46         {0, 0, 1, 0, 0, 1, 0},
47         {1, 0, 1, 1, 1, 0, 1},
48         {1, 0, 1, 1, 0, 1, 1},
49         {0, 1, 1, 1, 0, 1, 0},
50         {1, 1, 0, 1, 0, 1, 1},
51         {1, 1, 0, 1, 1, 1, 1},
52         {1, 0, 1, 0, 0, 1, 0},
53         {1, 1, 1, 1, 1, 1, 1}
54     }
55
56     goalEffectEnabled = false
57     lastGoalPlayer = nil
58     flashingState = false
59     flashingTimer = 0
60     goalHandle = sim.getObjectHandle('/goal')
61     flashingDuration = 0.5
62     flashingTimer = 0
63     isFlashing = false
64
65 end
66
67 sensor = sim.getObject('./sensor')
68 initialPosBall = sim.getObjectPosition(sensor, -1)
69 ball = sim.getObject('/ball')

```

```

66
67 function sysCall_actuation()
68     result = sim.readProximitySensor(sensor)
69     if (result > 0) then
70         score = score + 1
71         sim.setObjectPosition(ball, -1, {0,0,0.25})
72
73         -- Rotate the wheel joint by 36 degrees
74         local currentAngle = sim.getJointPosition(wheelJoint)
75         local targetAngle = currentAngle + math.rad(-36)
76         sim.setJointTargetPosition(wheelJoint, targetAngle)
77
78         for i = 1, 8 do
79             sim.setObjectPosition(robot[i], -1, initialPos[i])
80             sim.setObjectOrientation(robot[i], -1, initialOri[i])
81         end
82
83         goalEffectEnabled = true
