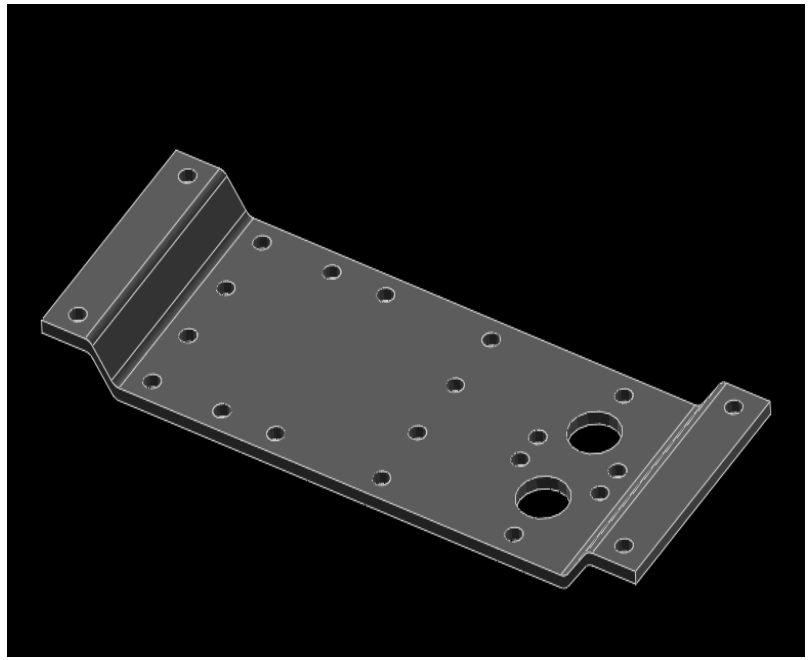
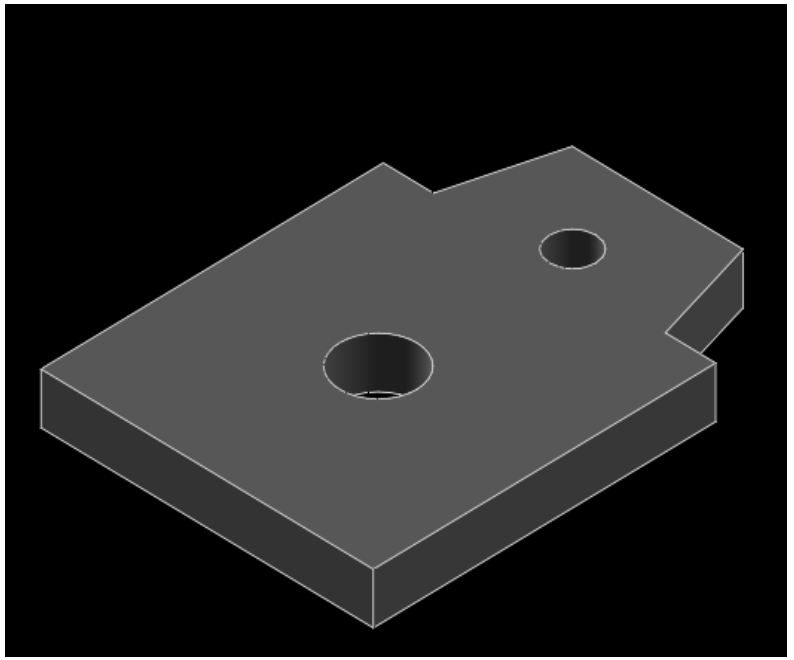
電腦輔助設計與實習-彈珠檯

