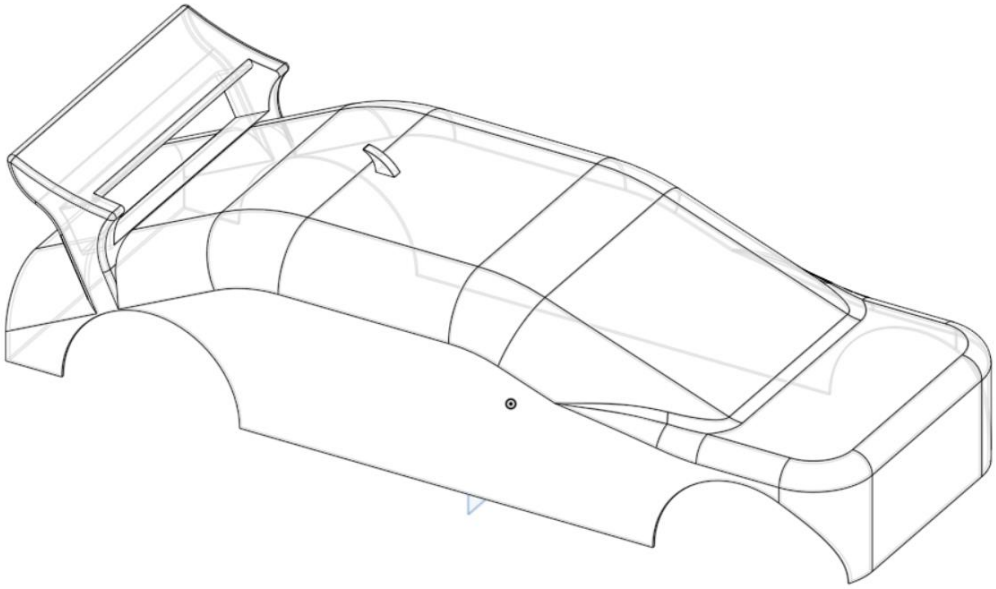
Assignment 3

Mechatronic project



第二組 AG2

40723144

40723147

40723148

40723150

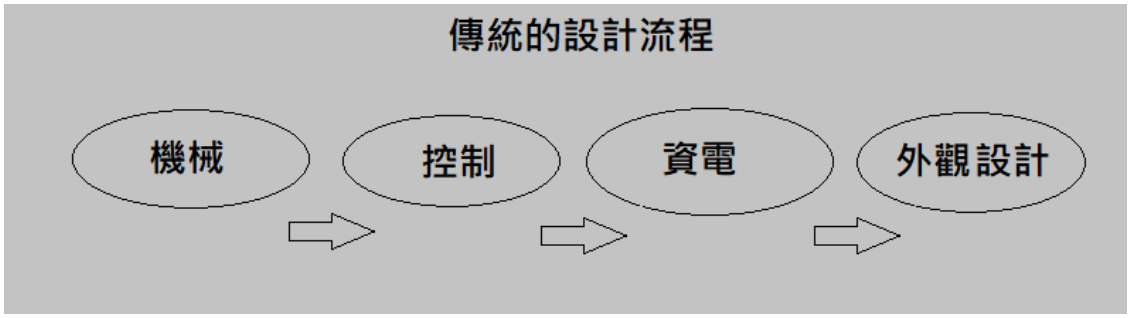
中華民國 109 年 6 月 24 日

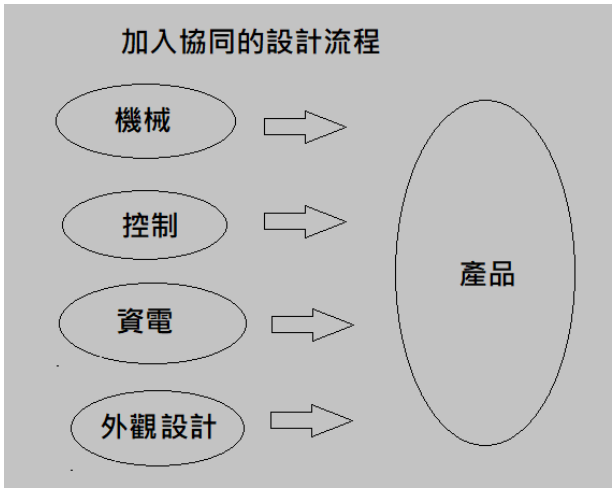
目錄

摘要

為了要提供出具現代化的產品設計流程，所以對此進行初步了解與分析・首先我們需要知道時間成本對整個設計流程的重要性，假設一個產品的誕生需要經過機構、配電、程式、電資、外觀設計等流程・若以過往的設計經驗來看需先由頭至尾一步一步經由每一位參與者的巧手下完成，在不發生不如意的情況下進行也至少需要花上好幾步的步驟來完成產品的設計・

協同產品設計可在同一個時間點將不同領域的人們聚集起來，各自發揮所長進行產品設計的流程，將以往需要一整串的步驟一一打破，且在更有效率的環境下進行溝通且完成最佳化設計・

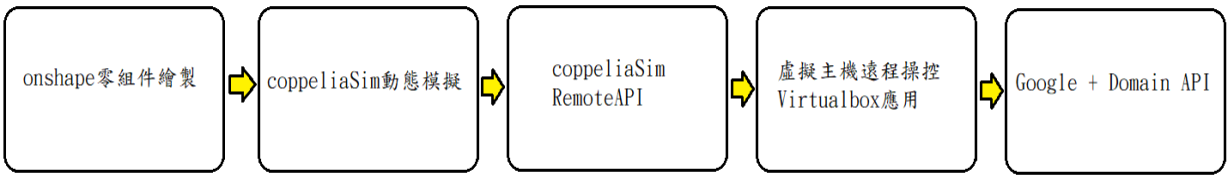




加入協同產品設計流程後，可減少時間成本的損失，故可以在更短的設計週期內完成產品設計・

專案設計過程

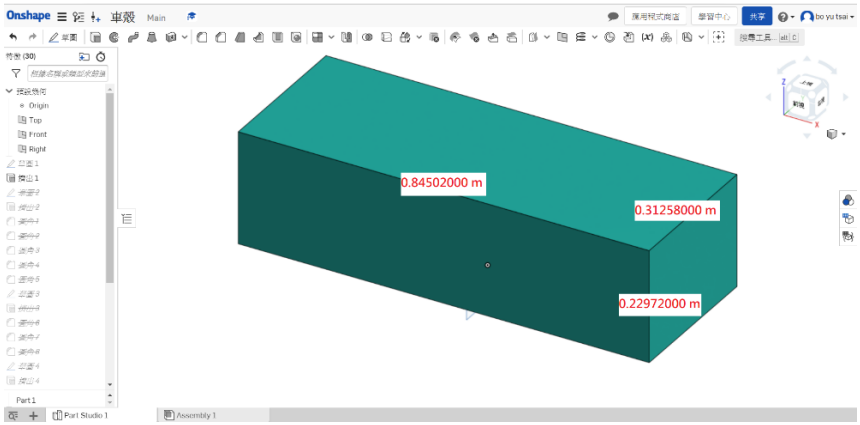
首先我們經由 onshape 建構我們的模型實體完成外觀設計與結構設計的部分，完成模型製作後再導入 coppeliaSim 中進行動態模擬與傳感器加入，藉由 coppeliaSim 的模擬得到我們想要的做動模式，最後加入虛擬主機可藉由虛擬主機達成遠端不同主機下的操控模擬。



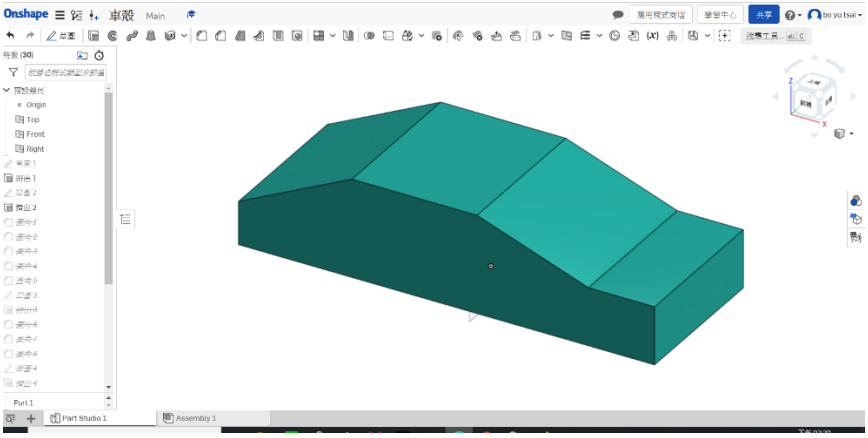
設計流程簡圖

模型設計繪製

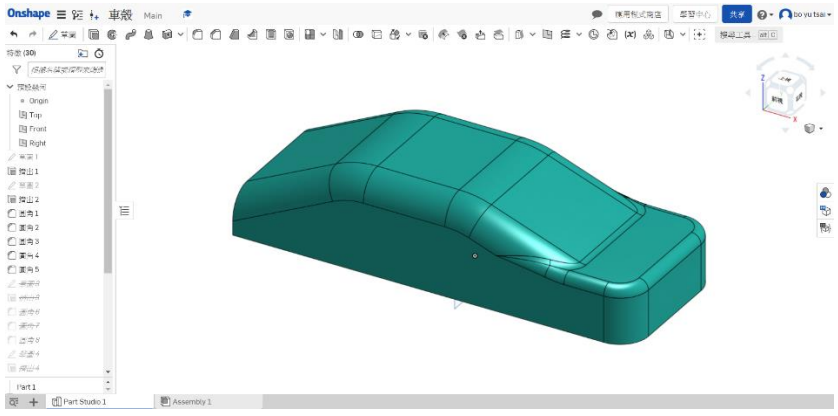
使用 onshape 來設計車殼，設計過程如下:



