

協同產品設計實習

Webot 模擬系統之投籃遊戲

姓 名：王樟皓

學 號：41023114

指導教授：嚴家銘 教授

1.基本資訊

在老師的範本中，是將 sensor 以圓球形置於球框，並且 radius 設 0.2(圖 1)

接著是 sensor 的很重要的一環，lookup table 是用來告訴 Webots 模擬器：感測器讀到的值要怎麼對應成實際回傳的數值。

如圖 3

- 1.距離是 0 公尺時，感測器回傳 1000，有 1%雜訊
 - 2.距離是 0.12 公尺時，感測器回傳 620，有 1%雜訊
- 依此類推

下方(圖 2)的 type 主要是對紅色物體敏感、不會畫出紅點、會忽略透明物體

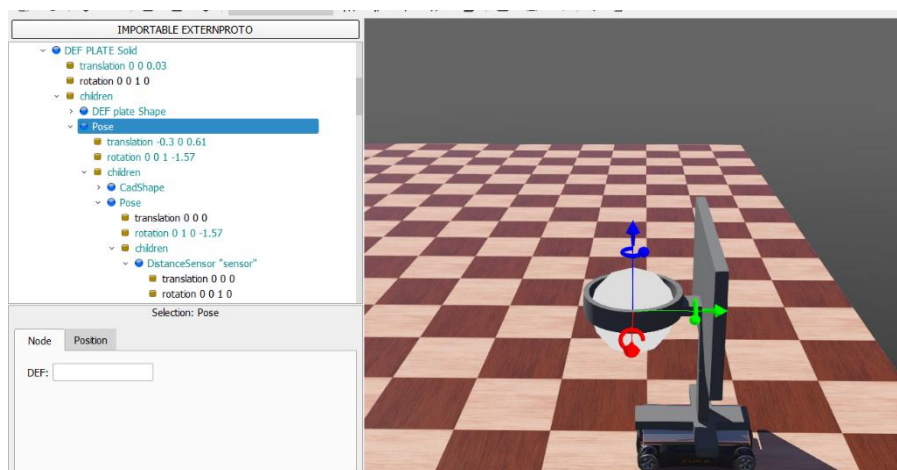


圖 1

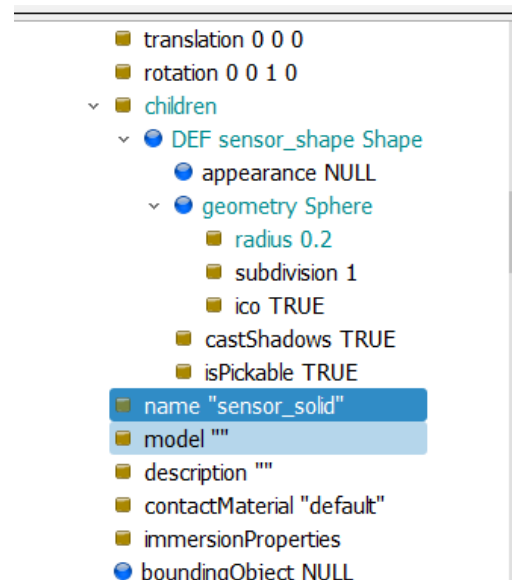


圖 2

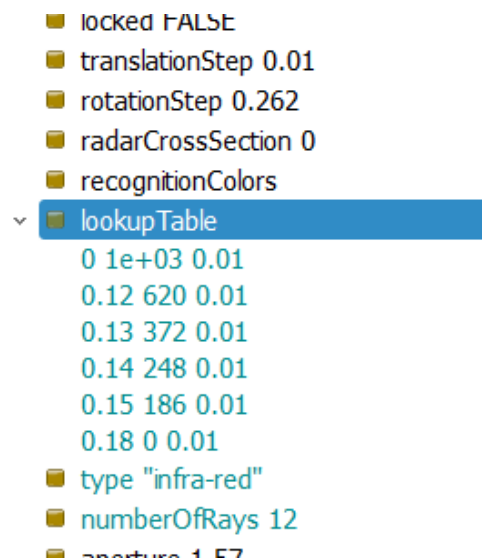


圖 3

Center of Mass 是質心，如果質心太高或偏一邊，在移動或受到力量時，就更容易翻倒。

Inertia Matrix 慣性矩陣簡單解釋：

它是一個 3×3 的矩陣（數學上叫「張量」），裡面包含了物體繞 x 、 y 、 z 三個方向旋轉時的轉動慣量。

如果你是圓盤或輪子那種簡單形狀，就只要一個值；但對於複雜 3D 形狀，就需要整個矩陣來描述。

$$\begin{vmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{vmatrix}$$

其中：

I_{xx} 是繞 x 軸的轉動慣量

I_{yy} 是繞 y 軸的

I_{zz} 是繞 z 軸的

而這數值投球機跟球框是一樣的，因為他們用的 youbot 是一樣的，輪子就是一樣的

如圖 4

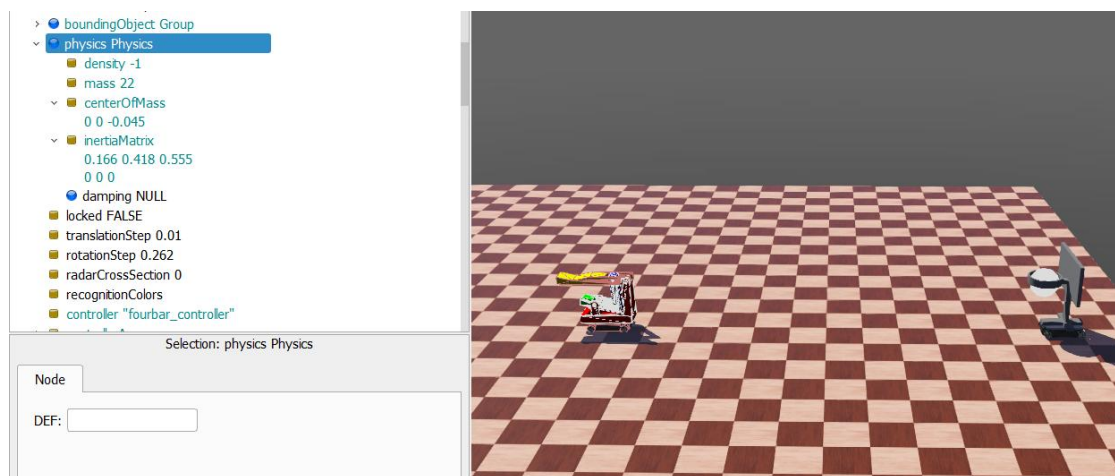


圖 4

在 worldinfo 中設置了兩個 ContactProperties，其中改動的參數有(圖 5)