84         lastGoalPlayer = result
85         flashingState = false
86         flashingTimer = 0
87     end
88
89     if goalEffectEnabled then
90         if not isFlashing then
91             flashingTimer = flashingTimer + sim.getSimulationTimeStep()
92             if flashingTimer >= flashingDuration then
93                 isFlashing = true
94                 sim.setShapeColor(goalHandle, nil, sim.
95                                 colorcomponent_ambient_diffuse, {1, 0.9, 0})
96             end
97             elseif isFlashing then
98                 flashingTimer = flashingTimer + sim.getSimulationTimeStep()
99                 if flashingTimer >= flashingDuration + 3 then
100                     goalEffectEnabled = false
101                     isFlashing = false
102                     sim.setShapeColor(goalHandle, nil, sim.
103                                     colorcomponent_ambient_diffuse, {0, 0, 0})
104                 end
105             end
106         end
107     end

```

```

106
107 function sysCall_cleanup()
108     sim.setShapeColor(goalHandle, nil, sim.colorcomponent_ambient_diffuse,{0,
109         0, 0})
110 end

```

檢測與球碰撞的球員並顯示其背號在七段顯示器中

```

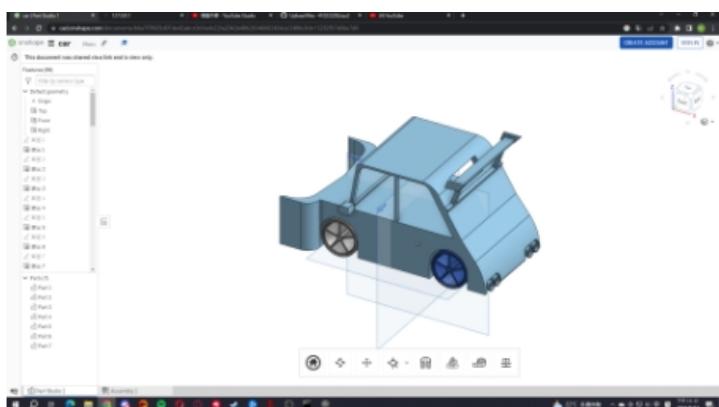
1 local robotHandles = {}
2 local ballHandle = -1
3 local displayHandles = {}