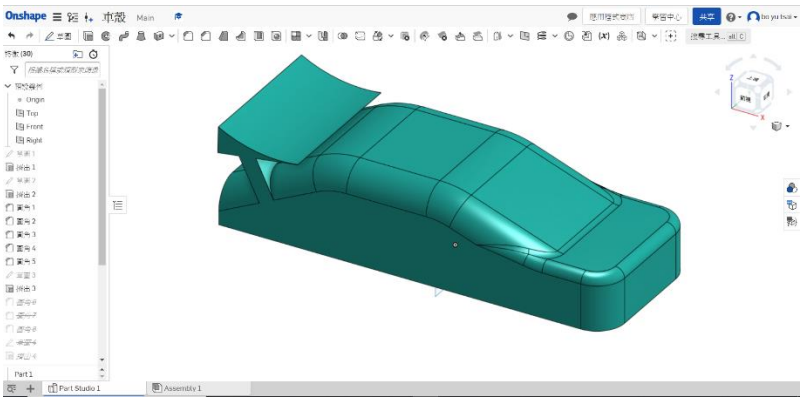
首先先了解車子的最大限度尺寸，開始規劃車子的流線型以及外觀初始構造設計。



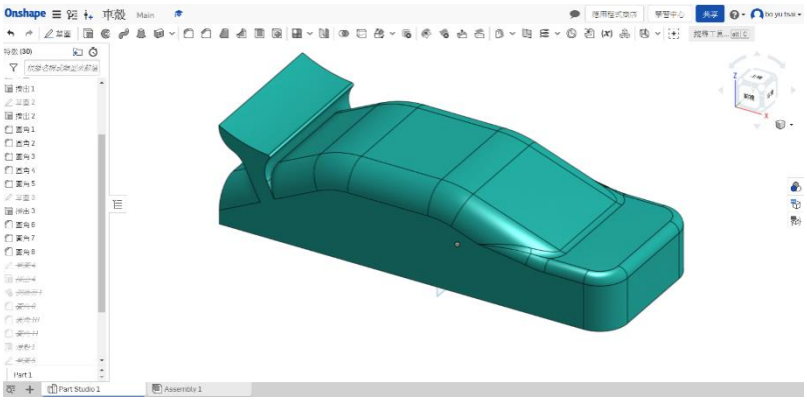
完成初步流線構造在來就是倒上精美的圓角



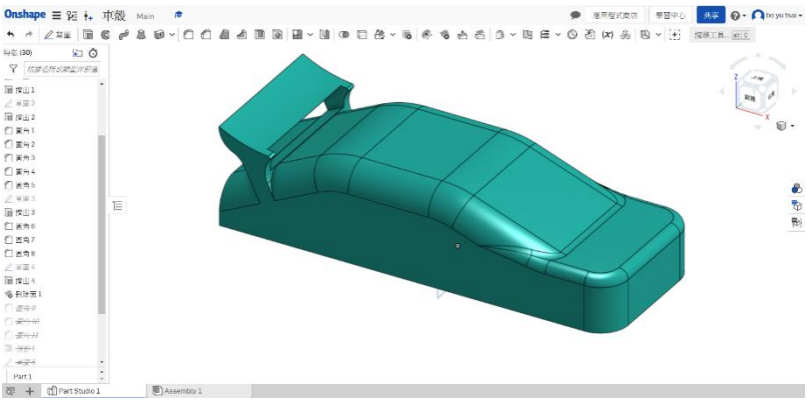
為了增加車體流線感，所以導入空力套件尾翼增加整體的線條變得更加動感吸引人。



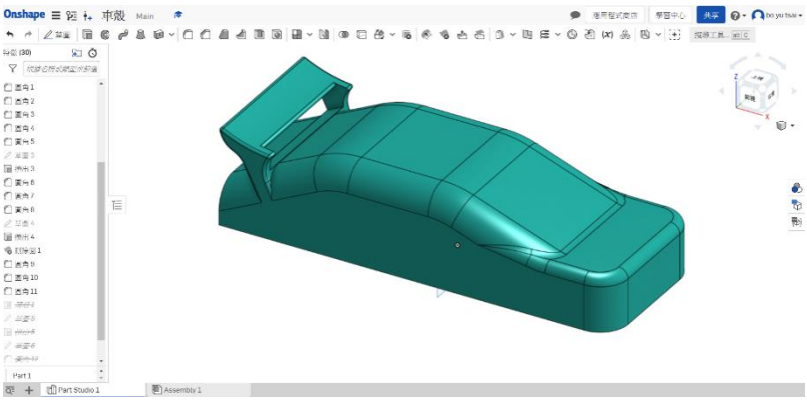
完成初步尾翼建構後再來就是要將銳利的邊邊角角導上圓角並且修飾其外觀如：



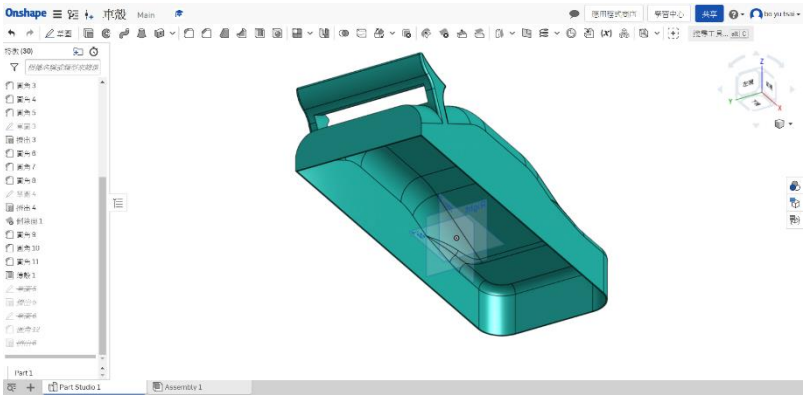
在加入一些空隙以讓流場更加優美



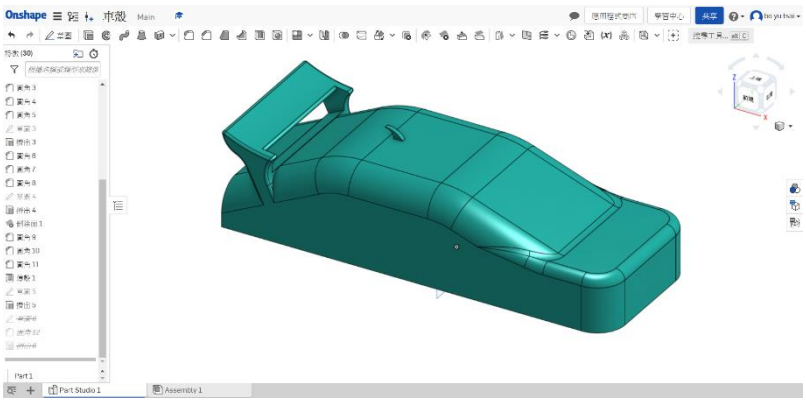
圓角修飾外觀



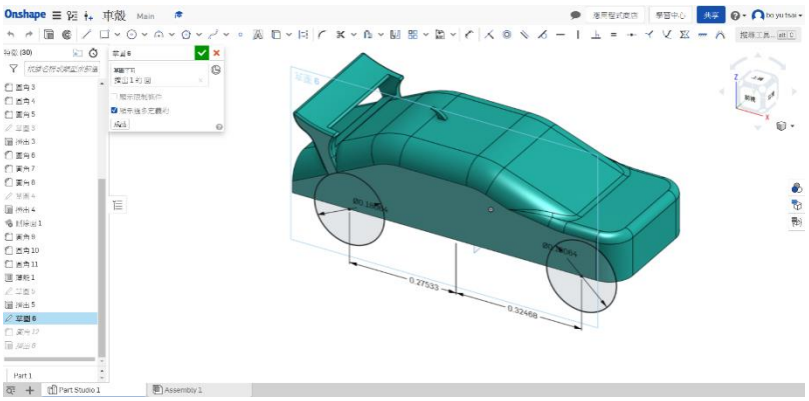
薄殼導入將車身輕薄化



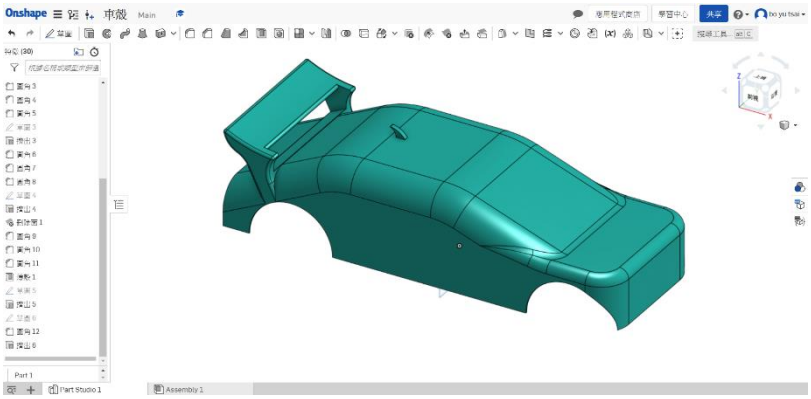
現代感鯊魚鰭設計，只是為了整體美觀。



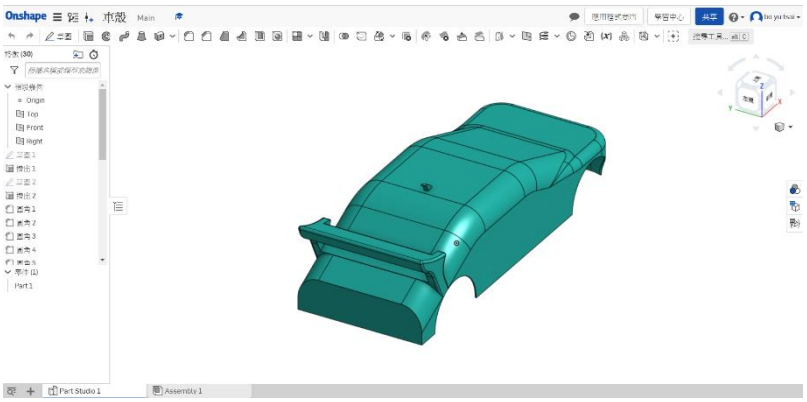
量測車軸間的距離並給定



