

電腦輔助設計與實習

彈珠檯

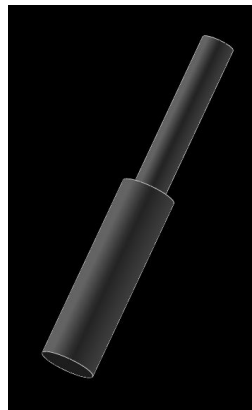
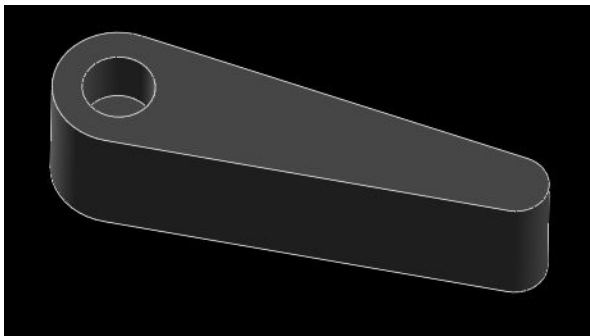
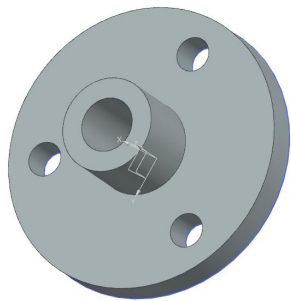
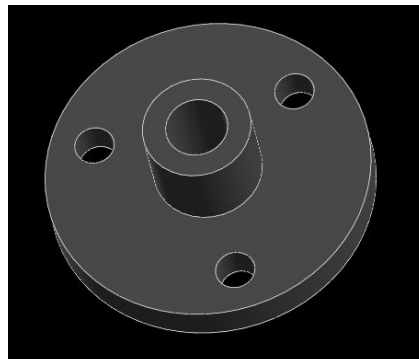
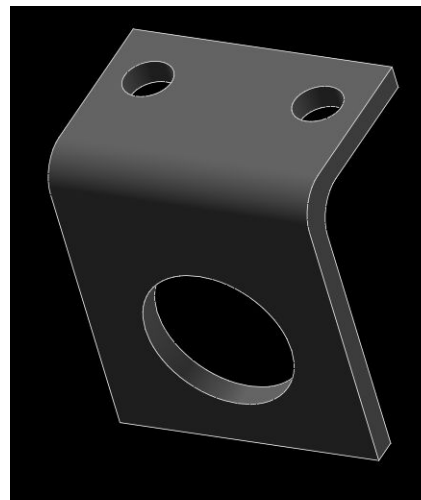
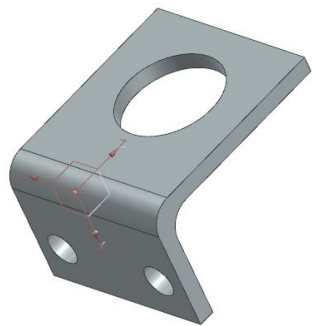
成員：

41223201 李安琪

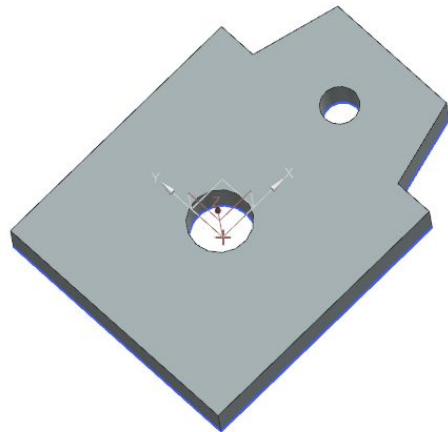
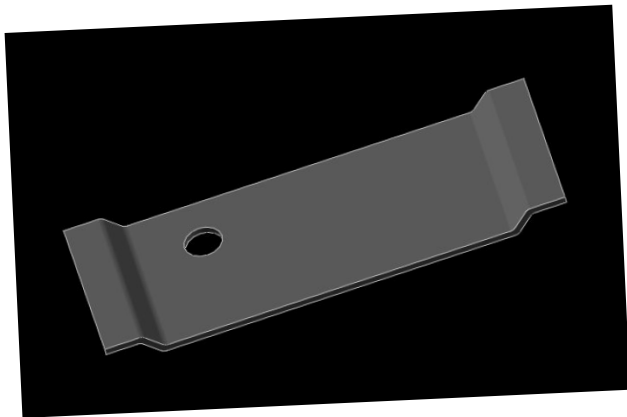
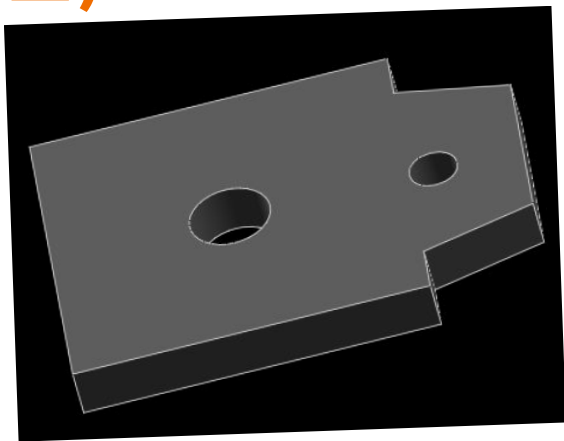
41223202 林庭暄

