

CAD電腦輔助設計與實習



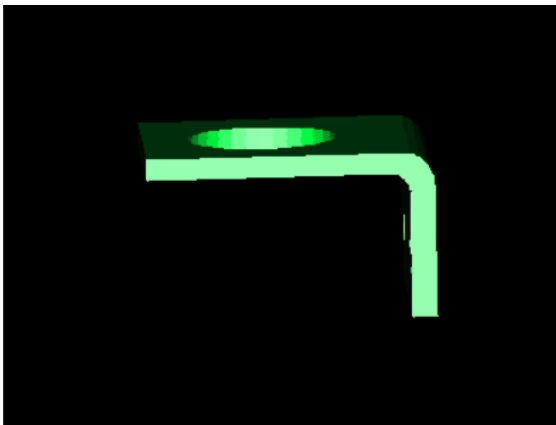
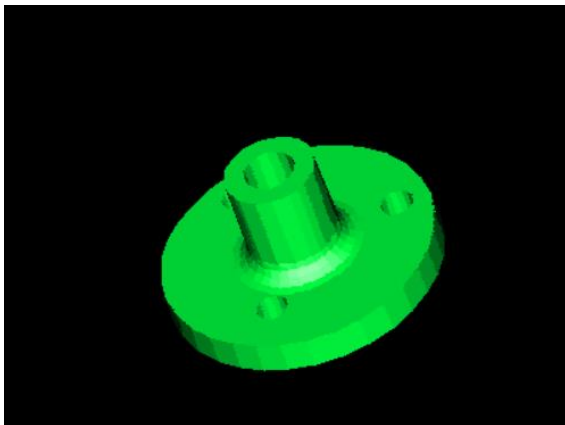
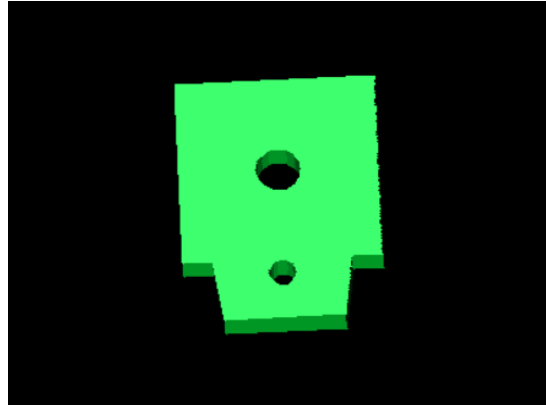
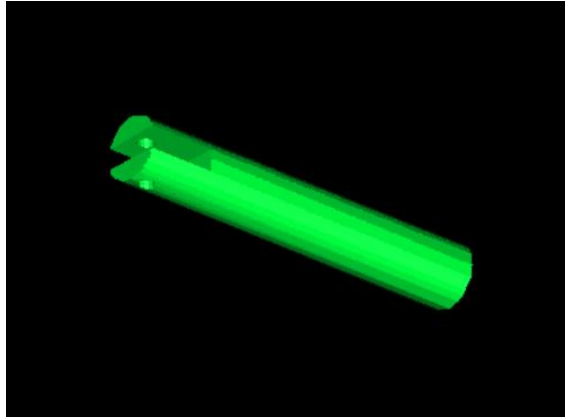
41223131 曹祐豪