4 local displayParameters = {
5     {0, 0, 1, 0, 0, 1, 0},
6     {1, 0, 1, 1, 1, 0, 1},
7     {1, 0, 1, 1, 0, 1, 1},
8     {0, 1, 1, 1, 0, 1, 0},
9     {1, 1, 0, 1, 0, 1, 1},
10    {1, 1, 0, 1, 1, 1, 1},
11    {1, 0, 1, 0, 0, 1, 0},
12    {1, 1, 1, 1, 1, 1, 1}
13 }
14
15 function sysCall_init()
16     for i = 1, 8 do
17         robotHandles[i] = sim.getObjectHandle('car' .. i)
18     end
19     ballHandle = sim.getObjectHandle('ball')
20
21     for i = 1, 7 do
22         displayHandles[i] = sim.getObjectHandle('/display' .. i)
23     end
24 end
25
26 function sysCall_actuation()
27     for i = 1, 8 do
28         local collision = sim.checkCollision(robotHandles[i], ballHandle)
29         if collision ~= 0 then
30             setDisplayDigit(i, i <= 4)
31             break
32         end
33     end
34 end

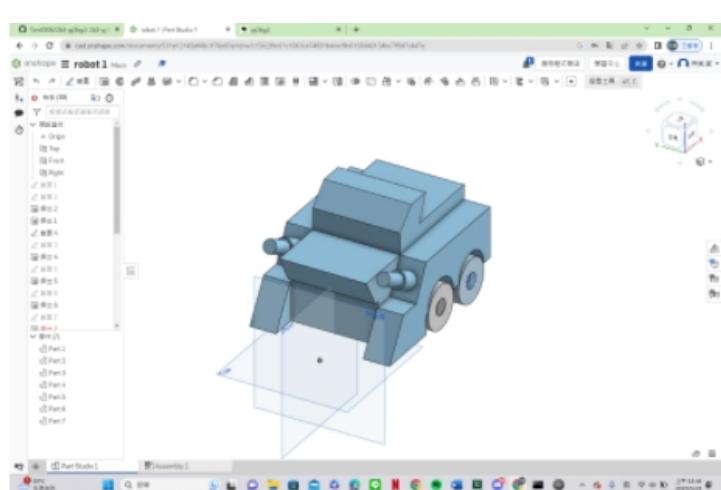
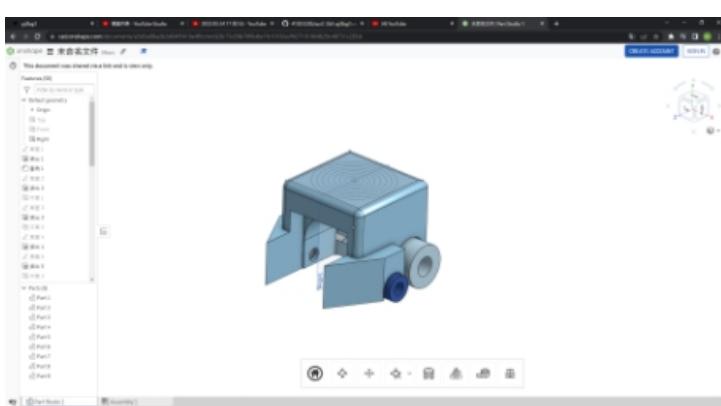
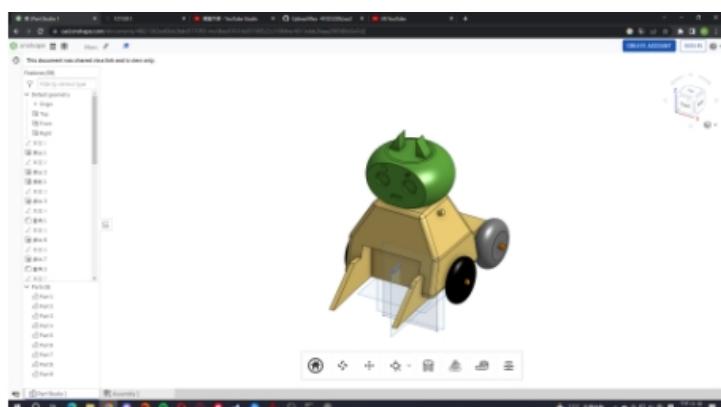
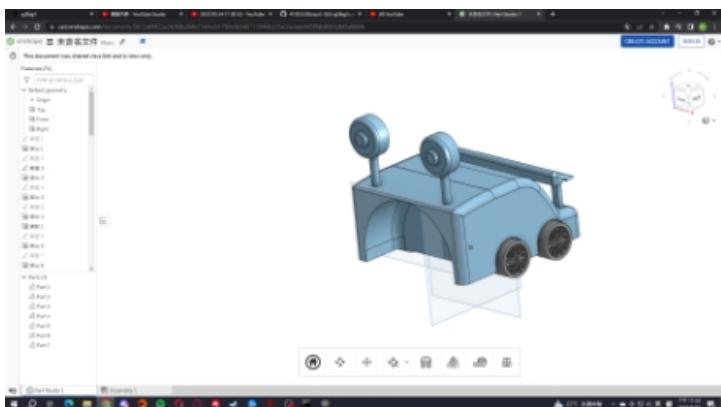
```

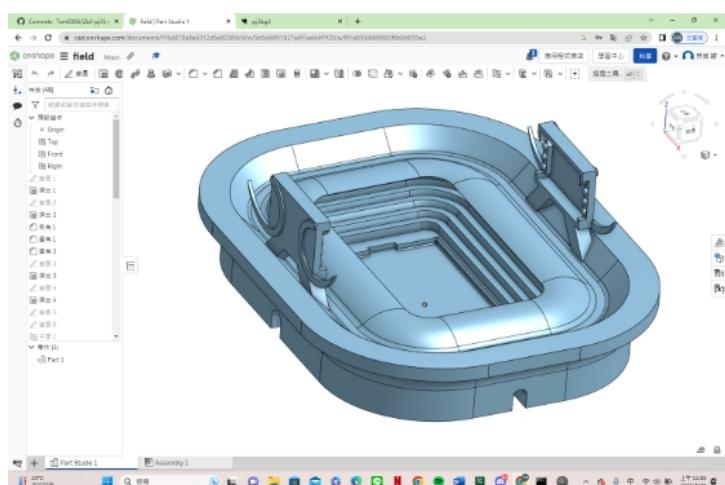
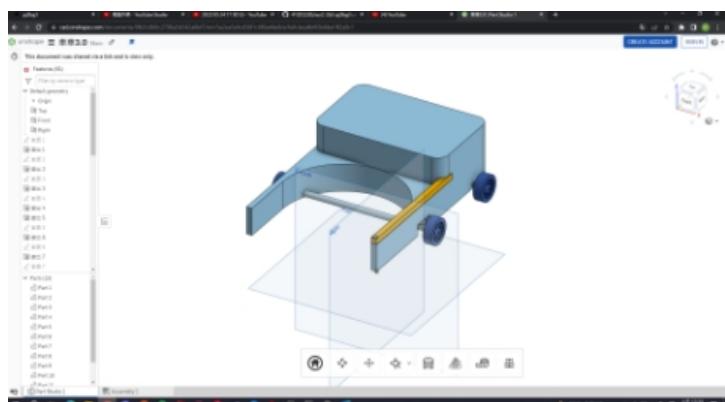
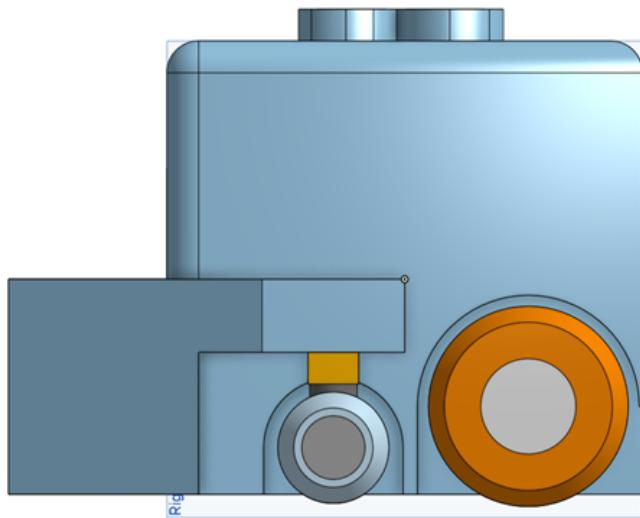
```
35
36 function setDisplayDigit(digit, isRed)
37     for i = 1, 7 do
38         if displayParameters[digit][i] == 1 then
39             if isRed then
40                 sim.setShapeColor(displayHandles[i], nil, sim.