將兩個輪子的定位挖出來



最後展示完整版車型

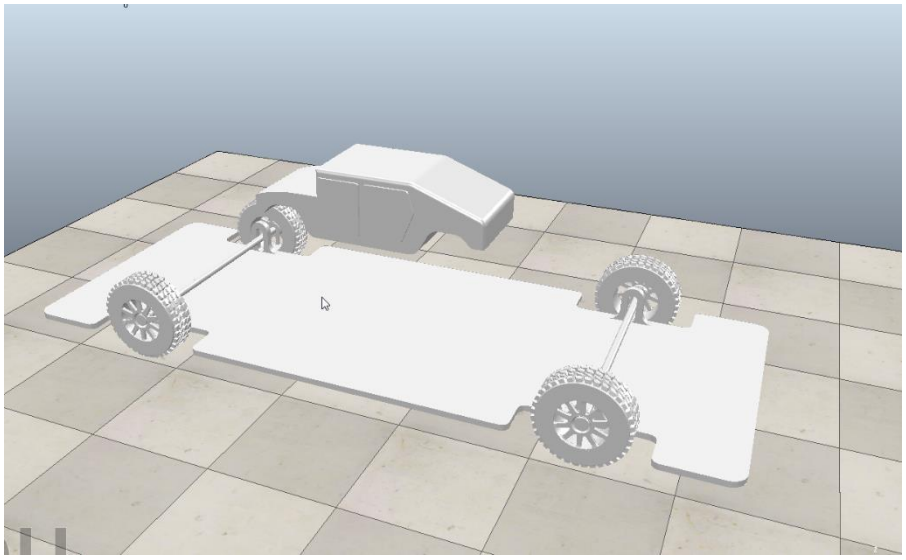


CoppeliaSim

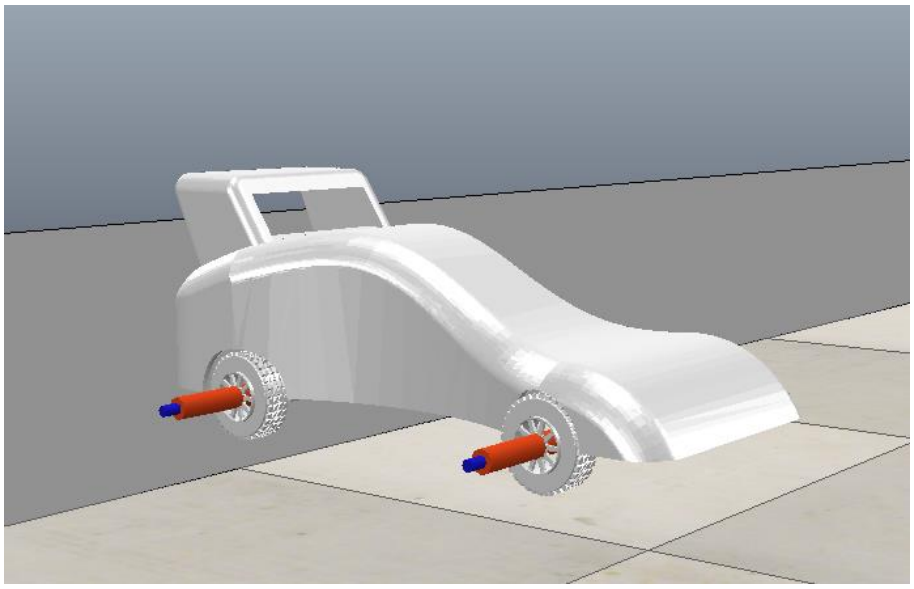
本學期的課程目標是希望利用 onshape 協同設計出一台四輪小車，且小車中包含了方向盤、差速器、避震器…等零件，再導入 CoppeliaSim 進行模擬

簡易模擬

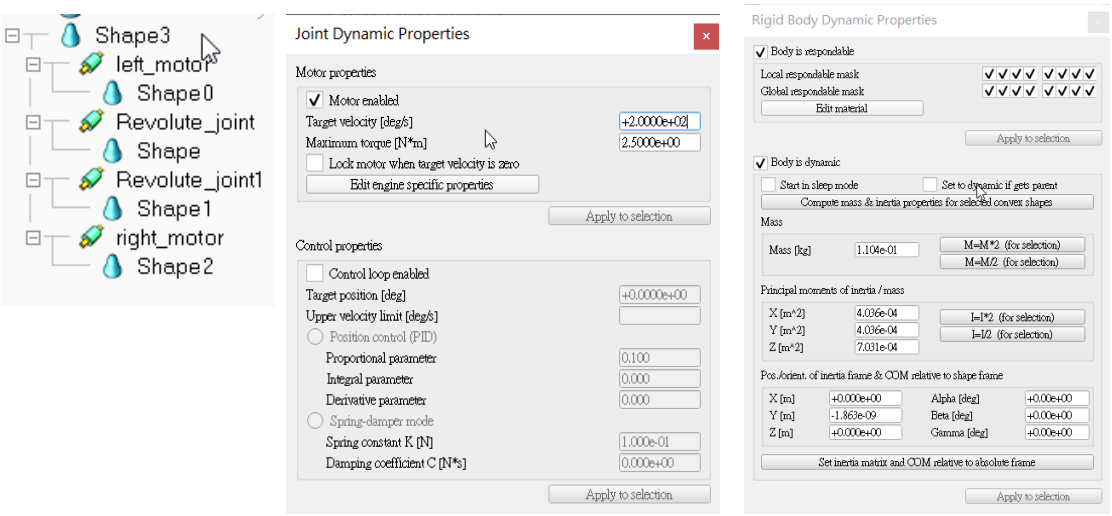
在剛導入模型時就發現零件因尺寸無法配合的問題(公英制混用)



故零時採用由 40723147 同學所繪製的簡易小車

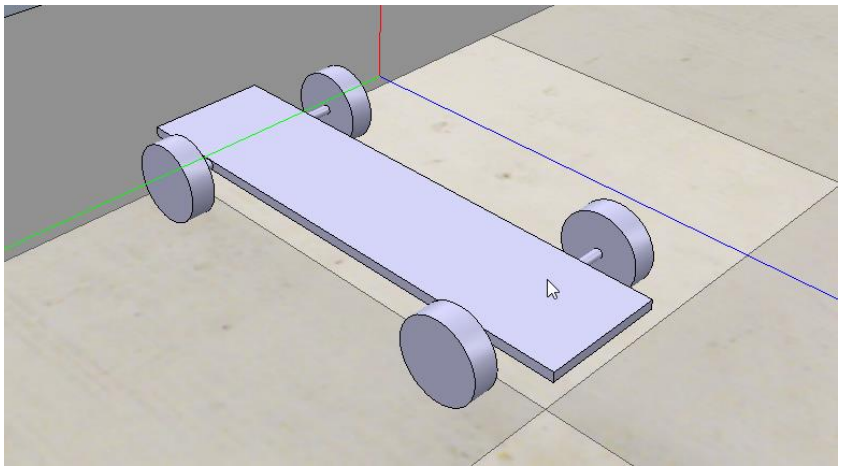


再將模型導入 CoppeliaSim 並進行設定

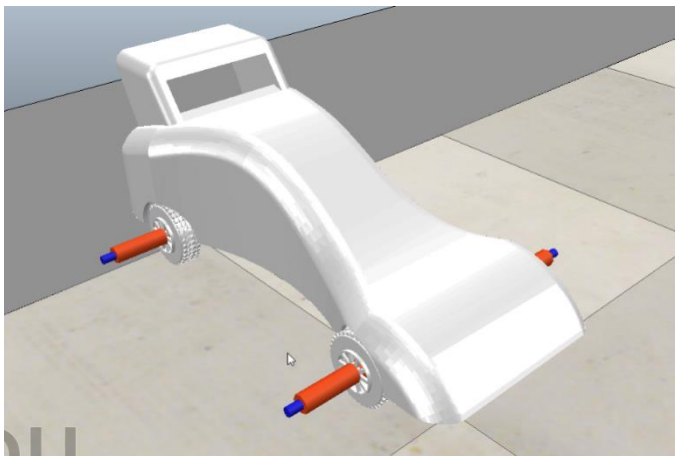


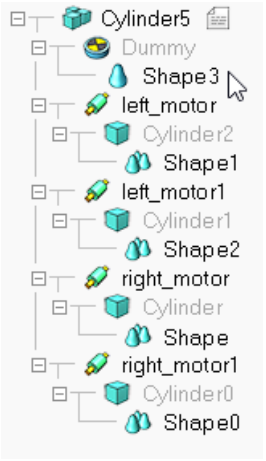
設定完成後進行了簡單的直線運動模擬，在直線運動時小車成功的移動了，但在導入我們想要導入腳本對小車進行控制時，發現小車無法操控，之後經過詢問老師後，了解到對 CoppeliaSim 來說我們導入的 STL 只是一個皮膚，我們不能直接的對皮膚進行控制，所以重新進行了貼皮的動作。