41223150 顏宇慶

41223141 楊博泓

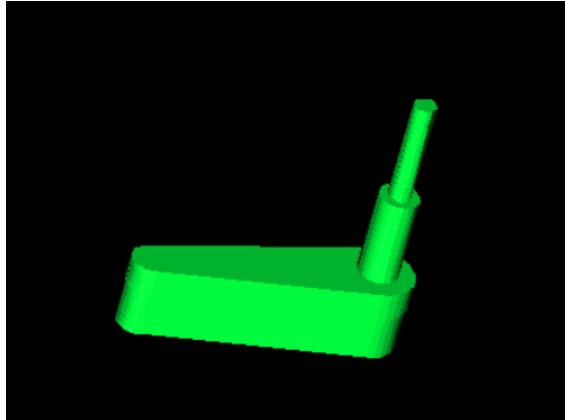
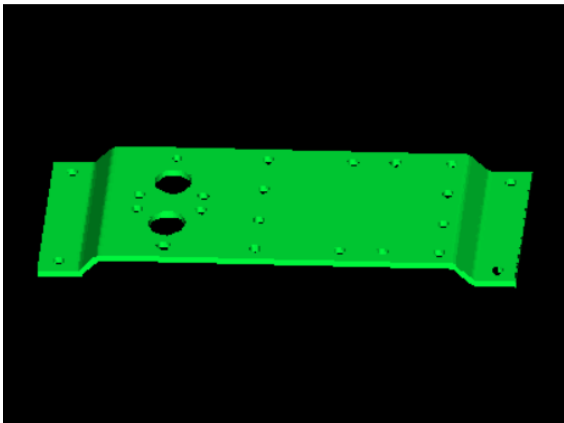
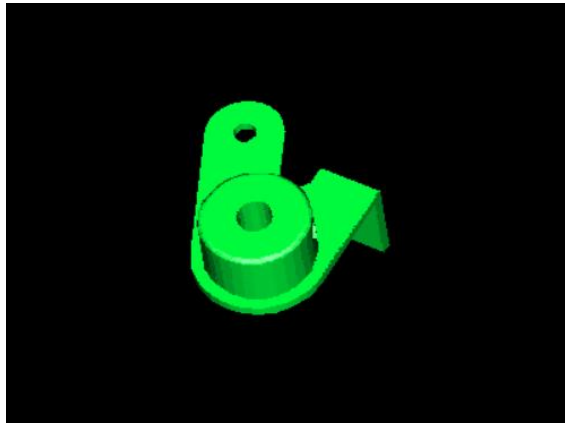
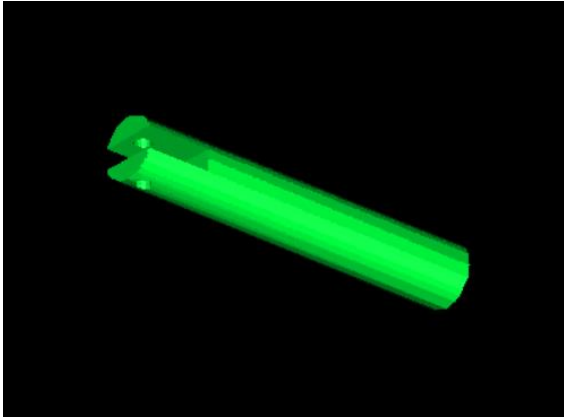
零件的分工-41223131