41                                 colorcomponent_ambient_diffuse, {1, 0, 0})
42             else
43                 sim.setShapeColor(displayHandles[i], nil, sim.
44                                 colorcomponent_ambient_diffuse, {0, 1, 0})
45             end
46         else
47             sim.setShapeColor(displayHandles[i], nil, sim.
48                                 colorcomponent_ambient_diffuse, {0, 0, 0})
49         end
50     end
51 end
```

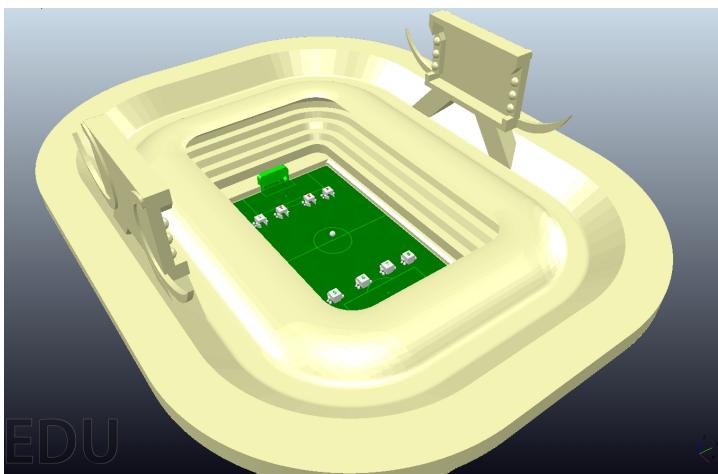
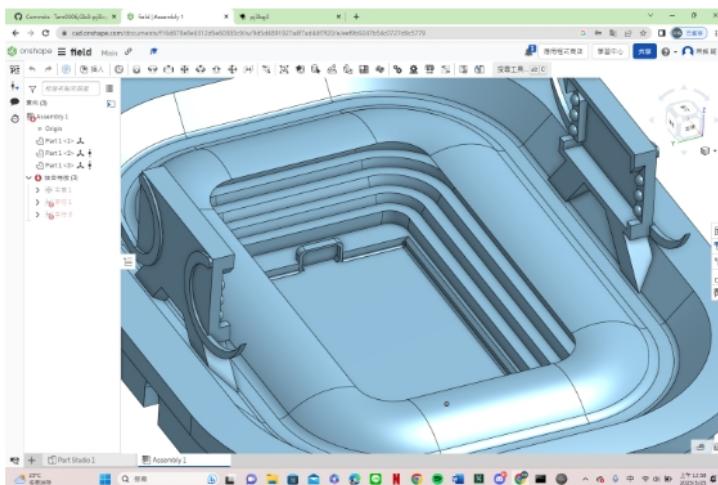
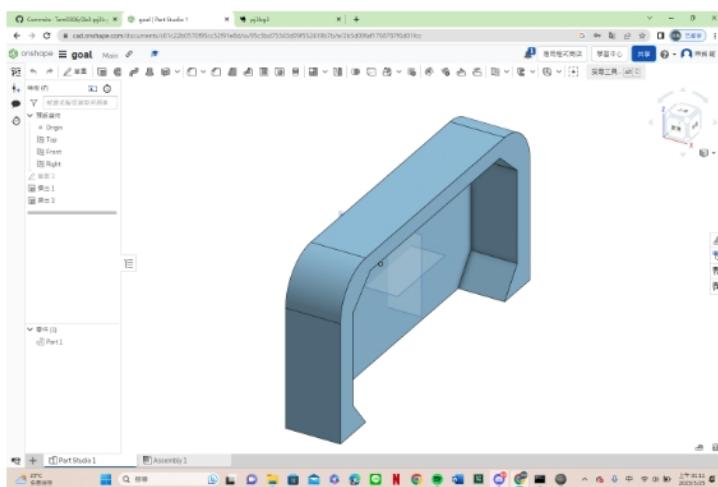
第五章 image

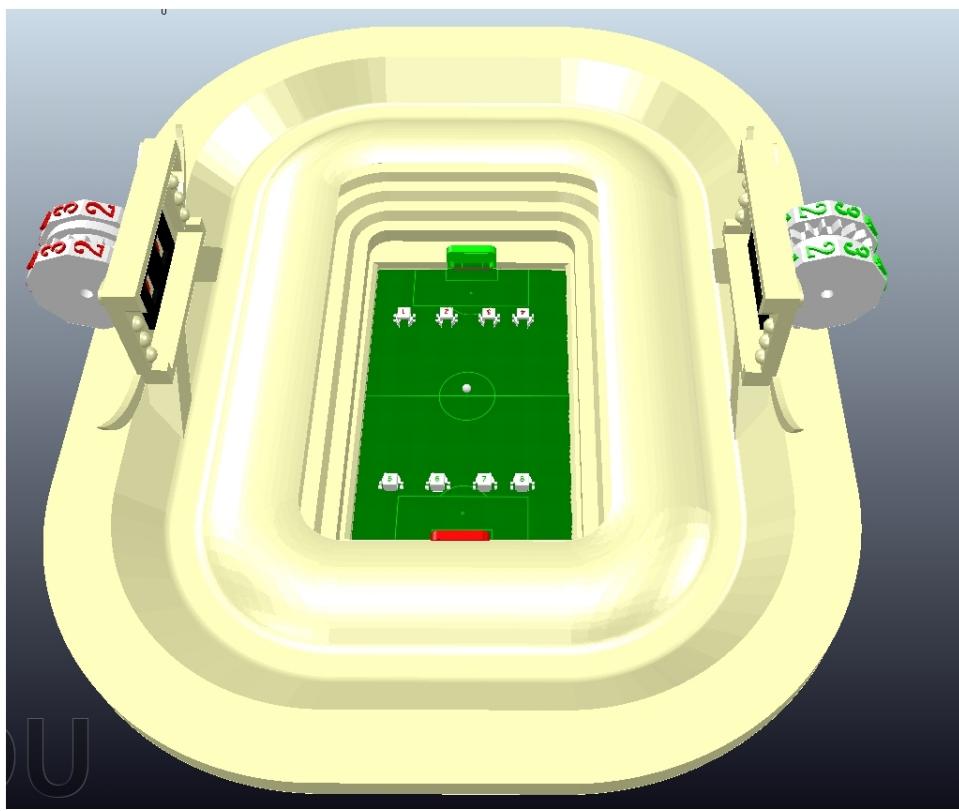
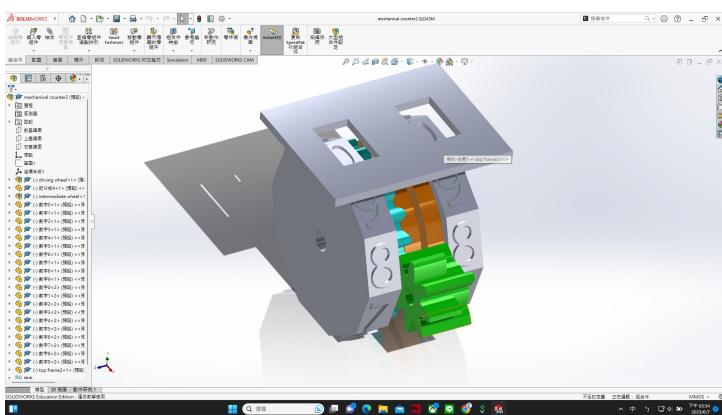