coulombFriction：滑動摩擦係數

frictionRotation：旋轉摩擦（抗扭轉）

forceDependentSlip：依據力量產生的滑動

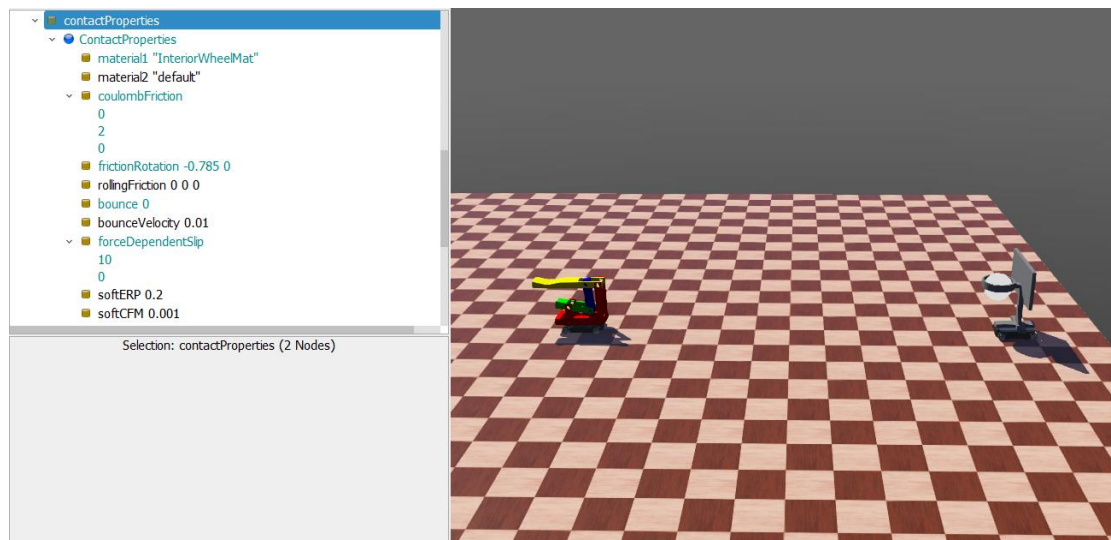


圖 5

2.排錯問題紀錄

1.關於 youbot_stand 到達第一個座標時會卡住的問題，後續找到是因為 InteriorWheel.prote 與 ExteriorWheel.proto 是 webot 本身預設的，雖然他是.proto 檔，但她前身也是一個 robot+hingejoint 群組，將這之下的 motor 的 maxVelocity 預設值 14.8 更改為 100 即可

```
HingeJoint {  
  
    device [  
  
        RotationalMotor {  
  
            name IS name  
  
            maxVelocity 100  
  
        }  
    ]  
}
```

2.在加入 supervisor 跟 score_board 後，遇到的第一個問題為無法使用 A 鍵生成球，原因是 webot 下的 python 沒有安裝 numpy，只要在 webot 中的 Python command 確認 python 的位址，隨後在使用 python -m pip install numpy 就可以了。

3.接著在 supervisor 下的 feed_ball 又出現問題，由於我自行建立的 youbot 一開始是直接使用 robot 群組下的 name 做更動，但 feed_ball 對於生成球的座標是根據 youbot 改變的，而改變是根據 DEF ... Robot，因此儘管不會顯示錯誤，但仍然無法生成球，因此只要在 feed_ball 下更改我的 DEFname 以及在.wbt 中給 robot 群組加 DEFname 就能做動，內容如下

```

def youbot_local_to_world(local_pos):
    youbot_node = supervisor.getFromDef('youBot_shooter')
    if youbot_node is None:
        raise RuntimeError("找不到 DEF 為 youBot_shooter 的 Robot 物件")
    youbot_translation = np.array(youbot_node.getField('translation').getSFVec3f())
    youbot_rotation = youbot_node.getField('rotation').getSFRotation()
    youbot_axis = youbot_rotation[:3]
    youbot_angle = youbot_rotation[3]
    youbot_rot_mat = axis_angle_to_rotation_matrix(youbot_axis, youbot_angle)
    rotated = youbot_rot_mat @ np.array(local_pos)
    world_pos = youbot_translation + rotated
    return tuple(world_pos)

```

以上設定中就是這一段使生成球的的座標與 shooter 斷開了

```

youbot_node = supervisor.getFromDef('youBot_shooter')
if youbot_node is None:
    raise RuntimeError("找不到 DEF 為 youBot_shooter 的 Robot 物件")

```

只要給予的 DEFname 相符就能做動

4. 更改 ThreeDigitSevenSegment.proto 的放置情形，使用 score_board supervisor 的 controller 時似乎會無法讀取.proto 的內容，儘管.proto 內容與在.wbt 中使用 Transform 節點的擺設情形是一致的，但這似乎是設定，需要更改成在.wbt 中放置七段顯示器不放置.proto，或是從 controll 下手更改

5. 更改完七段顯示器計分後，隨之而來的問題是無法使用 supervisor 偵測計分，檢查後發現計分上需要用的連動有 emitter 及 receiver，並且他們的 channel 必須一致才能連動，檢測完後開始投籃發現一直無法計分，但似乎是 lookup table 出問題，將他更改並改回來後再把 youbot_shooter 的起始距離往後移動一點即可

而本次內容是更改為 shooter 和 stand 都由玩家操控，使用 B 生成球 J 打擊 K 收回

6. 在自行製作.proto 檔案時發現 proto 檔案無法直接生成，必須得自行建立，尤其是得將"]{" 這兩個括號給分出來分清楚，而以下就是自行建立.proto 檔案的起手內容

之後導入自己的 proto-robot 後 fourbar_controller 無法支援 proto 檔案，根本原因為 proto 檔案會省略掉太多子節點，儘管在 Scite 中檢視也都沒有錯誤，但還是會無法讀取到 motor，接著發現不只因為 proto 檔案無法讀取 motor，導入後將 proto 轉換為 bass node 後也無法讀取，原因是主要 robot 為 youbot 群組，若在下面再加上 shooter 的 robot 群組會發生報錯，所以若想將 shooter 與 youbot 連動，那麼 shooter 必須使用 solid 群組

以下為.proto 檔案建置起手

```
#VRML_SIM R2023b utf8
```

```
PROTO shooter1 [  
  # 場景控制  
  field SFVec3f    translation 0 0 0  
  field SFRotation rotation    0 0 1 0  
] {  
  Robot {
```