利用solvespaces畫圖



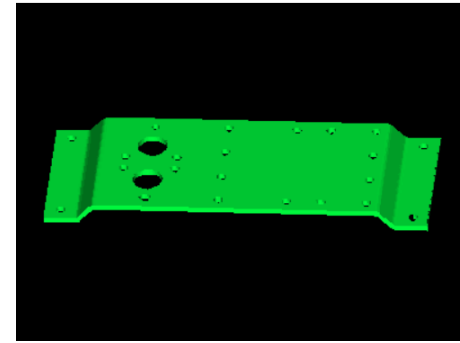
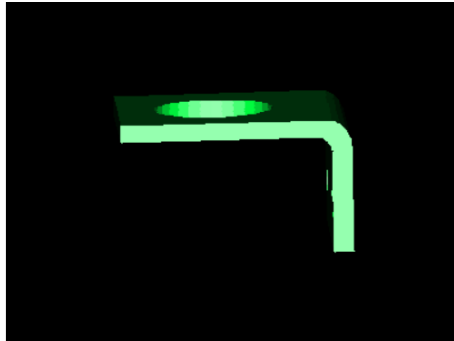
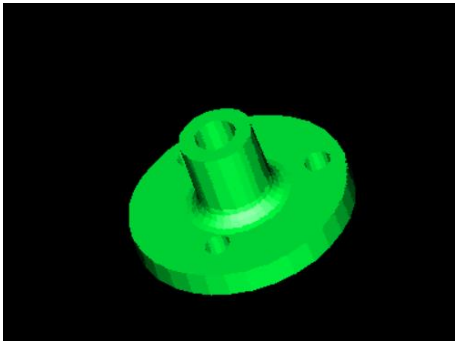
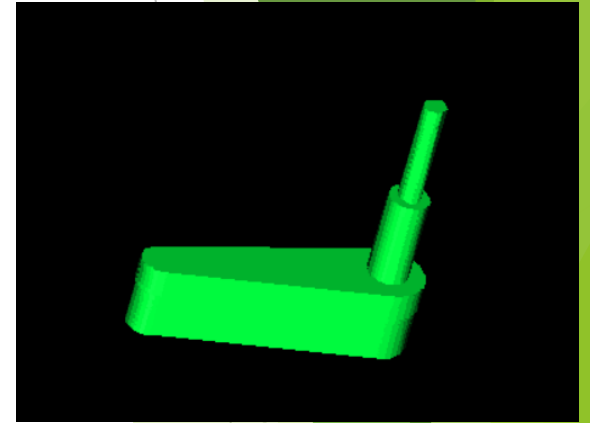
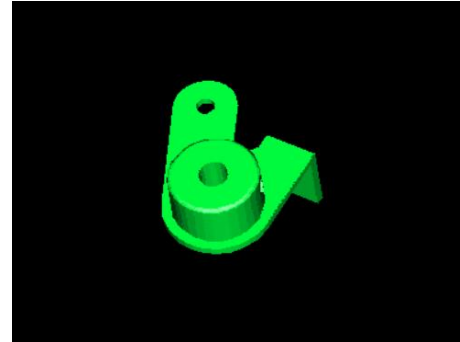
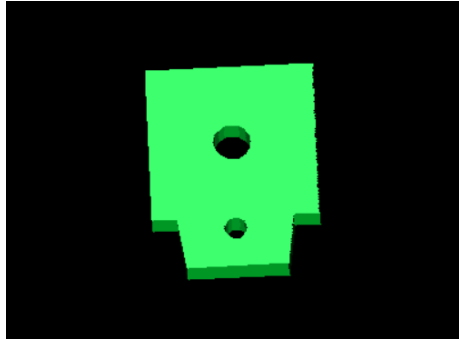
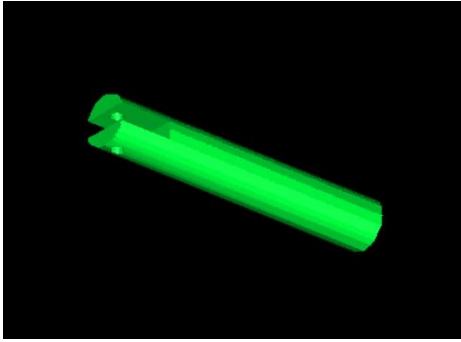
零件的分工-41223150

利用solvespaces畫圖

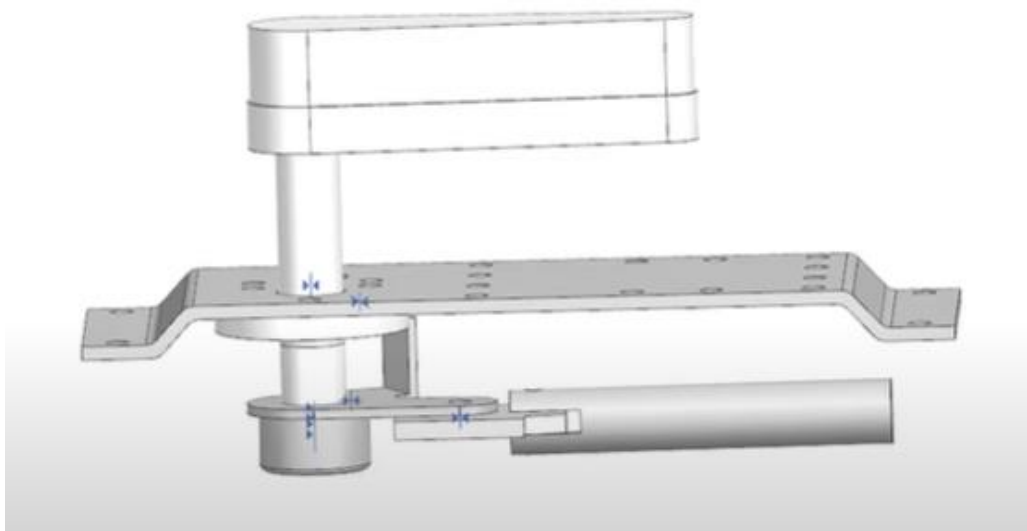


零件的分工-41223141(自行完成的)