首先，先給予小車簡單的實體骨架:



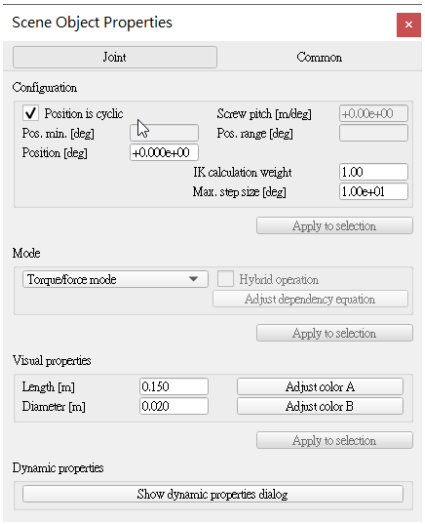
接著貼上皮膚加入馬達，並重新整理子母關係:

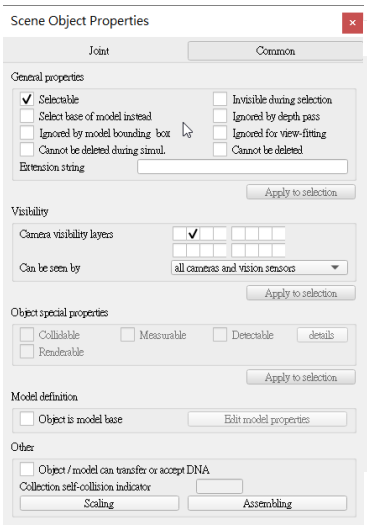


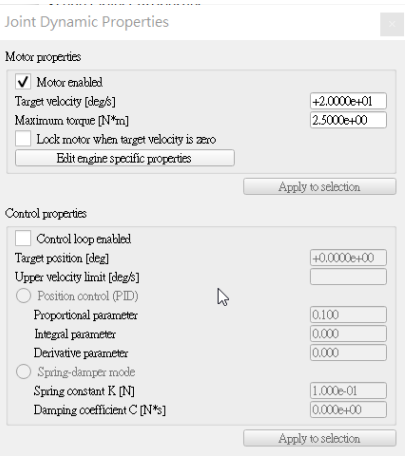


接著進行參數設定

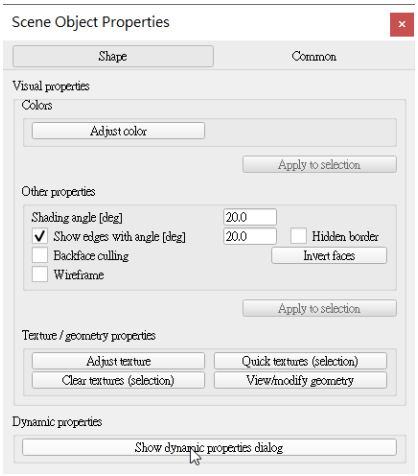
1. 馬達

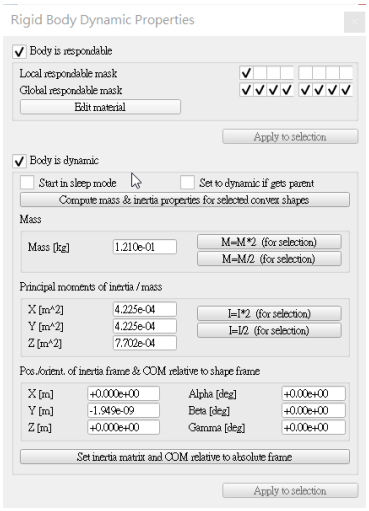


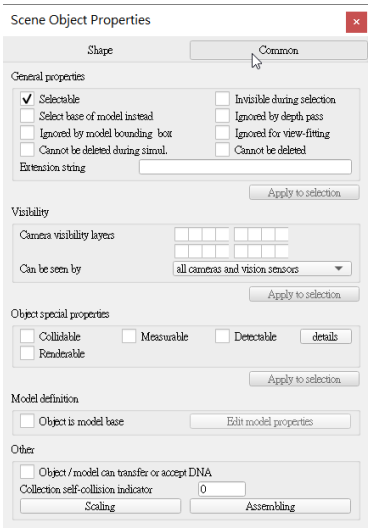


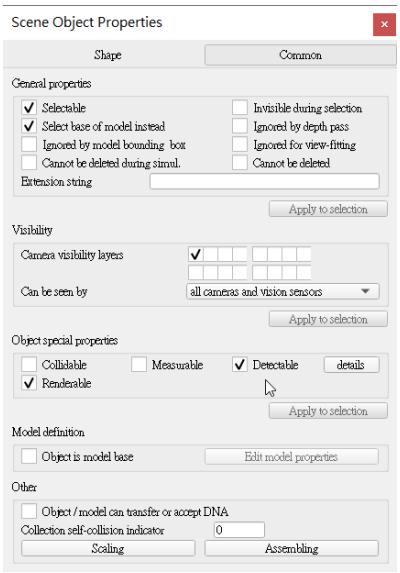


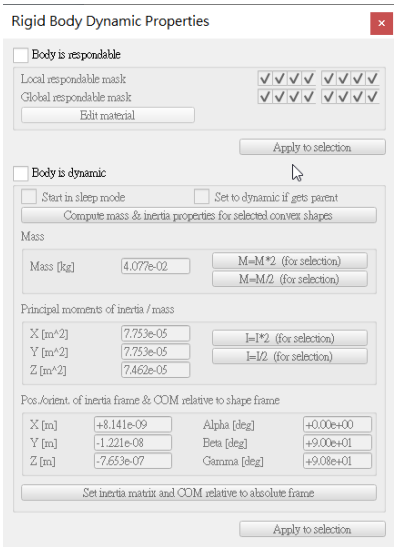
2. 車輪(實+虛)