(1) Youbot stand node tree

```
DEF youBot_stand Robot {
  translation 6.23 -0.12 0.103
  children [
    BodyMesh {
    }
    DEF WHEEL5 InteriorWheel {
      translation 0.228 -0.158 -0.055
      anchor 0.228 -0.158 -0.055
      name "wheel5"
      sensorName "wheel1sensor"
    }
    DEF WHEEL6 ExteriorWheel {
      translation 0.228 0.158 -0.055
      anchor 0.228 0.158 -0.055
      name "wheel6"
      sensorName "wheel2sensor"
    }
    DEF WHEEL7 ExteriorWheel {
      translation -0.228 -0.158 -0.055
      anchor -0.228 -0.158 -0.055
      name "wheel7"
      sensorName "wheel3sensor"
    }
    DEF WHEEL8 InteriorWheel {
      translation -0.228 0.158 -0.055
      anchor -0.228 0.158 -0.055
      name "wheel8"
      sensorName "wheel4sensor"
    }
    Solid {
      translation 0 0 0.03
      children [
        Pose {
          translation -0.3 0 0.61
          rotation 0 0 1 -1.57
          children [
            CadShape {
```



```

        url [
            "../cad/split_parts/basket_stand_small.obj"
        ]
    }
    Pose {
        children [
            DistanceSensor {
                children [
                    DEF sensor_solid Solid {
                        children [
                            DEF sensor Shape {
                                geometry Sphere {
                                    radius 0.2
                                }
                            }
                        ]
                        name "sensor_solid"
                    }
                ]
                name "sensor"
                lookupTable [
                    0 1000 0.01
                    0.12 620 0.01
                    0.13 372 0.01
                    0.14 248 0.01
                    0.15 186 0.01
                    0.18 0 0.01
                ]
                type "infra-red"
                numberOfRays 12
            }
        ]
    }
}

DEF PLATE Shape {
    appearance PBRAppearance {
        baseColor 0.75 0.75 0.75
    }
}

```

```

        }
        geometry Box {
            size 0.5 0.3 0.02
        }
    }
]
name "PLATE"
boundingObject USE PLATE
physics Physics {
    mass 0.5
}
}
GPS {
}
InertialUnit {
}
Emitter {
    name "score_emitter"
    channel 1
}
]
name "youBot_stand"
model "KUKA youBot"
description "KUKA youBot - Base with wheels only"
boundingObject Group {
    children [
        Pose {
            translation 0 0 -0.045
            children [
                Box {
                    size 0.34 0.34 0.09
                }
            ]
        }
    ]
    Pose {
        translation 0 0 -0.045
        children [
            Box {

```

```
        size 0.56 0.23 0.09
    }
]
}
]
}
physics Physics {
    density -1
    mass 22
    centerOfMass [
        0 0 -0.045
    ]
    inertiaMatrix [
        0.166204 0.418086 0.55459
        0 0 0
    ]
}
controller "stand_controller"
}
```

(2) counter_supervisor node tree

```
DEF counter_supervisor Robot {  
  children [  
    Receiver {  
      name "score_receiver"  
      channel 1  
    }  
  ]  
  name "counter_supervisor"  
  controller "counter_supervisor"  
  supervisor TRUE
```

以上**(1)(2)**是關於第五點提到的 **emitter** 及 **receiver** 他們在.wbt 檔案中所在的位置，並且他們的 **channel** 必須一致才能連動。

(3) stand controller

```
from controller import Robot, Keyboard

# Constants
#TIME_STEP = 32 # Simulation time step in milliseconds
WHEEL_RADIUS = 0.1 # Radius of the wheels in meters (10cm)
L = 0.471 # Half of the robot's length in meters
W = 0.376 # Half of the robot's width in meters
MAX_VELOCITY = 10.0 # Maximum velocity allowed for the wheels

# Initialize the robot
robot = Robot()

# Get simulation time step
timestep = int(robot.getBasicTimeStep())
emitter = robot.getDevice("score_emitter")
score_to_send = 2

# Get the DistanceSensor device
sensor = robot.getDevice('sensor')
sensor.enable(timestep)
score = 0
last_score_time = 0
cooldown = 1.0

# Initialize the keyboard
keyboard = Keyboard()
#keyboard.enable(TIME_STEP)
keyboard.enable(timestep)

# Get motor devices
wheel5 = robot.getDevice("wheel5") # Front-right wheel
wheel6 = robot.getDevice("wheel6") # Front-left wheel
wheel7 = robot.getDevice("wheel7") # Rear-right wheel
wheel8 = robot.getDevice("wheel8") # Rear-left wheel

# Set motors to velocity control mode
```

```

for wheel in [wheel5, wheel6, wheel7, wheel8]:
    wheel.setPosition(float('inf')) # Enable velocity control
    wheel.setVelocity(0) # Set initial velocity to 0

def set_wheel_velocity(v1, v2, v3, v4):
    """Set the velocity of all wheels."""
    wheel5.setVelocity(v1)
    wheel6.setVelocity(v2)
    wheel7.setVelocity(v3)
    wheel8.setVelocity(v4)

# lookupTable 轉成程式用的格式
lookup_table = [
    (1000, 0.00),
    (620, 0.12),
    (372, 0.13),
    (248, 0.14),
    (186, 0.15),
    (0, 0.18)
]

def ad_to_distance(ad_value):
    # 假設AD值遞減，距離遞增
    for i in range(len(lookup_table)-1):
        a0, d0 = lookup_table[i]
        a1, d1 = lookup_table[i+1]
        if a1 <= ad_value <= a0:
            # 線性插值
            return d0 + (d1 - d0) * (ad_value - a0) / (a1 - a0)
    # 超出範圍時回傳極值
    if ad_value > lookup_table[0][0]:
        return lookup_table[0][1]
    return lookup_table[-1][1]

# Main loop
print("Use 'W', 'A', 'S', 'D' keys to control the robot.")
print("W: Move forward, S: Move backward, A: Turn left, D: Turn
right.")