41223205 陳仙宸

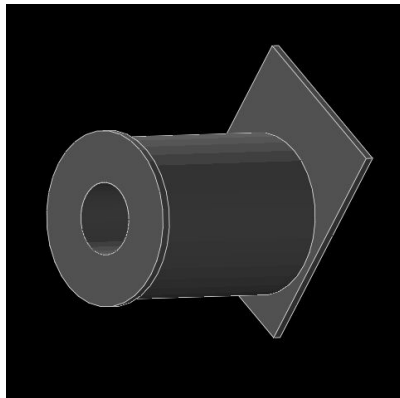
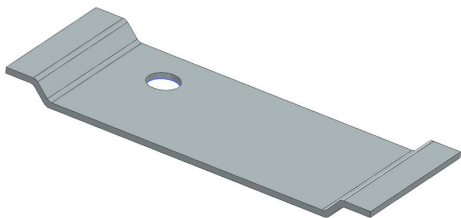
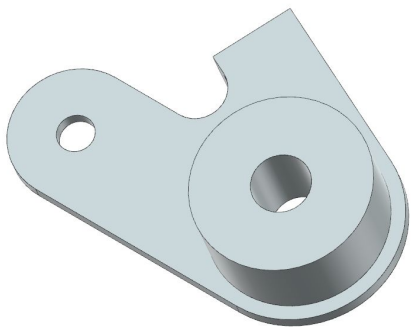
零件分工 (李安琪)



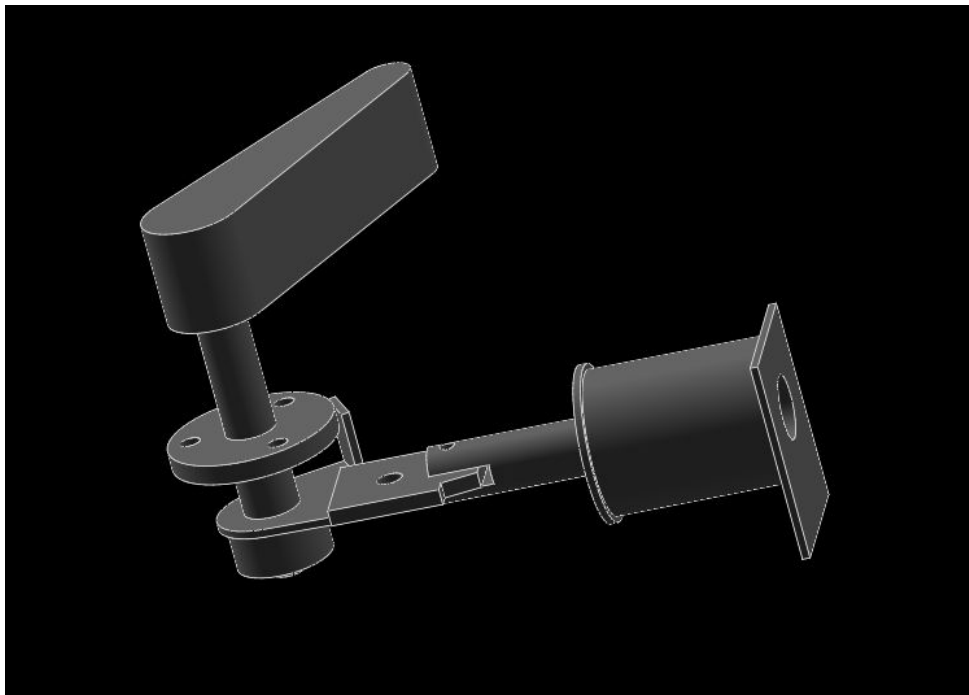
零件分工 (林庭暄)



零件分工 (陳仙宸)