利用solvespaces畫圖



組合

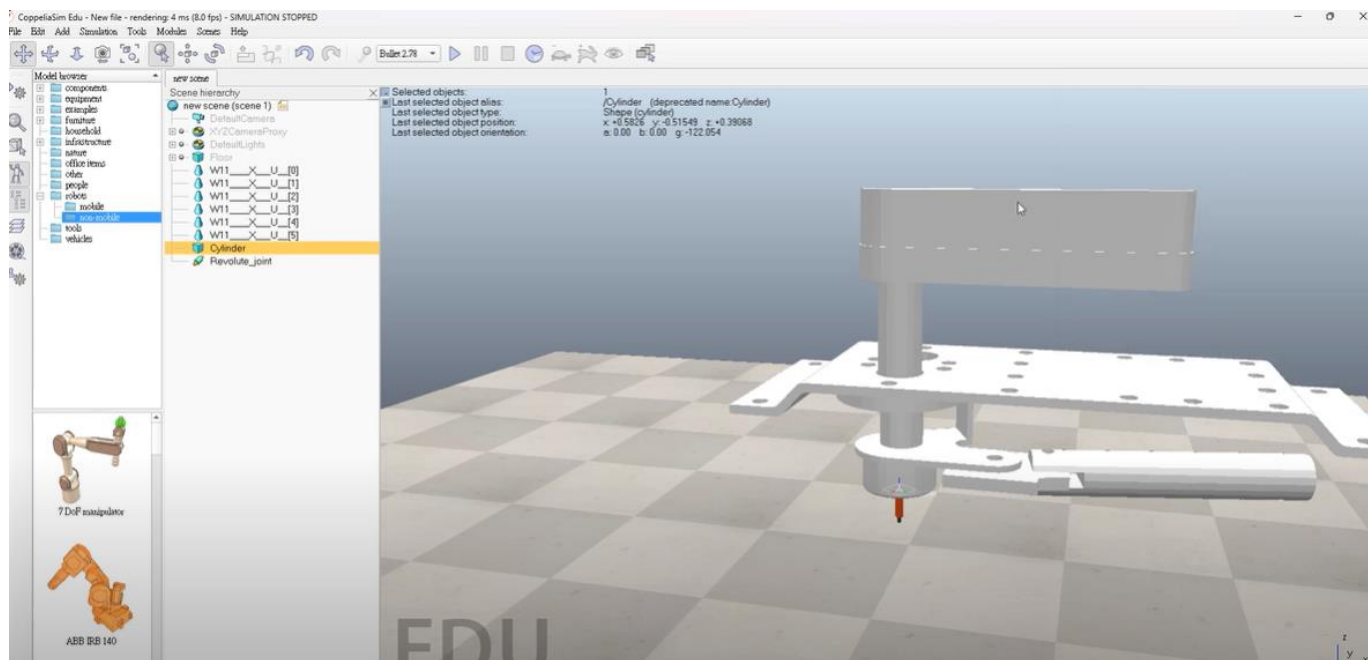


利用NX導入零件開始組合

操作方法：

1. 打開nx
2. 建立一個新的file開啟assembly
3. 匯入.prt 零件至裡面
4. 按Assemblies 裡面的move Component 移動至正確位置
5. 如果要控制位置與同心圓的部分
點選組體以及組合操作即可完成