```

```

print("Press 'Q' to quit.")

#while robot.step(TIME_STEP) != -1:
while robot.step(timestep) != -1:

    key = keyboard.getKey() # Read the key pressed
    # Read DistanceSensor value
    sensor_value = sensor.getValue()
    #print(sensor_value)
    distance = ad_to_distance(sensor_value)
    current_time = robot.getTime()
    #print(sensor_value)
    # Check if the ball blocks the sensor (you may need to adjust the
threshold based on your sensor's range)
    if key == ord('J') or key == ord('j'):
        print(distance)
    if key == ord('K') or key == ord('k'):
        print(distance)

    if distance < 0.11 and (current_time - last_score_time) >
cooldown:
        score +=2
        print("得分")
        print(distance)
        emitter.send(str(score_to_send))

    if key == ord('S') or key == ord('s'):
        # Move forward
        velocity = MAX_VELOCITY
        set_wheel_velocity(velocity, velocity, velocity, velocity)
    elif key == ord('W') or key == ord('w'):
        # Move backward
        velocity = -MAX_VELOCITY
        set_wheel_velocity(velocity, velocity, velocity, velocity)
    elif key == ord('D') or key == ord('d'):
        # Turn right

```

```
    velocity = MAX_VELOCITY
    set_wheel_velocity(-velocity, velocity, -velocity, velocity)
elif key == ord('A') or key == ord('a'):
    # Turn left
    velocity = MAX_VELOCITY
    set_wheel_velocity(velocity, -velocity, velocity, -velocity)
elif key == ord('Q') or key == ord('q'):
    # Quit the program
    print("Exiting...")
    break
else:
    # Stop the wheels when no key is pressed
    set_wheel_velocity(0, 0, 0, 0)
```


(4) counter controller

```
from controller import Supervisor
```

```
SEGMENTS = [  
    [1,1,1,1,1,1,0], # 0  
    [0,1,1,0,0,0,0], # 1  
    [1,1,0,1,1,0,1], # 2  
    [1,1,1,1,0,0,1], # 3  
    [0,1,1,0,0,1,1], # 4  
    [1,0,1,1,0,1,1], # 5  
    [1,0,1,1,1,1,1], # 6  
    [1,1,1,0,0,0,0], # 7  
    [1,1,1,1,1,1,1], # 8  
    [1,1,1,1,0,1,1], # 9  
]  
DIGIT_MATERIALS = [  
    ['a3mat', 'b3mat', 'c3mat', 'd3mat', 'e3mat', 'f3mat', 'g3mat'],  
    # 百  
    ['a2mat', 'b2mat', 'c2mat', 'd2mat', 'e2mat', 'f2mat', 'g2mat'],  
    # 十  
    ['a1mat', 'b1mat', 'c1mat', 'd1mat', 'e1mat', 'f1mat', 'g1mat'],  
    # 個  
]  
ON_COLOR = [0, 1, 0]  
OFF_COLOR = [0.05, 0.05, 0.05]  
  
def set_digit(supervisor, digit_index, value):  
    segs = SEGMENTS[value]  
    for i, seg_on in enumerate(segs):  
        mat_node =  
supervisor.getFromDef(DIGIT_MATERIALS[digit_index][i])  
        if mat_node:  
            mat_node.getField('diffuseColor').setSFColor(ON_COLOR if  
seg_on else OFF_COLOR)  
        else:  
            print(f"找不到 {DIGIT_MATERIALS[digit_index][i]} 這個  
DEF")
```

```

def set_display(supervisor, value):
    value = max(0, min(999, int(value)))
    h = value // 100
    t = (value // 10) % 10
    u = value % 10
    set_digit(supervisor, 0, h)
    set_digit(supervisor, 1, t)
    set_digit(supervisor, 2, u)

supervisor = Supervisor()
timestep = int(supervisor.getBasicTimeStep())

score = 0
receiver = supervisor.getDevice("score_receiver")
receiver.enable(timestep)

while supervisor.step(timestep) != -1:
    while receiver.getQueueLength() > 0:
        data = receiver.getString()
        if data.isdigit():
            try:
                received_score = int(data)
                score += received_score
                print(f"收到得分訊息: +{received_score}, 總分:
{score}")
            except Exception as e:
                print("訊息格式錯誤:", e)
        receiver.nextPacket()
    set_display(supervisor, score)

```

以上是關於第五點 **counter** 以及 **stand** 的連動原因，他們主要由其中的

```

timestep = int(robot.getBasicTimeStep())
emitter = robot.getDevice("score_emitter")
score_to_send = 2

```

```

if distance < 0.11 and (current_time - last_score_time) > cooldown:
    score +=2

```

```
print("得分")
print(distance)
emitter.send(str(score_to_send))
```

以及

```
while supervisor.step(timestep) != -1:
    while receiver.getQueueLength() > 0:
        data = receiver.getString()
        if data.isdigit():
            try:
                received_score = int(data)
                score += received_score
                print(f"收到得分訊息: +{received_score}, 總分: {score}")
            except Exception as e:
                print("訊息格式錯誤:", e)
        receiver.nextPacket()
    set_display(supervisor, score)
```

來傳送訊息

3.期末報告提問

1.本協同專案的目標與主要功能為何？

A:學習如何在不一樣的網路環境下，如何與其他人做同步，好進行作業

2.場景中包含哪幾個主要子系統或模組？

A:youbot-shooter-robot、youbot-stand-robot、feed ball-supervisor、counter board-robot

3.如何在 Webots 中建立一個可移動的投籃輪車？

A:使用 add node 導入 webot 內建 kuka 並轉化為單獨 proto 檔案或轉換成 bass node

4.籃框架的移動模式是如何設計與實現的？

A:基本上核心為鍵盤監聽，from controller import Robot, Keyboard，接著再對 kuka 車進行參數設計

5.投籃輪車的移動控制與發射功能分別由哪些裝置組成？

A: if platform_enabled:

```
    if key == Keyboard.UP:

        for wheel in wheels:

            wheel.setVelocity(MAX_VELOCITY)

    elif key == Keyboard.DOWN:

        for wheel in wheels:

            wheel.setVelocity(-MAX_VELOCITY)

    elif key == Keyboard.LEFT:

        wheels[0].setVelocity(MAX_VELOCITY)

        wheels[1].setVelocity(-MAX_VELOCITY)

        wheels[2].setVelocity(MAX_VELOCITY)

        wheels[3].setVelocity(-MAX_VELOCITY)

    elif key == Keyboard.RIGHT:
```

```

        wheels[0].setVelocity(-MAX_VELOCITY)

        wheels[1].setVelocity(MAX_VELOCITY)

        wheels[2].setVelocity(-MAX_VELOCITY)

        wheels[3].setVelocity(MAX_VELOCITY)

    elif key == ord('Q') or key == ord('q'):

        print("Exiting...")

        break

    else:

        for wheel in wheels:

            wheel.setVelocity(0)

```

此為 kuka 移動控制

```

if mechanism_enabled:

    # Read current motor position (not for relative, but for debug)

    _current_motor_position = sensor.getValue()

    # M key: only take action if allowed and not held

    if key == ord('J') or key == ord('j'):

        if not key_pressed['j'] and current_state == "allow_j":

            motor.setPosition(POSITION_M)

            current_state = "allow_k"

            key_pressed['j'] = True

        else:

            key_pressed['j'] = False

    # K key: only take action if allowed and not held

    if key == ord('K') or key == ord('k'):