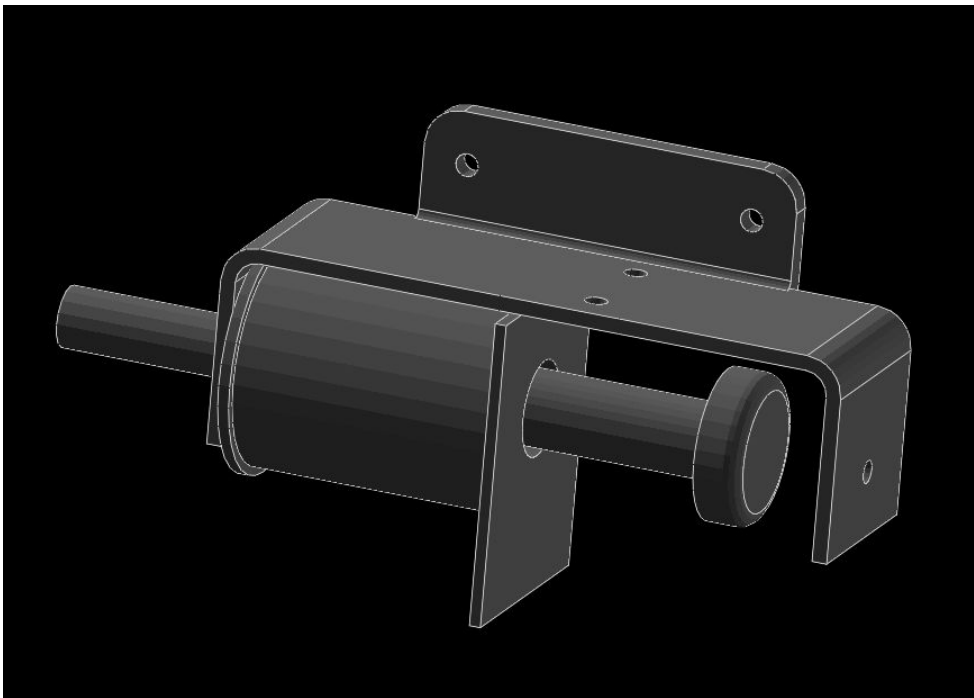
利用solvespaces導入零件開始組合



操作步驟:

- 1.開啟solvespace
- 2.匯入零件
- 3.參考老師給的檔案進行組裝
- 4.零件限制條件:平行、垂直、共線。
- 5.存檔

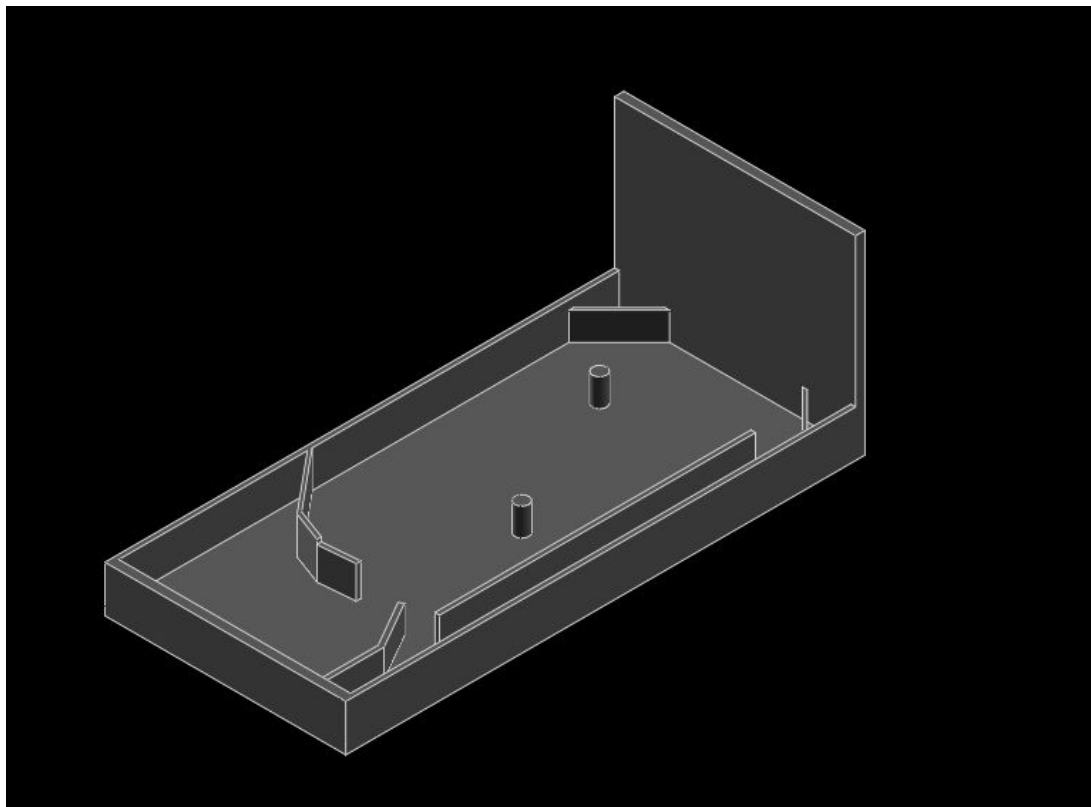
利用 solvespaces 導入零件開始組合



操作步驟:

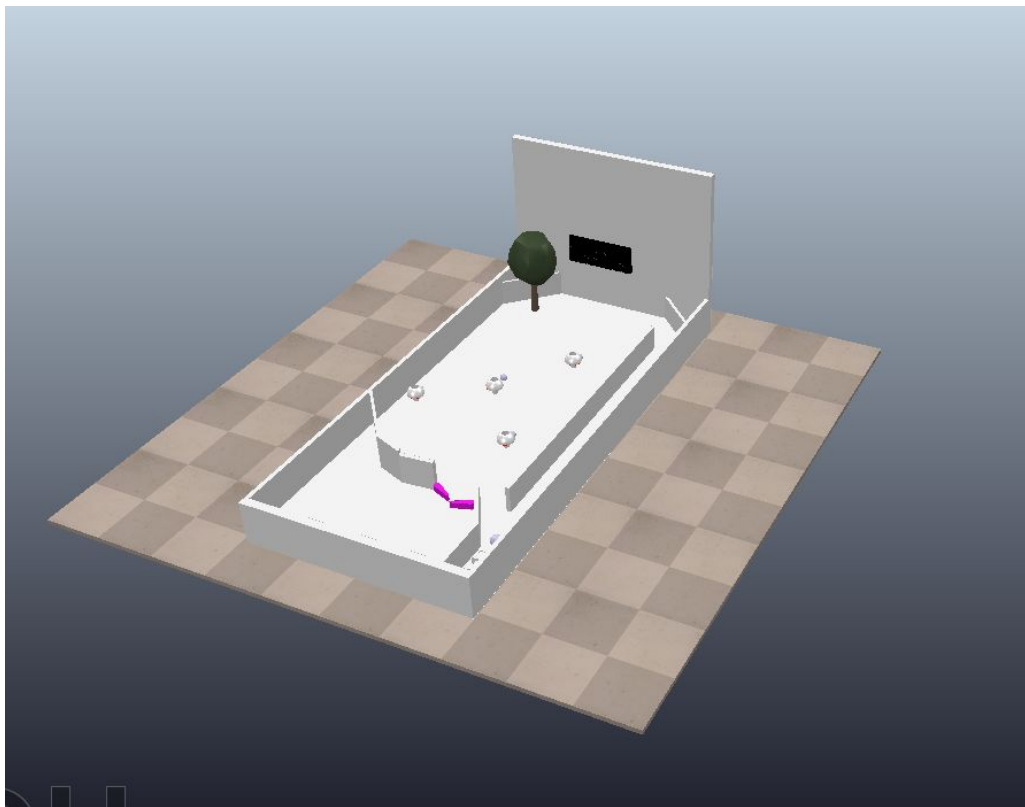
1. 開啟 solvespace
2. 匯入零件
3. 參考老師給的檔案進行組裝
4. 零件限制條件: 平行、垂直、共線。
5. 存檔

彈珠檯繪製 - solvespaces



在solvespaces畫出
寬260mm長560mm的彈珠
台底板，外圍40mm，厚
5mm，內部設計一些擋板，
讓球路線更多選擇。

彈珠檯-coppeliasim模擬



1. 打開coppeliasim
2. 刪除原本floor, 重新匯入一個floor調整成15x15
3. 加入檯面、發射器、撥桿
4. 調整位置
5. 加入回彈裝置模型及記分板
6. 進行程式模擬及修改

模擬過程中遇到的問題

- 1.發射器會彈射出去(更改座標的距離)
- 2.記分板跳不出數字(重新抓取檔案)
- 3.回彈模型沒有作用(重新抓取模型)
- 4.控制撥桿無法做動(修改Python程式)
- 5.球會飛出遊戲台(勾選Body is responsable)

模擬程式

簡單說明按鍵控制:

點選Tools go 開始運作

w 與 s 控制發射器前後

e d 與 p l 控制撥桿

最後 q 停止/回歸原始狀態

將發射器與撥桿程式結合在一起, 如右圖。

```
Plunger1.py - SciTE
File Edit Search View Tools Options Language Buffers Help

1 Plunger1.py
# pip install pyzmqswswsq 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)
slider = sim.getObject('/P_joint')
cw = sim.getObject('/Prismatic1_joint')
ccw = sim.getObject('/Prismatic2_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('w'): # Move slider to -0.15 position
            print("w is pressed")
            sim.setJointTargetPosition(slider, 0.3)

        if keyboard.is_pressed('s'): # Reset slider to the original position
            print("s is pressed")
            sim.setJointTargetPosition(slider, -0.1) # Reset to the initial position

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

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

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

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