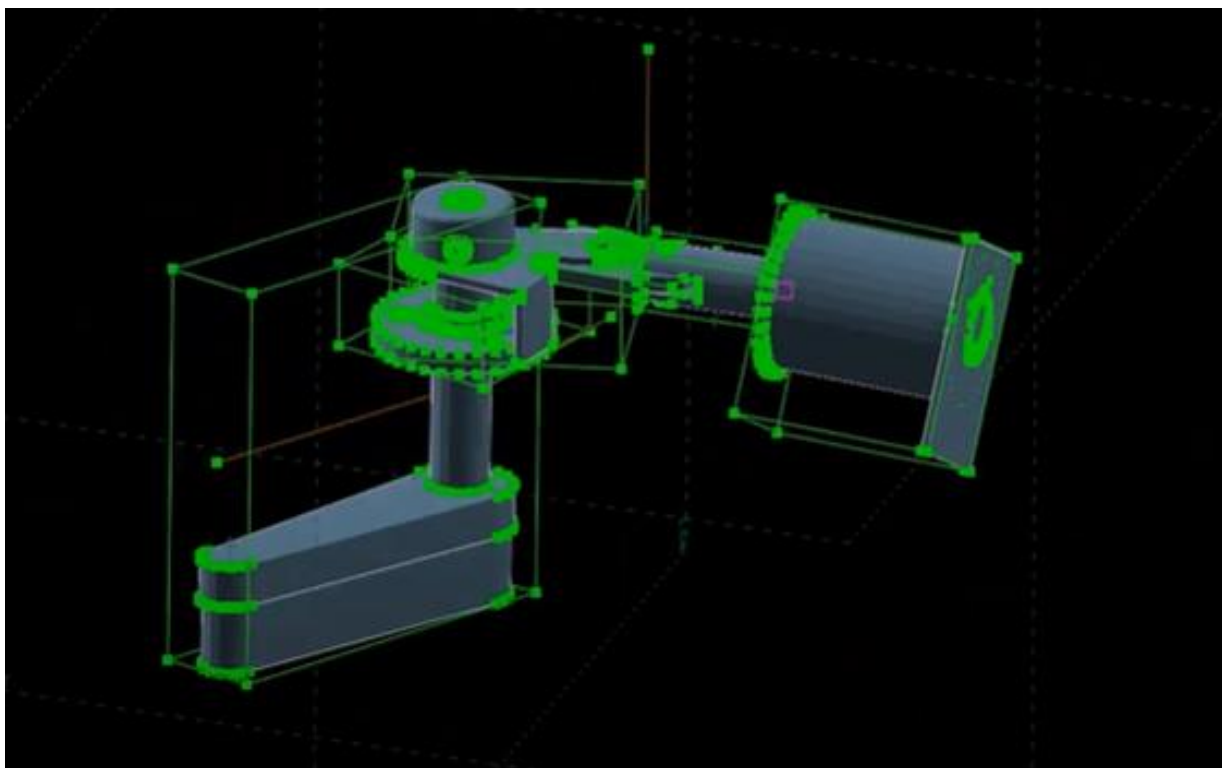
旋轉（試用軟體）



操作方法：

1. 利用coppelasim嘗試著進行旋轉
2. 匯入stl組合好的零件 注意XYZ軸的方向
3. 分解零件
4. 使用一系列做出傳動桿件
5. 叫出旋轉接頭與傳動桿件搭配(傳動桿件的速度可調整 可以隱藏起來)
6. 調整對應位置(調整碰撞檢測與物體動態)

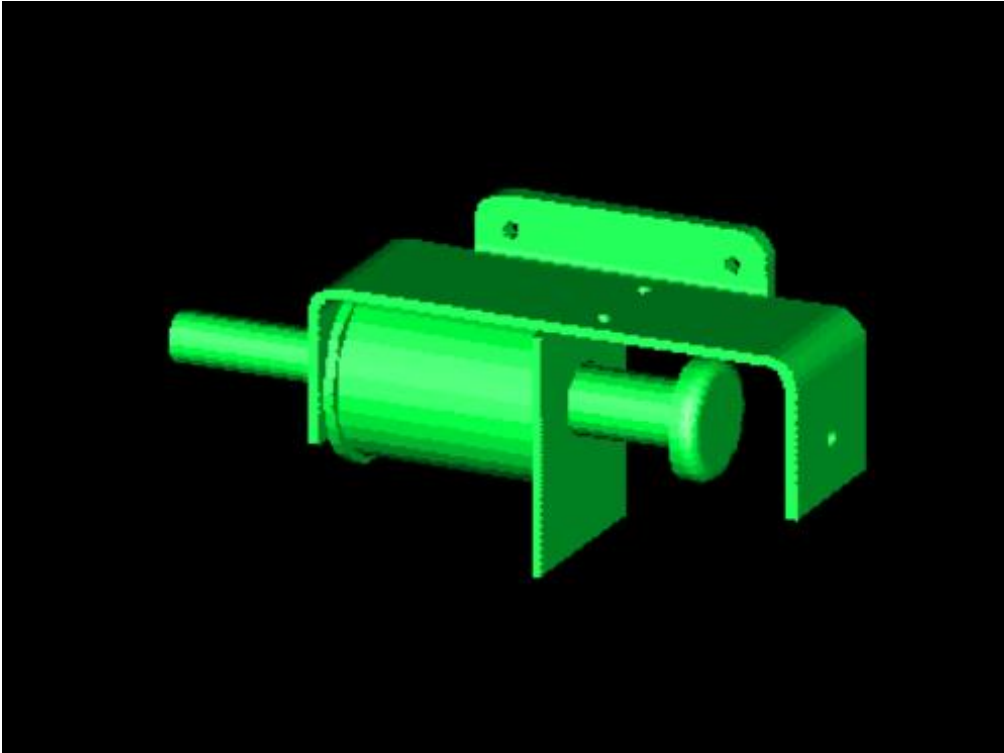
solvespaces組合



操作方法：

1. 利用solvespaces 匯入零件 (s l v s)
2. 參考老師的輔助線與做動的關係(重力方向)
3. 進行組合(這部分還是不太會操作)