```

```

        if not key_pressed['k'] and current_state == "allow_k":

            motor.setPosition(POSITION_K)

            current_state = "allow_j"

            key_pressed['k'] = True

    else:

        key_pressed['k'] = False

```

此為 shooter 控制

6.如何設定投籃輪車的初始位置與限制其活動範圍？

A:直接使用 translation 設定初始位置，限制範圍則靠 floor 限制

7.籃球如何於場景中建立、數量如何設定？

使用 feed ball supervisor 建立，而球限制一次生成只有一顆，但不會限制總數

```

if key == ord('B') and (current_time - last_key_time >= debounce_time):

    if waiting_ball_def is None:

        create_static_sphere(supervisor, *default_feed_pos)

        current_tracked_def = waiting_ball_def

        delete_trajectory_points() # 新球產生時清除舊軌跡

    else:

        print("還有一顆球等待擊出，請先擊出再產生新球。")

        last_key_time = current_time

```

8.如何設計並實作籃球的補球或回收（送球）機構？

球落地後消失

```

else:

    # 球消失，停止追蹤並清除軌跡

    delete_trajectory_points()

    current_tracked_def = None

```

9.投籃動作是如何由控制器程式觸發與執行的？

```

if mechanism_enabled:

    # Read current motor position (not for relative, but for debug)

    _current_motor_position = sensor.getValue()

    # M key: only take action if allowed and not held

    if key == ord('J') or key == ord('j'):

        if not key_pressed['j'] and current_state == "allow_j":

            motor.setPosition(POSITION_M)

            current_state = "allow_k"

            key_pressed['j'] = True

        else:

            key_pressed['j'] = False

    # K key: only take action if allowed and not held

    if key == ord('K') or key == ord('k'):

        if not key_pressed['k'] and current_state == "allow_k":

            motor.setPosition(POSITION_K)

            current_state = "allow_j"

            key_pressed['k'] = True

        else:

            key_pressed['k'] = False

```

由此段控制 shooter 發射及收回

10.操作者如何利用鍵盤控制輪車移動及投籃？

使用 fourbar_controller 並且要 from controller import Robot, Keyboard

11.投籃發射機構的關節與馬達如何配置與驅動？

在 youbot robot 群組下使用 solid 建立如之前作業一樣的 node tree 並在第一個

device 下建立 `motor = robot.getDevice('motor1')`

`sensor = robot.getDevice('motor1_sensor')`

12.籃框架的隨機慢速移動是怎麼實現的？

13.記分板的設計邏輯是什麼？如何判斷得分？

靠 stand 下的 sensor 檢測，並聯絡到 counter board，要建立 emitter 及 receiver，頻道要一致

14.籃球進框判斷使用了哪些感測器或演算法？

靠球體十分靠近 sensor 後觸發條件發送得分訊息，並靠這項訊息聯絡到記分板進行加分

15.投籃輪車與籃框架的碰撞設定如何調整？

使用 boundobject

16.籃框架的位置與運動狀態如何被記錄或顯示？

17.投籃次數與剩餘籃球數是如何統計與顯示的？

18.投籃遊戲結束條件的判定方式為何？

A:`print("\n 已完成圍繞半徑 7m 圓周 12 點繞行並回到起點，任務結束。")`

指令運作後，運行到這裡遊戲結束

19.如何設計多台輪車或多名操作者協同投籃？

A:多產生 youbot robot 群組，並將群組裡的 DEF 及 name 都獨立好，在使用指令命名判別

20.場景中各組件的物理屬性（質量、摩擦等）如何調整？

A:在 worldinfo 中設置了兩個 ContactProperties，

coulombFriction：滑動摩擦係數-->值越高，越不容易滑動

frictionRotation：旋轉摩擦（抗扭轉）-->值越高，越難在原地旋轉或扭轉

mess 則是由 physics 更動

21.YouBot 機構的四輪配置與移動原理為何？

A:使用內建的 kuka 車分離而成，配置是輪框跟輪子、馬達

```
while robot.step(timestep) != -1:

    key = keyboard.getKey()

    # Platform control

    if platform_enabled:

        if key == Keyboard.UP:

            for wheel in wheels:

                wheel.setVelocity(MAX_VELOCITY)

        elif key == Keyboard.DOWN:

            for wheel in wheels:

                wheel.setVelocity(-MAX_VELOCITY)

        elif key == Keyboard.LEFT:

            wheels[0].setVelocity(MAX_VELOCITY)

            wheels[1].setVelocity(-MAX_VELOCITY)

            wheels[2].setVelocity(MAX_VELOCITY)

            wheels[3].setVelocity(-MAX_VELOCITY)

        elif key == Keyboard.RIGHT:

            wheels[0].setVelocity(-MAX_VELOCITY)

            wheels[1].setVelocity(MAX_VELOCITY)

            wheels[2].setVelocity(-MAX_VELOCITY)

            wheels[3].setVelocity(MAX_VELOCITY)

        elif key == ord('Q') or key == ord('q'):

            print("Exiting...")
```

`break`

`else:`

`for wheel in wheels:`

`wheel.setVelocity(0)`

22.投籃機構的多關節設計與驅動邏輯？

A: 發射結構由多層 `HingeJoint` 串接，形成多個旋轉關節，每個關節可配置 `RotationalMotor` 與 `PositionSensor`，根據狀態切換馬達目標角度，實現依序推進與歸位。

23.如何實現籃球補球機構的自動偵測與動作觸發？

A: `feed_ball` 監控場上球數與球狀態，當球被擊出或落地、球數少於上限時，於預設位置自動生成新球。可根據 `Receiver/Emitter` 訊息實現更複雜的自動補球流程。

24.如何在 `Webots` 場景中顯示即時計分與剩餘球數？

A:計分板以七段顯示器設計，`counter_supervisor` 根據 `Receiver` 訊息即時顯示分數。剩餘球數可由 `feed_ball supervisor` 統計並顯示於終端機或額外設計的顯示元件。

25.投籃過程動畫與物理真實性的平衡關鍵為何？

A:機構動畫靠多關節程式控制，物理真實性則由 `Webots` 物理引擎自動處理（質量、摩擦等）。

需合理設定質量、摩擦、馬達參數，並適當設置 `boundingObject`，避免穿透與異常反彈。

26.協同設計過程中，組員如何分工合作？

A:依各控制器/模組分工（如 `youBot_shooter`, `stand`, `supervisor`, 計分板），測試驗證等。以及 `Git` 或雲端同步方案協作，定期合併測試。