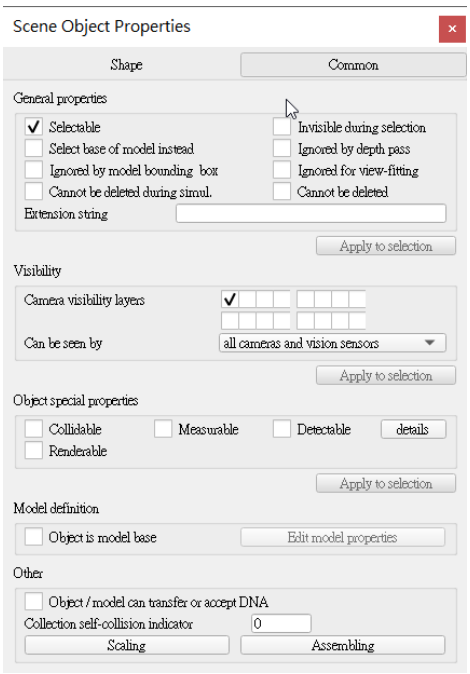


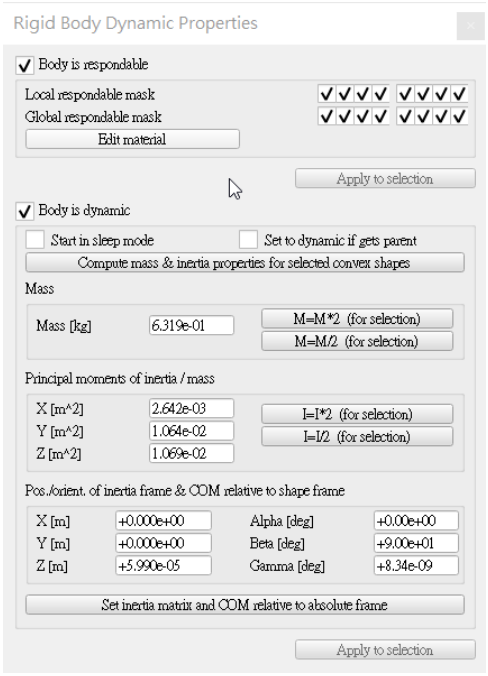


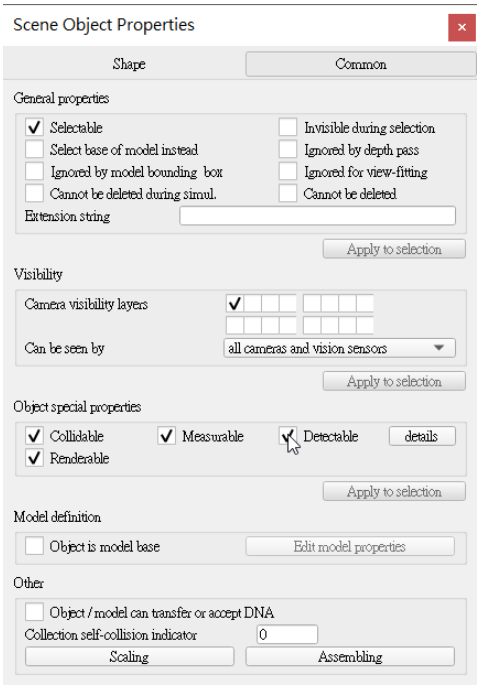


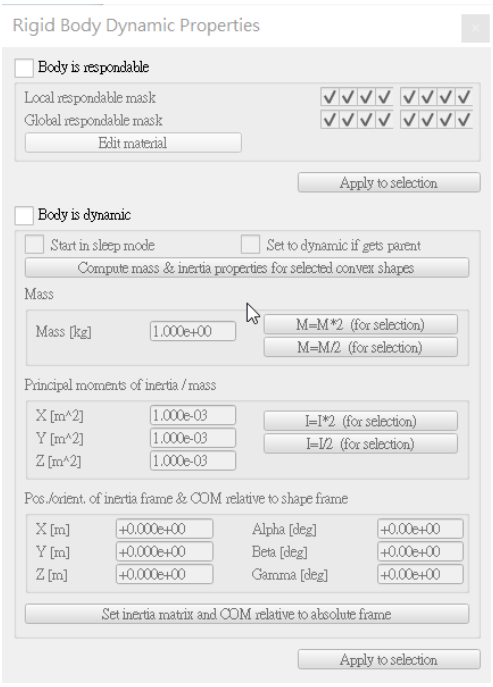


3. 車身(實+虛)





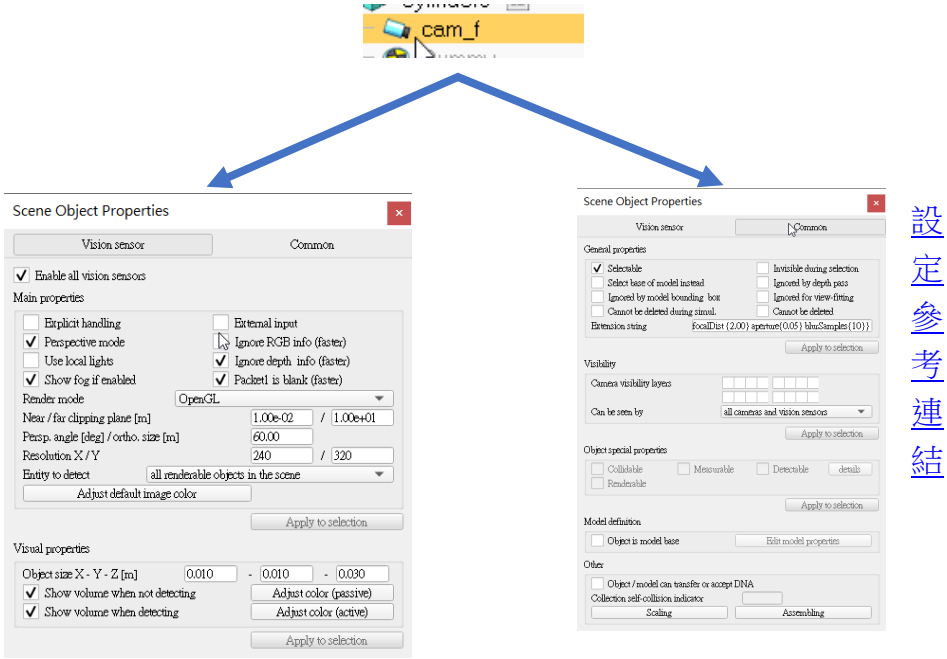




接著設定 camera


```
simRemoteApi.start(19999)
```

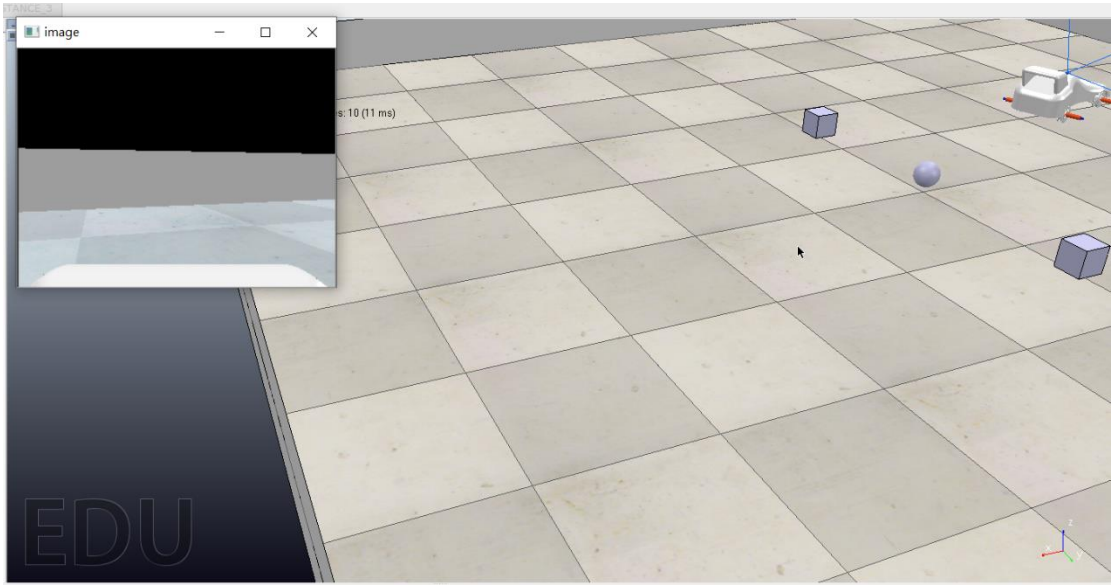
在小車加入 sensor(命名為 cam_f)並進行設定



[設定參考連結]

設定完成後可導入 VirtualBox 中測試 RemoteAPI

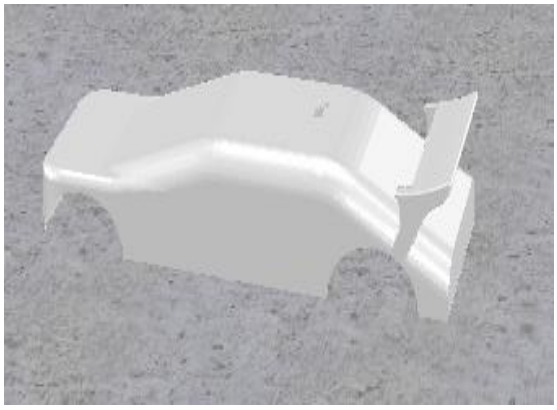
使用老師以製作好的工具，用 sciTE 執行 cae_model.py(要先把 VirtualBox 裡面的 CoppeliaSim 開啟並讓小車開始模擬)，如果成功了會出現以下畫面