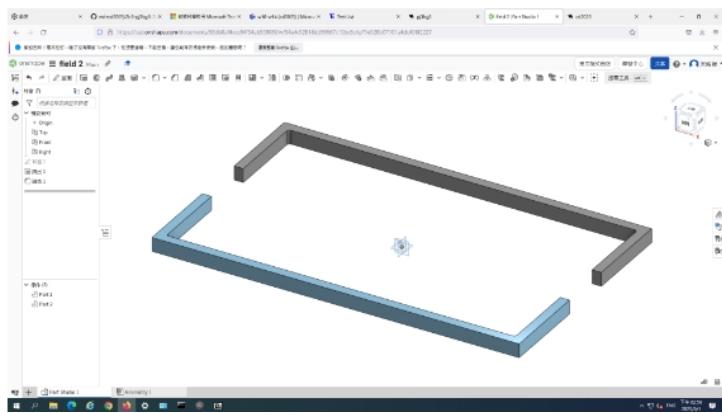
- 5.1 image_Quadricycle First Edition
- 5.2 Quadricycle First Edition_41023230 機器人
- 5.3 Quadricycle First Edition_41023231 機器人
- 5.4 Quadricycle First Edition_41023232 機器人
- 5.5 Quadricycle First Edition_41023233 機器人
- 5.6 Quadricycle First Edition_41023253 機器人
- 5.7 image_Quadricycle Second Edition
- 5.8 Quadricycle Second Edition_41023233 機器人
- 5.9 Quadricycle Second Edition_41023250 機器人
- 5.10 image_court
- 5.11 court_ 球場本體
- 5.12 court_ 球門
- 5.13 court_ 球場球門組合
- 5.14 court_ 組合並加入車子
- 5.15 court_ Independence Stadium
- 5.16 image_scoreboard
- 5.17 image_race
- 5.18 court_ 得分球員板