27.專案過程的影片、簡報與 `PDF` 報告應包含哪些重點？

A:專案目標與動機、系統結構圖、場景設計、各模組功能、核心程式流程、實際操作（錄影）、分工、遇到的問題與解決策略。

可附錄參數設定、重要程式片段或成果影片連結。

28.如何將專案過程影片嵌入報告網頁？

A:將影片上傳至 youtube 後使用近端的 video 導入影片網址

29.若協同操作過程中發生衝突，如何協調與解決？

A:必須使用 pull requests 將近端儲存好的檔案拉下來覆蓋

30.專案結束後，如何整理心得並反饋於課程平台？

A:使用 git clone 後的近端環境並 cms 編輯內容

31.請描述 youBot 投籃輪車在場景中的初始座標與朝向？

A: .wbt 內 DEF youBot_shooter Robot 設定：translation 0.58 -0.3 0.103

rotation 0.999... -2.89e-10 ... 1.69e-7

代表 $X=0.58$, $Y=-0.3$, $Z=0.103$ ，並有預設朝向。

32.場景中的 youBot 投籃車和籃框架 stand 機器人各自的控制器名稱是什麼？

A: youBot_shooter：controller "fourbar_controller1"

youBot_stand：controller "stand_controller"

33.如何在 Webots 場景檔案中正確設置每個 wheel 的 RotationalMotor 名稱？

A: 每個 wheel (InteriorWheel/ExteriorWheel) 都包含一個 RotationalMotor，name 應分別設為 "wheel1"、"wheel2"、"wheel3"、"wheel4" (youBot_stand 為 wheel5~8)，以便控制器能正確取得，更精確的話可以將 proto 切換至 bass node

34.為什麼每個輪子旁邊都設有 PositionSensor？用途為何？

A: 用於讀取輪子當前角度，便於診斷、PID 控制、或日後精確定位（如自動導航），也可判斷是否正常運轉。

35.該如何利用 IMU 和 GPS 完成 youBot 的定點移動與旋轉？

A: 於控制器取得 GPS (robot.getDevice('gps')) 與 IMU (robot.getDevice('inertialUnit')) 數據，讀取當前座標與朝向。

實作移動控制邏輯（如 PID），當到達目標座標與朝向時停止馬達，在程式內判斷座標誤差與角度誤差是否在容許範圍。

36.如何設計 youBot 能夠精確側移指定距離？程式上如何判斷移動完成？

A: 設定對應四輪速度組合實現側移，啟動前記錄當前 GPS X/Y，持續比較與目

標距離，達成預設距離時停止馬達，可加上 IMU 監控，避免偏轉。

37.是否試過讓 youBot 自動從起點移動到指定座標並朝向某點嗎？描述操作步驟。

A:否，不過應該可以利用 `stand_circular` 進行修改實現

38.在控制器內，如何區分前進、後退、旋轉、側移等不同移動模式？

A: `while robot.step(timestep) != -1:`

```
    key = keyboard.getKey()
```

```
    # Platform control
```

```
    if platform_enabled:
```

```
        if key == Keyboard.UP:
```

```
            for wheel in wheels:
```

```
                wheel.setVelocity(MAX_VELOCITY)
```

```
        elif key == Keyboard.DOWN:
```

```
            for wheel in wheels:
```

```
                wheel.setVelocity(-MAX_VELOCITY)
```

```
        elif key == Keyboard.LEFT:
```

```
            wheels[0].setVelocity(MAX_VELOCITY)
```

```
            wheels[1].setVelocity(-MAX_VELOCITY)
```

```
            wheels[2].setVelocity(MAX_VELOCITY)
```

```
            wheels[3].setVelocity(-MAX_VELOCITY)
```

```
        elif key == Keyboard.RIGHT:
```

```
            wheels[0].setVelocity(-MAX_VELOCITY)
```

```
            wheels[1].setVelocity(MAX_VELOCITY)
```

```
            wheels[2].setVelocity(-MAX_VELOCITY)
```

```
            wheels[3].setVelocity(MAX_VELOCITY)
```

```
elif key == ord('Q') or key == ord('q'):
```

```
    print("Exiting...")
```

```
    break
```

```
else:
```

```
    for wheel in wheels:
```

```
        wheel.setVelocity(0)
```

39.在程式中 `set_wheel_velocity` 的四個參數順序分別對應哪四個輪子？

YouBot 的 Keyboard 控制有哪些特定按鍵操作？（請舉例說明）

A: 對應 `wheel5`, `wheel6`, `wheel7`, `wheel8`，一般順序為：右前、左前、右後、左後。

例如 `set_wheel_velocity(v1, v2, v3, v4) -> wheel5, wheel6, wheel7, wheel8`

`youBot_shooter`：

上下左右鍵：四輪移動

J：發射動作

K：歸位動作

Q：退出程式

`youBot_stand`：

W/A/S/D：控制四輪移動

J/K：可自訂功能

Q：退出

`Supervisor`：

B：產生新球

J：球變動態可擊出

（自行調整過後）

40.投籃機構的多個 `HingeJoint` 關節是如何層層串接的？結構如何設計？

A: 在 `.wbt` 場景檔中，`Solid` 下嵌套 `HingeJoint`，每個 `HingeJoint` 的 `endPoint`

再連接下一個 Solid，形成多節機械臂（ joint1→press→joint2→connector→joint3→shooter）。

每層可加馬達與感測器，達到多自由度投射動作。

41.如何觸發投籃發射動作？對應按鍵是什麼？程式流程為何？

A: 按下 J 鍵（Keyboard），fourbar)controller 檢查目前狀態允許，將馬達目標角度設為發射位置（POSITION_M），完成後切換狀態只允許 K 歸位。

歸位則按 K 鍵，motor 設回初始角度（POSITION_K），狀態切回。（2）

42.籃框架 stand 的移動演算法是否有隨機性？該如何實現？

A: 預設用鍵盤操作，如需隨機可於主迴圈每隔一段時間隨機選擇方向與速度，或用簡單的隨機數決定四輪馬達速度。

43.補球/送球機構自動補球的訊息傳遞流程是什麼？使用了哪些 Webots 裝置？

A: 由 feed_ball superviso 負責監控與生成新球，當偵測到球被擊出或落地，即於指定位置產生新靜止球，如需多機協同，可用 Emitter（stand）與 Receiver（supervisor）傳遞補球請求。