測試影片

接著後面我們試著把老師的車子套用我們的皮膚上去

包括車體、輪胎、輪鼓和避震



車體

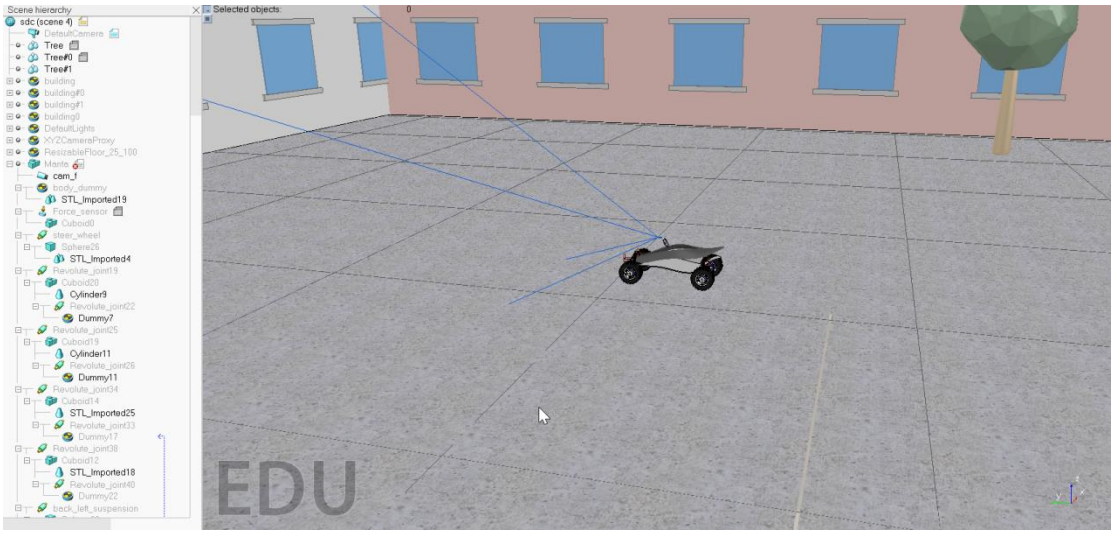
輪胎

輪鼓

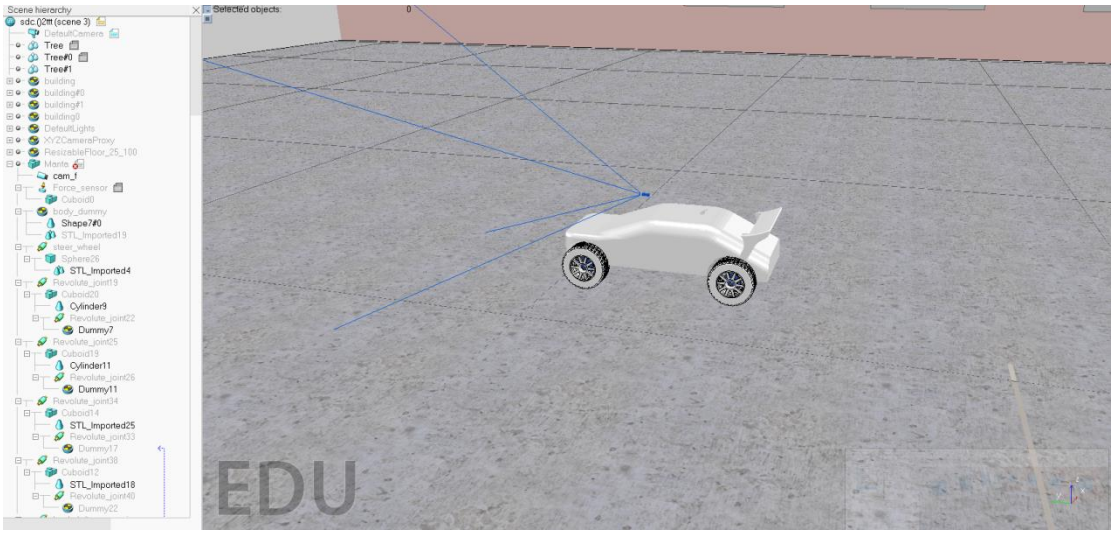
避震器

以下是兩張圖的比較

原圖:



套用後的圖:



[測試影片 2]

VirtualBox

透過 VirtualBox 來模擬透過其他電腦利用網路來控制 CoppeliaSim 裡的小車模擬，為了能讓小車可以順利的透過網路控制，所以要先完成 VirtualBox 的網路設定及連線設定。

網路設定

網路設定又分為 IPv4 和 IPv6 兩種網路協定版本，因此在不同的網路協定下設定的內容也會有所差異。底下會詳細的解說電腦連上(NAT)虛擬主機所用到兩種協定下的設定流程。

IPv4

現今大部分的網站採此種網路協定。網路設定較 IPv6 簡單些。
在 VirtualBox→Files→Preferences→Network
先新增一個 NAT 的網路連接方式，選取支援 DHCP(圖 1)。
虛擬主機(cd2020pj1)網路連線模式選擇我們剛剛新建的 NatNetwork(圖 2)。