組員:41223208-葉如意 41223209-鄭佳青 41223234-陳奕廷

零件繪製分工

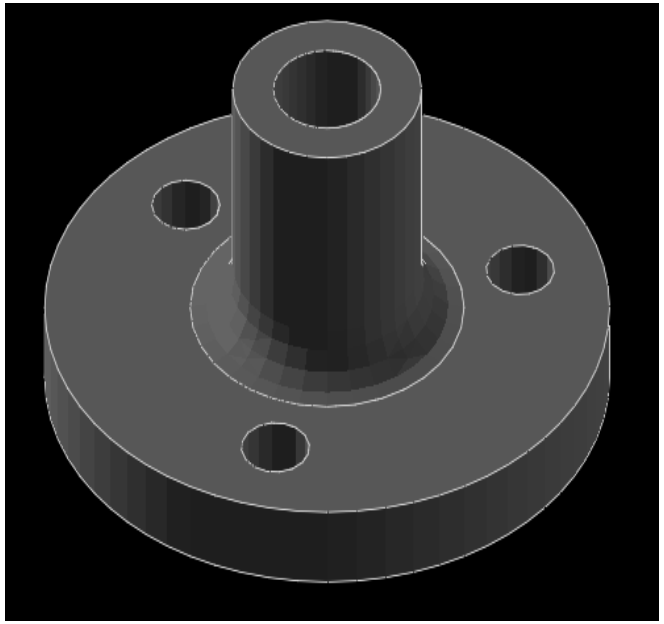
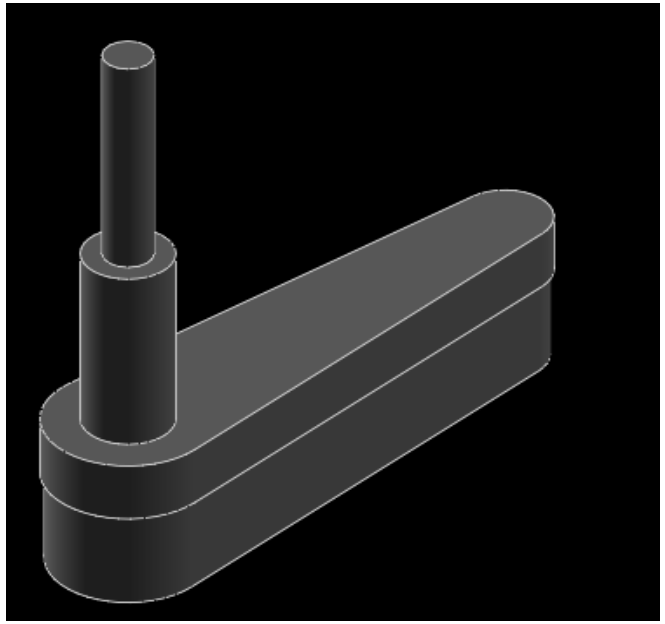
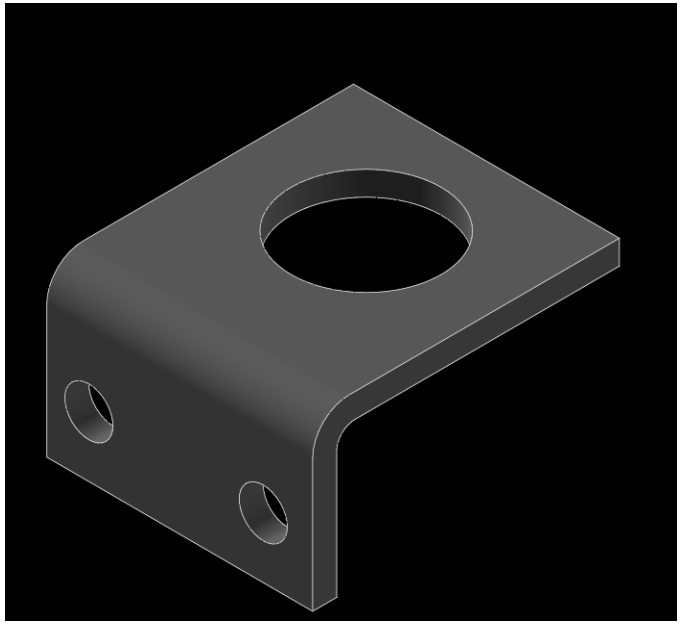
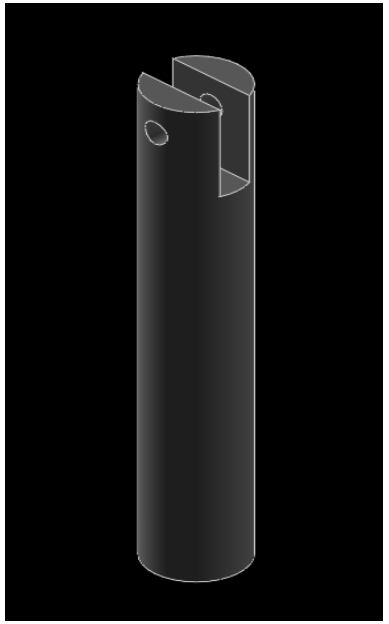
41223208-葉如意 負責繪製零件