44.Emitter/Receiver 在記分板與補球控制中扮演什麼角色？請詳細說明一個訊息傳遞範例。

A: 當球進框時，youBot_stand stand_controller 偵測到得分，透過 Emitter.send(str(score_to_send)) 傳送訊息。

controller Supervisor 用 Receiver.getString() 讀取訊息，解析後累加分數並更新七段顯示器。

補球也可用類似機制由輪車或 stand 向 supervisor 傳遞球數需求。

45.七段顯示器記分板的 segment 如何根據得分訊息動態變色？

A: controller Supervisor 根據分數拆解百、十、個位，對應 segment 陣列，依照數字設定各 segment 的 Material.diffuseColor（ON_COLOR/ OFF_COLOR）。

46.投籃得分是根據哪個感測器資料來判斷？感測器設置於何處？

A: 主要依靠 Infra-red DistanceSensor，設於籃框下方，當球距離低於某閾值即視為進框。

47.是否設計過球進框後自動加分的程式嗎？其邏輯為何？

A: 是，於 `youBot_stand` `stand_controller` 檢查距離感測值與冷卻時間，若滿足條件則自動加分並發送訊息給計分板。

48.如何防止球體與場景物件發生非預期的穿透或異常彈跳？

A: 應調整物理屬性：`boundingObject` 精確包覆、質量設定合理、摩擦與彈性 (`contactProperties`)調整、避免球初速度過大。

49.請說明如何設定與調整球體和機械部件的質量、摩擦等物理參數。

A: 在 `Solid` 內部的 `physics` 區塊設定 `mass`、`density`，在 `WorldInfo` 設定 `contactProperties`（摩擦、彈性、滑移等）。

50.是否試過多台 `youBot` 輪車同場協同投籃嗎？多機協同時，訊息或控制會有什麼衝突？

A: 可複製多台 `youBot`，每台配專屬控制器與 `Emitter channel`，若共用補球/計分資源，需設計訊息區分與資源互斥，否則可能產生搶球、重複加分等衝突。

51.補球流程中如何確保每次只會有一顆球被正確補給到輪車？

A: else:

```
print("還有一顆球等待擊出，請先擊出再產生新球。")
```

```
last_key_time = current_time
```

由這一段進行球是否為被更動

52.若希望依據剩餘球數或投籃次數動態結束遊戲，程式該如何編寫？

A: 在 `supervisor` 或主控制器中設計變數統計剩餘球或已投次數，達到上限即結束主循環，顯示結束訊息或通知計分板。

53.自動補球與自動投籃能否串接成完整循環？你如何設計流程與狀態管理？

A: 可以。設計 `state machine`，當球補給就緒時自動進行投籃發射，投籃後自動生成新球或等待補球訊號，直到遊戲結束。

54.場景中 `Viewpoint` 設定有何作用？你試過調整攝影機角度觀察不同視角嗎？

A: `Viewpoint` 決定初始攝影機位置與朝向，我可自由調整 `position` 與 `orientation` 觀看不同角度，但更常使用滑鼠左鍵右鍵進行視角的旋轉或橫移。

55.請描述該如何嵌入 `YouTube` 影片到個人或分組期末報告網頁？

A: 將影片上傳至 `youtube` 後使用近端的 `video` 導入影片網址

56.專案過程中曾遇到什麼協作或整合困難？如何解決？

A: 遇到場景物件名稱重複、控制器通訊衝突、檔案合併時 mesh 路徑錯誤等問題。解決方法為明確命名規範、使用 Git 分支、合併前多測試、溝通協調。

57.如何調整記分板七段顯示器 segment 的顏色及樣式？

A: DEF c3 Shape {

 appearance Appearance {

 material DEF c3mat Material {

 diffuseColor 0 1 0

由這個節點控制預設顏色

58.場景與控制器檔案如何在其他同組協同組員的電腦上執行？

A: .前置步驟與 3-1、3-2 是一樣的，要先設置好規範好的 ipv6

將老師的 youbot_cart_w10_websocket 下載下來

將裡面的 client、server 的.py 檔案放至 SciTE 使用，並將內容中最上面的 pip install websocket-client websocket-server keyboard 使用 cmd 下載下來

使用 cmd 進行 webot 與網頁串聯

SciTE 中執行 client 跟 server 檔，因為 youbot 中的 controller 是 extern，是指向外部

在 start.bat 檔案中，必須將 python 路徑設置成 set

```
PYTHONPATH=%Disk%:\Python313\DLLs;%Disk%:\Python313\Lib;%Disk%:\Python313\Lib\site-packages;%Disk%:\NX\NXBIN\python;%Disk%:\Webots_2023b\msys64\mingw64\bin;%Disk%
```

python 路徑下有一個 REM for Webot 其中也得新增 webot home

```
set PYTHON_PATH=%Disk%:\Python313
```

```
set WEBOTS_HOME=%Disk%:\Webots_2023b
```

都執行完畢後就可以在 localhost 上執行，若第一次無法成功可以將 webot 重新啟動執行串流

59.該如何驗證投籃流程、計分、補球、遊戲結束等每個功能都運作正常？

A: 直接進行模擬測試或手動操作測試每個流程（投籃、得分、補球、遊戲結束），觀察終端輸出、計分板顯示、球生成等是否正確。

60.請自行於 Webots 場景新增或刪除一顆籃球？操作步驟為何？

A: 新增：按下 B 鍵，呼叫 `create_static_sphere` 於指定位置生成一顆球。

刪除：於 `supervisor` 追蹤該球的 DEF 名稱，`supervisor.getFromDef().remove()` 將其移除，也可直接於 `.wbt` 場景檔加入或刪除籃球

61.如何在場景中設定籃球的尺寸、顏色與物理屬性？

A: 在 `create_static_ball/create_dynamic_ball` 時設定 `Sphere` 的 `radius`、`Material.diffuseColor (r,g,b)`，並於 `physics` 區塊設 `mass`、`density`。

也可於 `.wbt` 直接編輯球的定義內容。

62.是否試過修改 `youBot` 或 `stand` 機構的質量或摩擦係數嗎？如何修改？

A: 是，可於 `.wbt` 的 `physics (mass、density)`、`WorldInfo.contactProperties (coulombFriction、frictionRotation 等)` 調整；也可針對每個部件單獨設置。

63.若要限制 `youBot` 只能於指定區域活動，如何於控制器實作？

A: 讀取目前座標（GPS），判斷是否超出預設區域範圍，若超出則將輪子速度設為 0 或自動修正回區域內。

64.該如何設定並驗證 `youBot` 的初始朝向與旋轉角度？

A: 在 `.wbt` 檔 DEF `youBot_shooter Robot` 設定 `rotation` 欄位（四元數格式）可指定初始朝向，控制器啟動時可讀取 `IMU/InertialUnit` 確認是否為預期值，必要時自動修正。