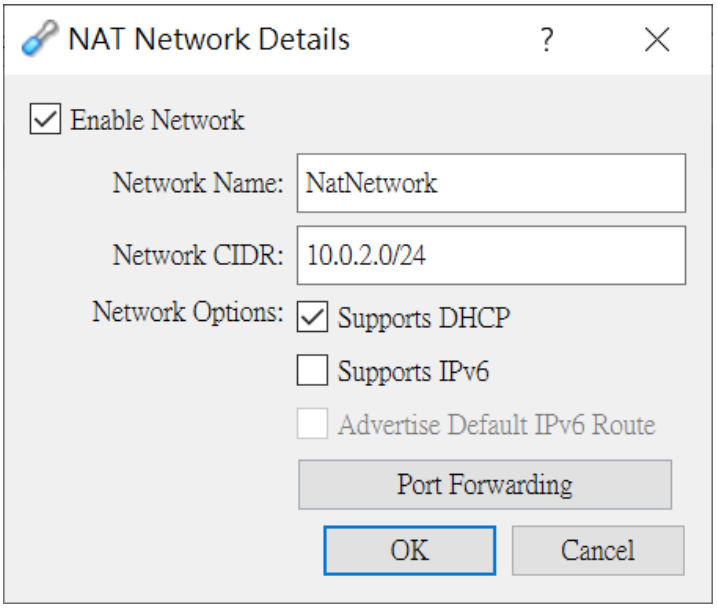


圖 1 IPv4_NAT 新建

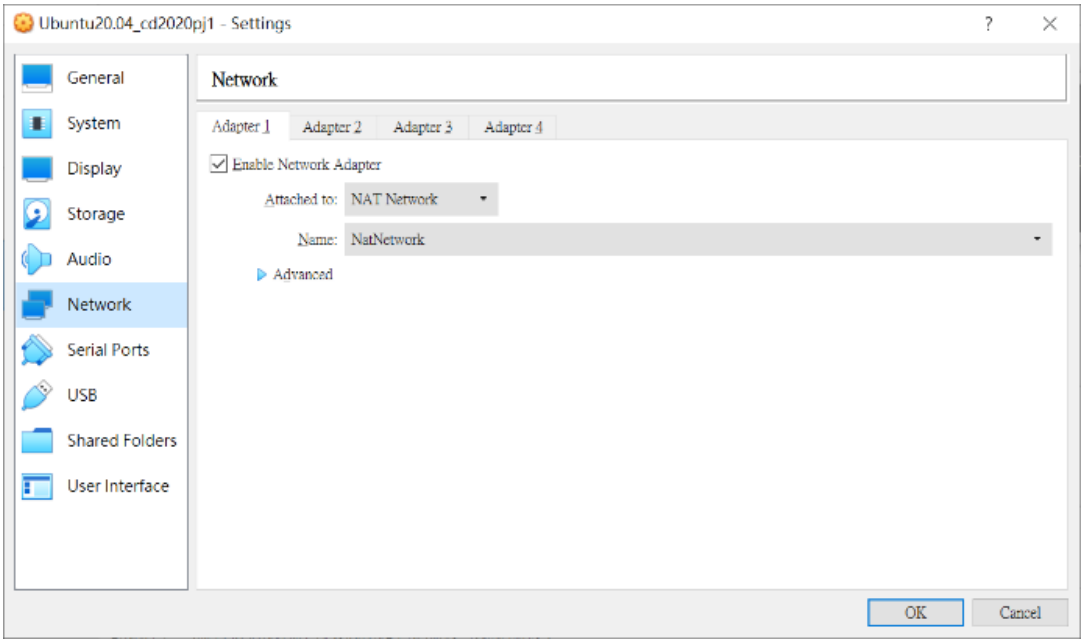


圖 2 IPv4 網路連線型式

IP 查詢

利用 ifconfig 指令查詢虛擬主機的 IP 位置。

```
ifconfig
```

若跳出尚未安裝 net-tools，請執行以下指令安裝。

```
sudo apt install net-tools
```

將查到的 IP(圖 3)填入 Guest IP(圖 4)。

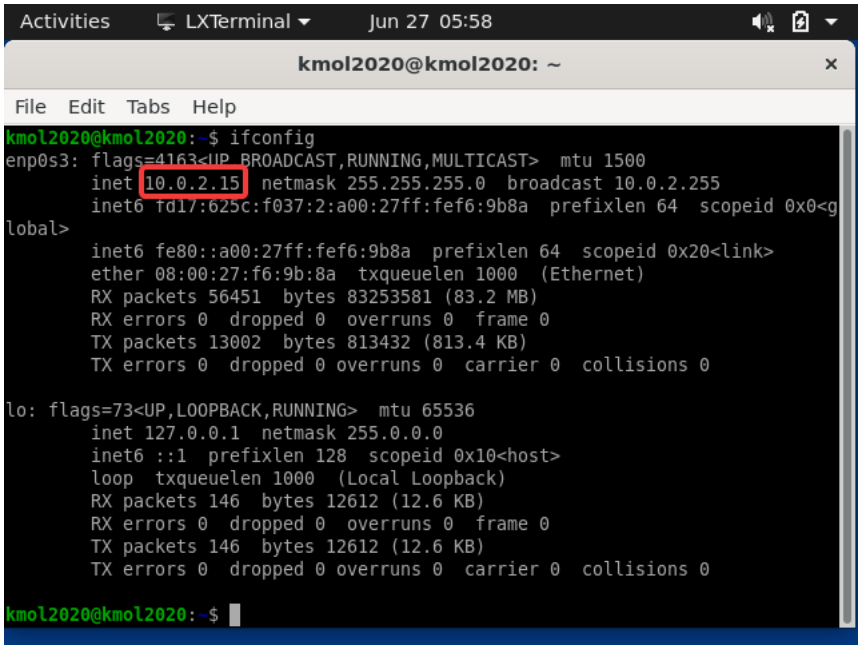


圖 3 IPv4 _ifconfig

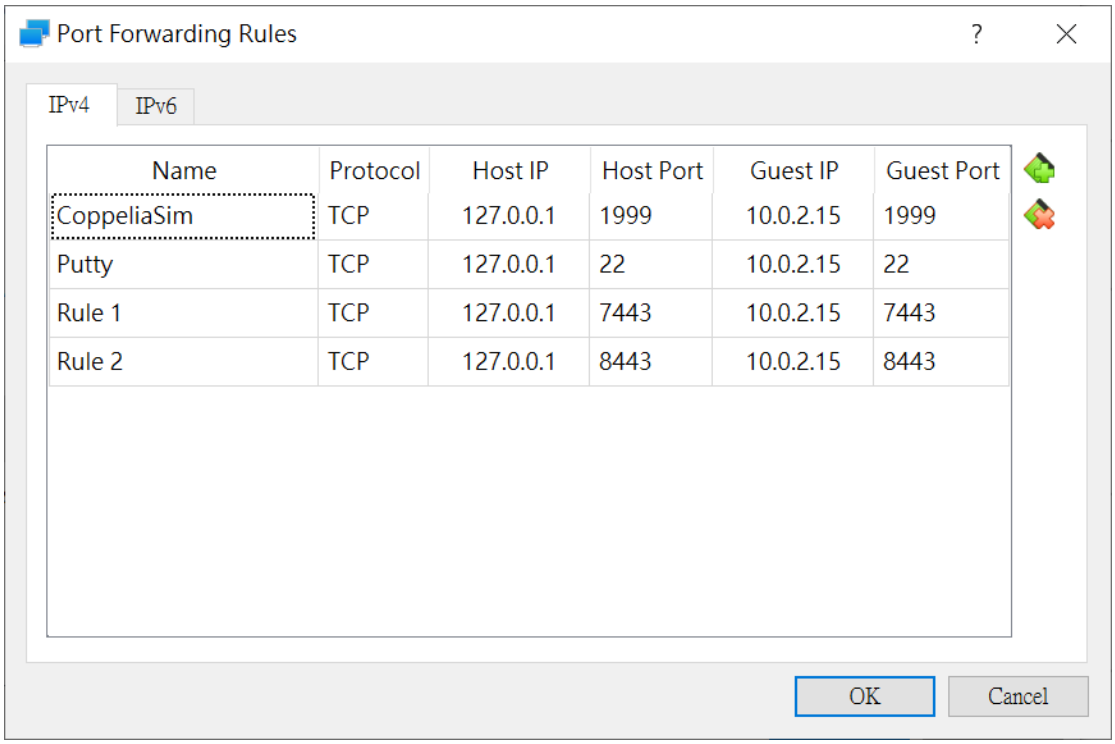


圖 4 IPv4_Guest IP

IPv6 (有 proxy)

在 VirtualBox→Files→Preferences→Network
先新增一個 NAT 的網路連接方式，選取支援 DHCP、IPv6(圖 5)。
虛擬主機(cd2020pj1)網路連線模式選擇我們剛剛新建的 NatNetwork(圖 2)。

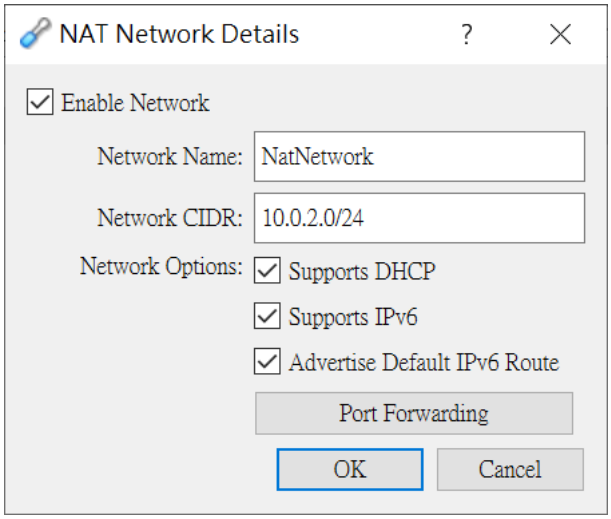


圖 5 IPv6_新建

DHCP 設定

到 /etc/netplan 目錄下 00-installer-config.yaml 檔案內的設定。

```
cd /etc/netplan
sudo vi 00-installer-config.yaml
```

00-installer-config.yaml 的檔案內容

```
# This is the network config written by 'subiquity'
network:
    ethernets:
        enpes3:
            dhcp4: true
            dhcp6: true
            nameservers:
                addresses:
                    - 2001:b000:168::1
    version: 2
```

編輯完成後:wq 跳出編輯並儲存，並更新 netplan 設定。

更新 netplan 設定

```
sudo netplan apply
```

Proxy 設定

到 /etc/apt/apt.conf.d 目錄下檢查是否有 proxy.conf 檔案，若沒有這個檔案，請新增(直接用 vi 編輯器開啟 proxy.conf 就會自動開啟/新增)。

```
cd /etc/apt/apt.conf.d
sudo vi proxy.conf
```

proxy.conf 的檔案內容

```
Acquire::http::proxy "http://[2001:288:6004:17::填port號]:3128";
```