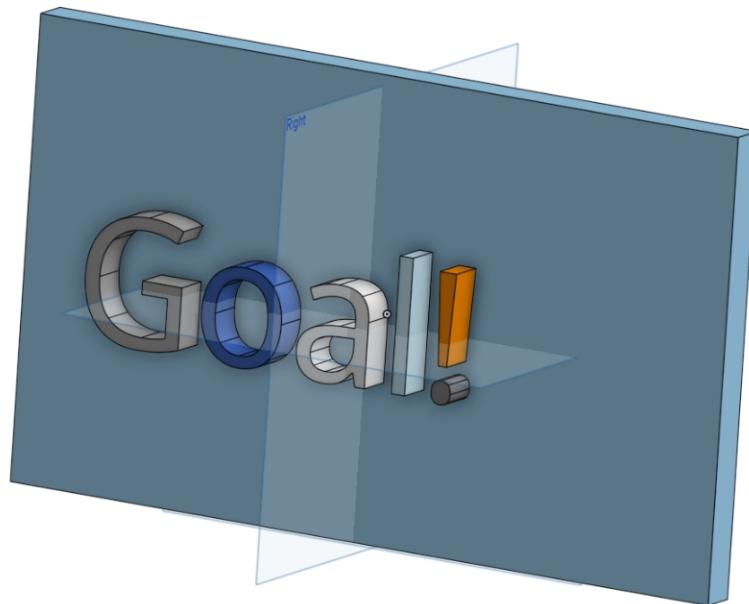












作者簡介



姓名 : 陳冠
學號 : 41023220
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :



姓名 : 陳建霖
學號 : 41023226
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :



姓名 : 彭聖宗
學號 : 41023230
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :

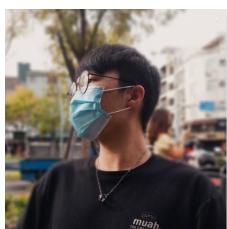


姓名 : 湛有杰
學號 : 41023231
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :

作者簡介 2



姓名 : 黃文彥
學號 : 41023233
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :



姓名 : 蔡叡得
學號 : 41023250
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :



姓名 : 謝宗銘
學號 : 41023253
就讀學校 : 國立虎尾科技大學
機械設計工程系
經歷 :

【13】

分
類
編
號

.. pj3bg3 網際足球場景設計

一一二級