65.場景中籃框（`stand`）如何設定隨機移動的範圍與方向？

A: 於 `stan_controller` 設計隨機移動演算法，每次隨機選擇方向速度，並根據 GPS 判斷不得超出設定區域，需自動轉向或停下。

66.你試著讓投籃輪車自動對準並靠近籃框嗎？程式邏輯是什麼？

A: 於控制器讀取兩者座標，計算相對向量，自動規劃移動方向與距離，先側移再前進，最後調整朝向，直到距離與角度誤差低於閾值。

67.當投球未命中籃框時，如何於程式中記錄「未得分」？

A: 每次發射後設定 `timer` 或 `flag`，若在一定時間內未觸發距離感測器則判定為

未得分，記錄於變數或顯示於終端。

68.你曾經手動重置全部籃球的位置嗎？操作流程為何？

A: 於 `.wbt` 檔案直接重設球的 `translation`。

70.你是否嘗試過將計分板顯示改為兩位數或四位數？如何調整？

A: 在 `counter` 下再建立一個 `Pose` 作為 `a4mat` 等等並在 `conter_supervisor` 下新增或刪減

71.如何讓記分板在遊戲結束後自動顯示「Game Over」或其他訊息？

A: `controller Supervisor` 偵測到遊戲結束條件時，可將所有 `segment` 設為紅色或特定組合（如 `'88:88'`），或於終端顯示訊息。

若需顯示文字，可設計 `LED/matrix` 顯示裝置。

72.你是否有針對球體數量不足時自動補球？程式如何設計？

A: 是，於 `feed_ball supervisor` 統計場上球數，若低於預設上限則自動產生新球，或監聽來自輪車/`stand` 的補球請求。

73.如何於控制器中實現「投球冷卻時間」以防止連續發射？

A: 設計 `last_shot_time` 變數，每次投球後紀錄時間，只有當現時距離上次投球已過 `cooldown` 秒數時才允許再次發射。

74.你能否讀取與顯示 `youBot` 的即時速度或移動路徑？

A: 可以，使用與 `stand_circular` 控制器內容，同理能夠使用再 `stand_controller` 進行更改

75.你有測試過將投籃流程的控制訊息以文字訊息（如 `LOG`）即時顯示在終端機嗎？

A: 有，於所有控制器程式（`print/logging`）可即時顯示當前狀態、座標、動作、分數、感測器數值等。

76.場景中的 `DistanceSensor` 如何設定量測距離與靈敏度？

A: 於 `.wbt` 設置 `lookupTable`、`numberOfRays`、`type`，可於控制器端設定啟用與讀取頻率，調整臨界值以適應球體尺寸。

77.是否有令籃框架進行非線性或加速度運動？如何實作？

A: 可於 `stand_controller` 設計速度變數隨時間變化，或隨機調整輪速，實現非線

性、加速度或震盪運動。

78.如何將 youBot 移動與投籃流程整合成一個自動化測試腳本？

A: 於控制器設計自動行為狀態機：依序自動移動至指定點→調整朝向→發射投籃→記分→補球→重複，直到結束。

可用計時器與座標判斷切換狀態。

79.是否試過在同一場景下同時建立兩個以上的計分板嗎？操作方式為何？

A: 可以，於 .wbt 複製 counter_supervisor Robot，分別設不同 Receiver channel 與控制器，於 counter_supervisor 內設計 channel 區分訊息來源。

80.如何於 Webots 場景中加入倒數計時器或遊戲計時功能？

A: 啟動時記錄起始時間，於每步 step() 計算剩餘/經過秒數，達到指定時間即結束遊戲並顯示訊息，若需視覺顯示，可設計額外顯示裝置或於終端輸出。

81.能否於程式中偵測 youBot 是否卡住或發生碰撞？如何處理？

A: 可以。於控制器中連續比較 GPS/IMU 資料，若輪子有速度但座標或角度變化極小，可判斷為卡住，或透過 Webots 物理引擎的碰撞回饋（如有），或輪子 PositionSensor 角度不變時警示。

處理方式可自動嘗試倒退、側移，或提示操作者。

82.當有多台 youBot 輪車時，如何區分各自的控制與訊息通道？

A: 每台 youBot 可設不同 controller 名稱及 Emitter/Receiver 的 channel，於程式內指定通訊頻道，這樣各自只接收/發送自己的訊息，避免干擾。

83.若希望將補球機構設計為「多點補球」，該如何設計？

A: 場景中可放置多個補球點（如多個 supervisor 或多個球生成區），每台輪車靠近指定點由該點生成球。

每個補球區可有專屬 Emitter/Receiver channel，或由 supervisor 控制球分配。

84.是否測試過不同籃球質量對投籃飛行軌跡的影響？結果如何？

A: 是。於球體 physics 區塊調整 mass/density，發現質量變小，飛行軌跡較平穩不易跳動；質量太大則彈跳或受空氣、碰撞影響大，且容易出現 bug。

需多次調整找最佳值

85.如何於模擬中手動或自動錄製並回放投籃過程？

A: Webots 有內建錄影功能（start video recording），點擊錄影/回放按鈕即可。

86.是否能於控制器程式內同時整合多種鍵盤/滑鼠/自動控制模式？

A: 可以。於程式內設計模式變數，根據鍵盤輸入切換手動、半自動、全自動（如自動移動/發射）。

87.若曾遇到過場景載入出現 mesh 檔案遺失或錯誤，該如何排除？

A: 主要先檢查 mesh/cad 檔案是否與 .wbt 檔案相對路徑正確，再確認 obj 檔案是否毀損

88.是否於專案過程中建立過專屬的 proto 或自訂場景物件？如何設計？

A: 在自行製作 proto 檔案時發現 proto 檔案無法直接生成，必須得自行建立，尤其是得將"]{" 這兩個括號給分出來分清楚，而以下就是自行建立 proto 檔案的起手內容

```
#VRML_SIM R2023b utf8
```

```
PROTO shooter1 [
```

```
    # 場景控制
```

```
    field SFVec3f    translation  0 0 0
```

```
    field SFRotation rotation    0 0 1 0
```

```
]{
```

```
    Robot {
```

89.如何批次測試多種投籃參數（如角度、力度）並記錄結果？

A: 先於.wbt 中設定 maxvelocity 於控制器設計迴圈自動調整投籃馬達角度/速度，每次發射後記錄球飛行座標、最遠距離、是否得分等資料，存檔分析。

90.當 Webots 更新版本或變更 API，該如何維護與修正原有專案？

A: 可以嘗試確認.wbt 檔案中（文字檔）#VRML_SIM R2023b utf8 的這一部分，進行修改