組合與零件繪圖



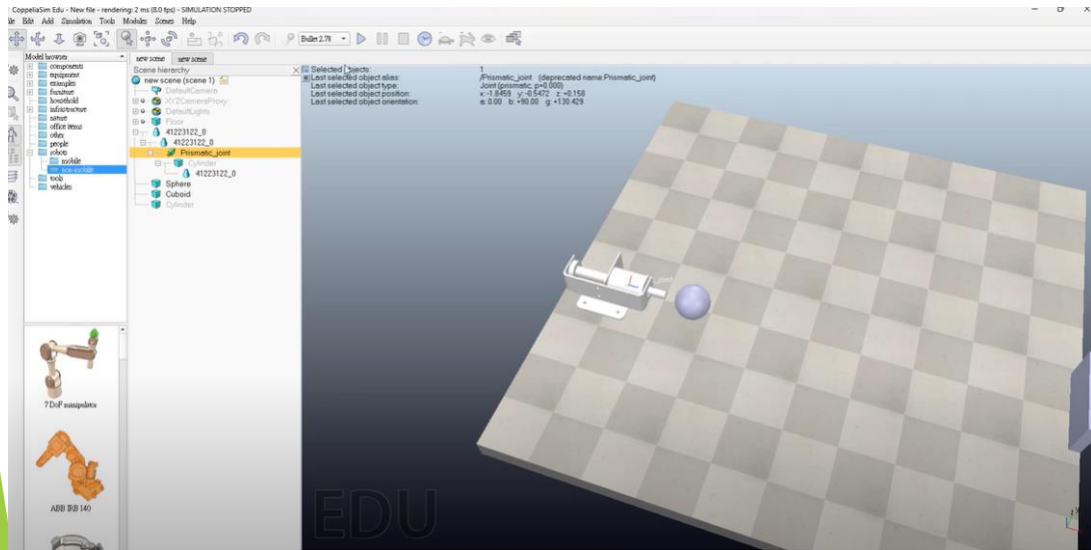
操作方法：

1. 利用slovespaces與NX繪圖進行組合
2. 注意傳動桿件與是否放在施力最佳的地方
(也避免與外殼碰撞到)