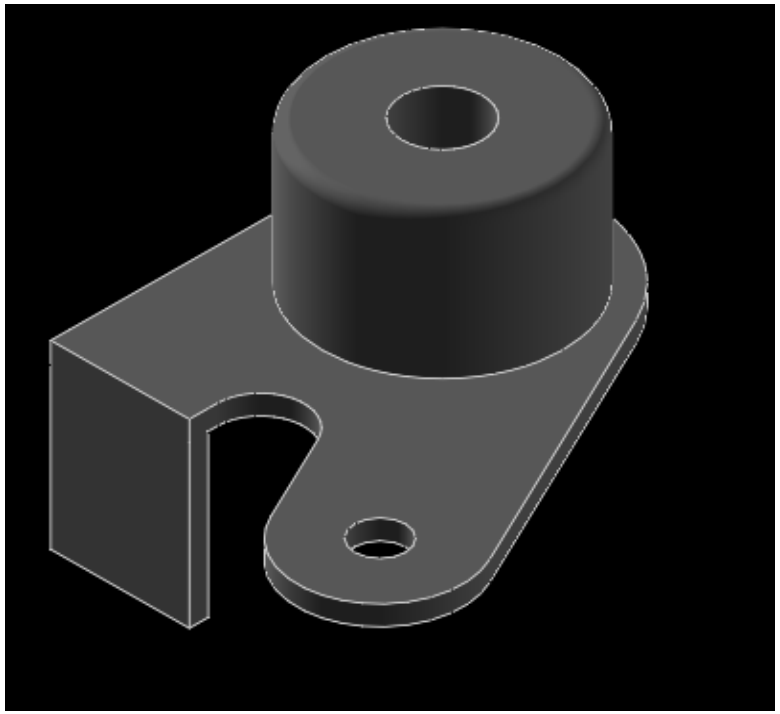
零件繪製分工

41223209-鄭佳青 負責繪製零件

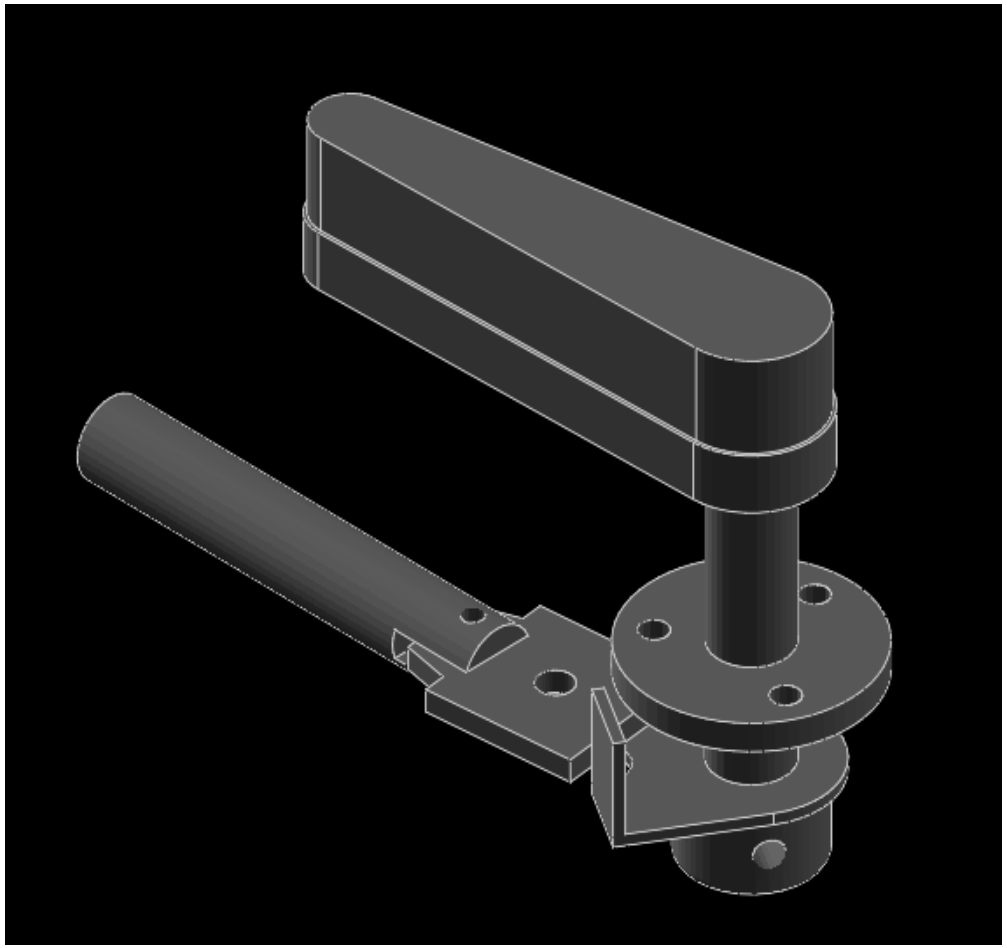





零件繪製分工

41223234-陳奕廷 負責繪製零件



組合-全員討論



1.開啟solvespace

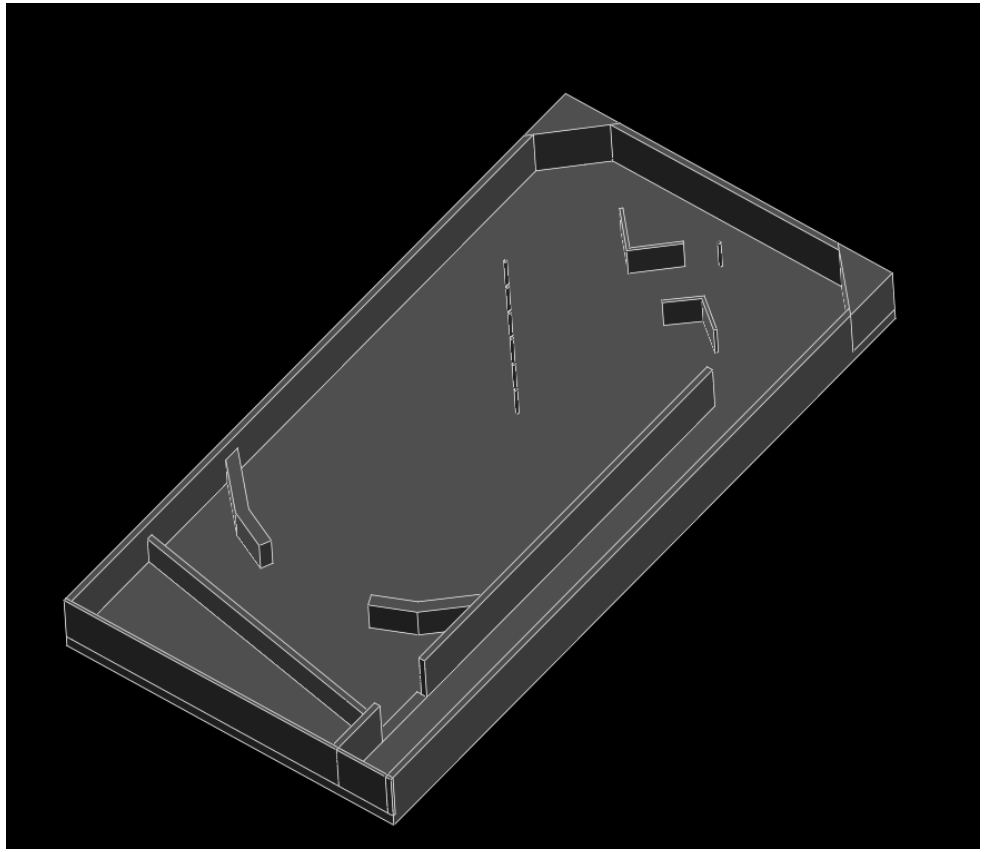
2.確認儲存的零件有開啟輪廓線

3.建立新檔，點選Link/Assemble，新增繪製好的零件。

4.組裝，給零件限制條件，平行/垂直/重和/共線等。

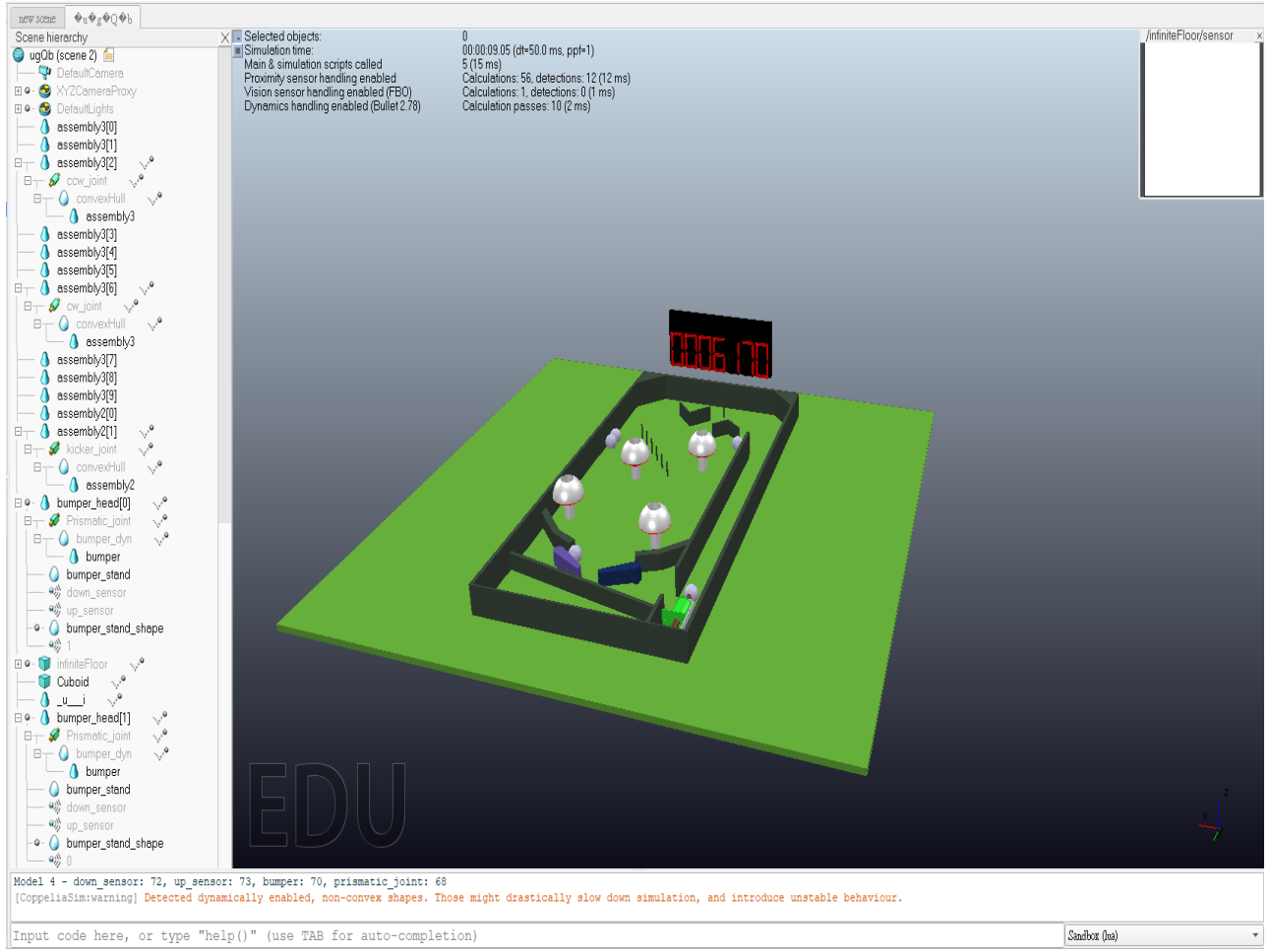
5.完成組合

彈珠台檯面設計



設計發想是參考夜市傳統彈珠台，在檯面內預留回彈計分器以及fipper、彈珠台拉桿擺放位置，預想球滾動之動向設計軌道，並利用SolveSpace繪製出來，底板:長120x寬60x高2mm，外側包圍板子:高8mm 厚度1mm，再繪製內部零件，轉成stl檔。

COPPELIASIM動態模擬



1.將檯面、擊球器、fipper等零件加入Coppeliasim

2.調整位子

3.加入回彈裝置模型及記分板

4.更改操控程式

動態模擬過程中遇到的困難

1.擊球桿會亂跑
解決方法:要單獨新增動力模型，不然與地板接觸會導致桿件擊飛。

2.球無法依照重力滾動
解決方法:讓機構簡單化，像是彈珠台不要設計過於複雜，新增板子，作為新的彈珠台底部。

3.記分板無法正常顯示
解決方法:重新複製模組，因為程式有可能會卡機

模擬程式

透過程式，讓動態模擬作動，點選Tools go，程式就會開始執行，並透過按鍵盤上的英文字母W、S操控左邊Fipper，P、L操控右邊Fipper，K、J操控擊球桿

彈射利刃.py - SciTE

File Edit Search View Tools Options Language Buffers Help

1 彈射利刃.py

```python
# 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')
slider = sim.getObject('/kicker_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(ccw, -0.5)

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

        if keyboard.is_pressed('w'):  # Move slider to -0.15 position
            print("w is pressed")
            sim.setJointTargetPosition(cw,0.5)

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

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

        if keyboard.is_pressed('k'):  # Reset slider to the original position
            print("k is pressed")
            sim.setJointTargetPosition(slider, 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()
            break

# Start the main control loop
main()
```