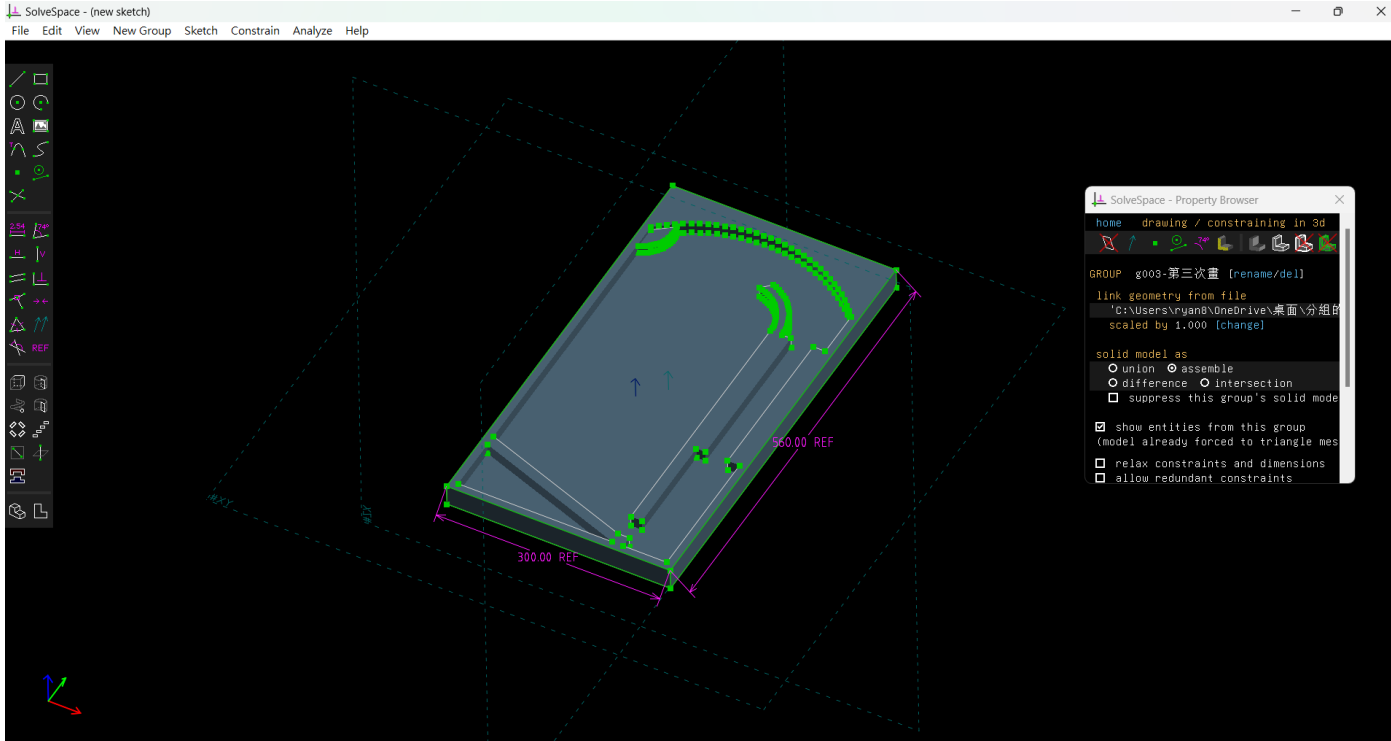
組合與零件繪圖

操作方法：

1. 先匯入組合好的發射器，且調整位置
2. 分解零件 順便叫出檔板以及球體
3. 傳動桿件的部分使用移動功能，盡量調整到能施力最大的部分(因 $W=FS$ 的關係)，作用力相同的情況下，才能使球初始的速度上升(球體質量也會有所關係 $F=MA$ 質量越大，作用力也就越大，因為質量和作用力成正比)
4. 傳動桿件做出一個圓棒主要目的:是模擬轉動運動的傳遞行為
5. 叫出接頭與馬達相接，主要目的是使傳動桿件可以進行移動施力
6. 調整順序:外殼-馬達-傳動接頭-傳動圓棒-傳動桿件
7. 調整傳動桿件的位置（作用方向）就可以開始擊球

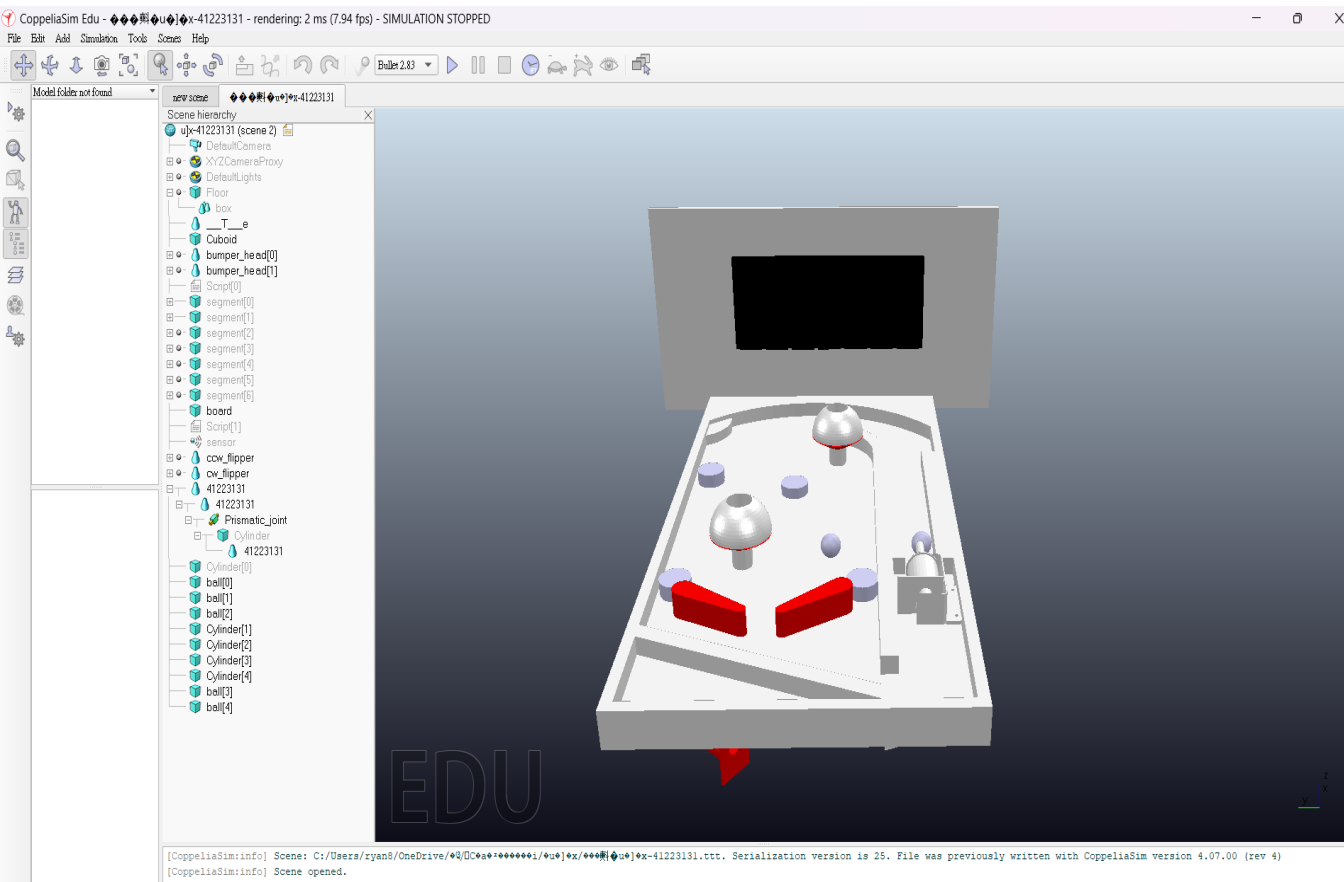


彈珠檯繪製-solvespaces



1. 先在solvespaces 畫出寬300mm長560mm(尺寸有自己調整過)的彈珠台模型(和內建特徵)

彈珠檯繪製-coppeliasim



1. 打開coppeliasil開啟stl，放大10倍進行匯入，
2. 匯入且複製做好的發射器、緩衝頭、撥桿、記分板(目前還尚未了解記分板的模式)
3. 調整好位置，匯入的stl打開碰撞檢測，才能不會讓球體掉入虛空(調整速度、球的質量)，匯入內建的圓柱狀障礙物或著長方體的隔板(打開碰撞檢測，關閉動態檢測，避免說物體在亂移動)
4. 啟動且開啟Ipv6匯入進行程式模擬即可完成。

彈珠檯繪製-程式控制

```
control.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1 control.py
# pip install pyzmq cbor keyboard
from coppeliasim_zmqremoteapi_client import RemoteAPIClient
import keyboard

# Connecting to the CoppeliaSim server
client = RemoteAPIClient('localhost', 23000)

print('Program started')
sim = client.getObject('sim')

# Get the handle for the slider (prismatic joint)
cw = sim.getObject('/cw_joint')
ccw = sim.getObject('/ccw_joint')

# Starting the simulation
sim.startSimulation()
print('Simulation started')

# Main control loop
def main():
    # Keep running until simulation is stopped
    while True:
        if keyboard.is_pressed('p'): # Move slider to -0.15 position
            print("p is pressed")
            sim.setJointTargetPosition(cw, -0.25)

        if keyboard.is_pressed('l'): # Reset slider to the original position
            print("l is pressed")
            sim.setJointTargetPosition(cw, 0.0) # Reset to the initial position

        if keyboard.is_pressed('w'): # Move slider to -0.15 position
            print("w is pressed")
            sim.setJointTargetPosition(ccw, -0.28)

        if keyboard.is_pressed('s'): # Reset slider to the original position
            print("s is pressed")
            sim.setJointTargetPosition(ccw, 0.0) # Reset to the initial position

        if keyboard.is_pressed('t'): # Stop the simulation when 'q' is pressed
            print("t is pressed - stopping simulation")
            sim.stopSimulation()
```

開啟ipv6啟動這程式
開始進行撥桿的操作

前提:

(發射器不是用程式控制的)

W、S-----左邊撥桿

P、L-----右邊撥桿