Port 號請填寫當前所使用的 proxy 的 port 號。

完成編輯後:wq 跳出編輯並儲存，並更新 apt 設定(proxy)。

```
sudo apt update
```

IP 查詢

利用 ifconfig 指令查詢虛擬主機的 IP 位置。

```
ifconfig
```

若跳出尚未安裝 net-tools，請執行以下指令安裝。

```
sudo apt install net-tools
```

將查到的 IP(圖 6)填入 Guest IP(圖 7)，若要使用老師分配的 IPv6 位置就將 ::1 替換成被分配到 IPv6 位置。

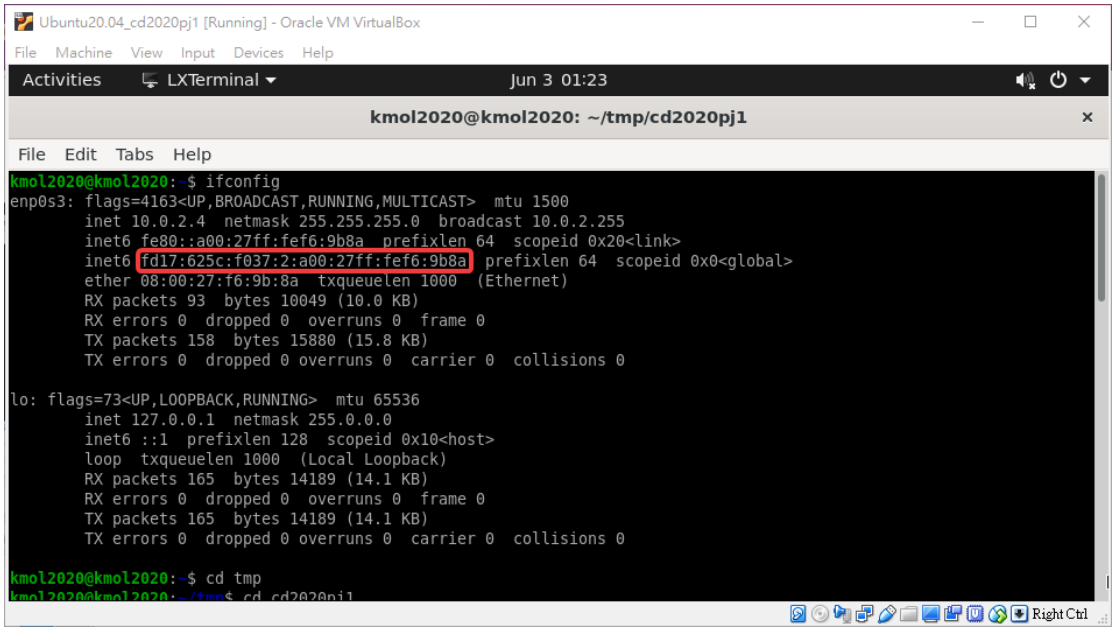


圖 6 IPv6_ifconfig

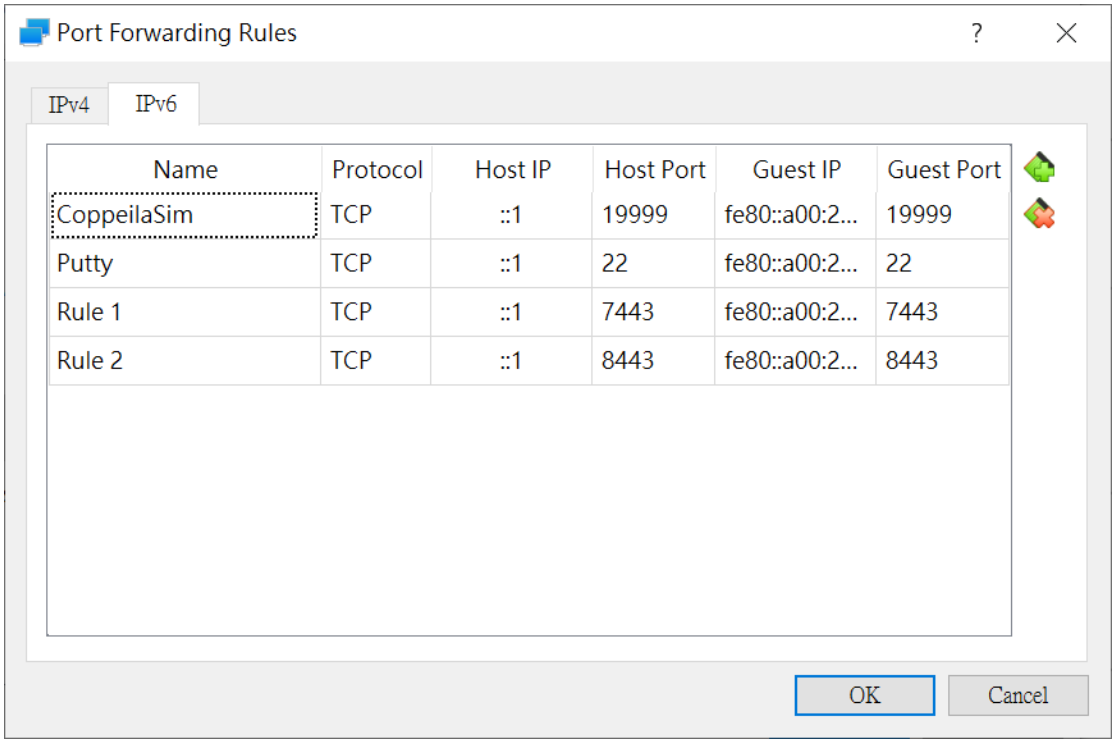


圖 7 IPv6_Guest IP

連線設定

設定 putty、Xming、nautilus 和一些啟動 CMS 所需要的模組。

Putty 連線設定

IPv4

Host Name：localhost
Port：22
Connection type：SSH

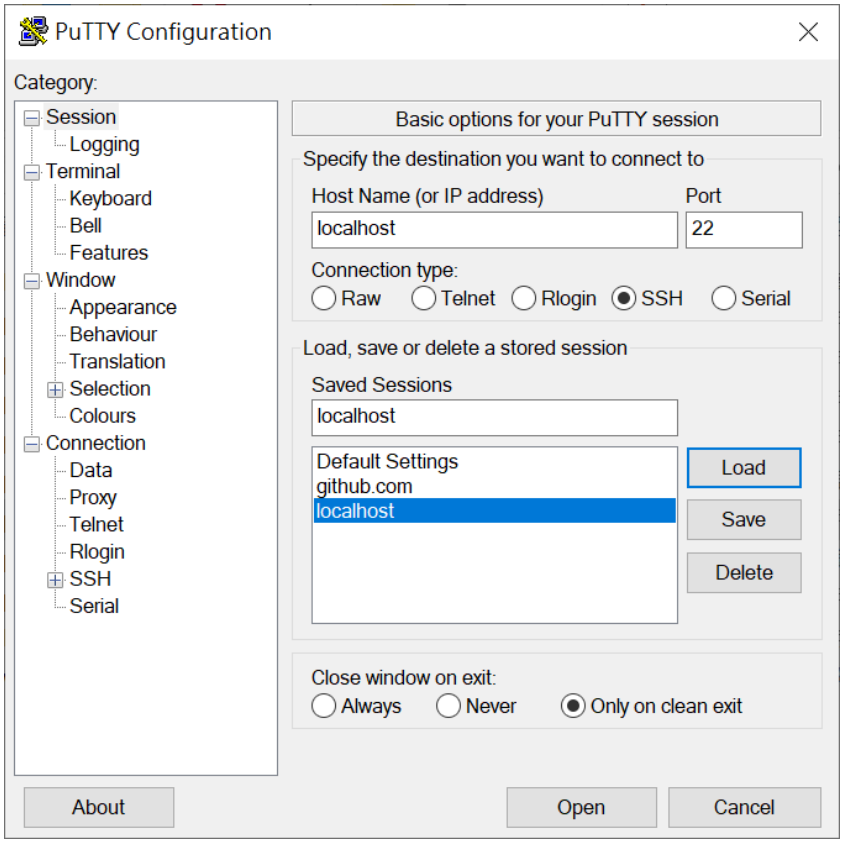


圖 8putty_IPv4 Session 設定

IPv6

Host Name：::1

Port：22

Connection type：SSH

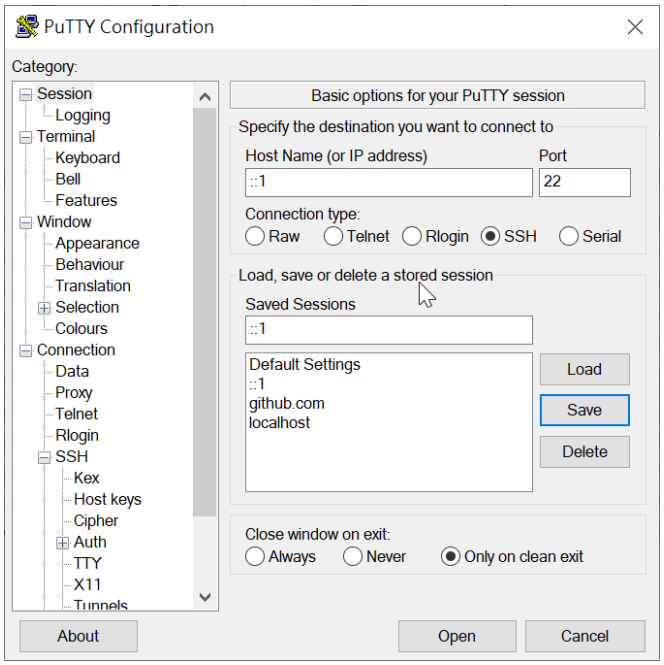


圖 9putty_IPv6Session 設定

X11(IPv4 或 IPv6 設定都一樣)

▣ Enable X11 forwarding

X display location：localhost:0.0

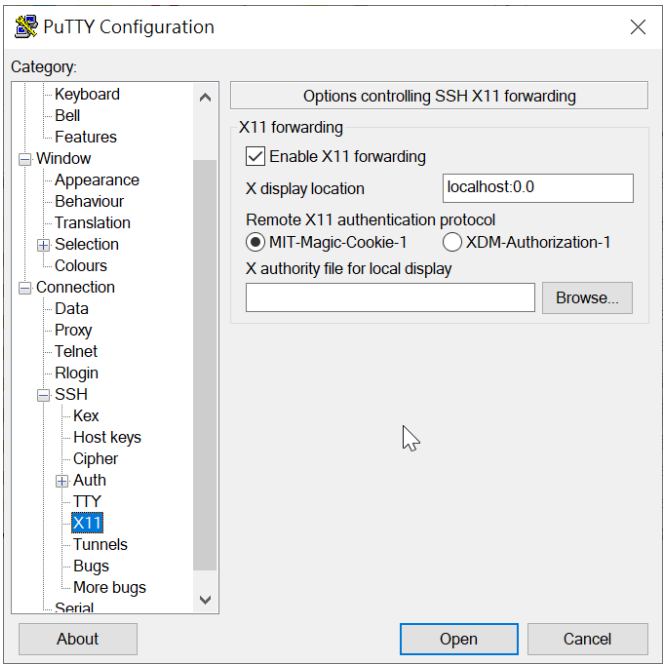


圖 10putty_X11 Session 設定

Xming

安裝 Xming，直接安裝後搬移至可攜，或是直接安裝在隨身碟成可攜(圖 11)。

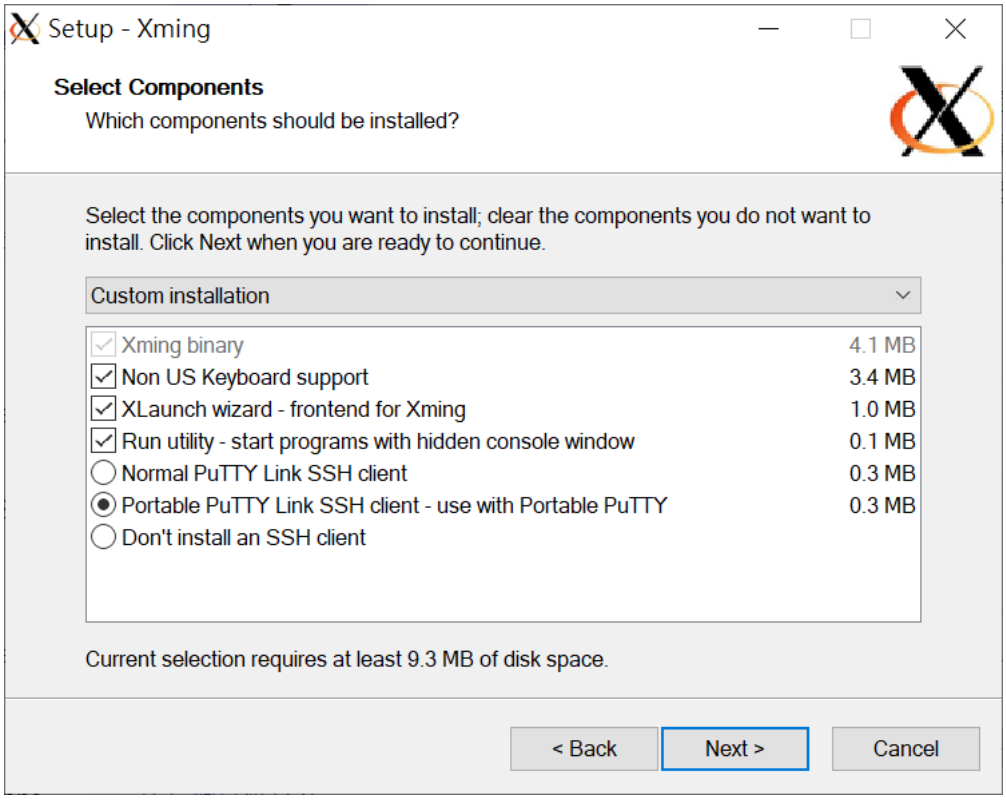


圖 11Xming_install

File Manager

在開啟 putty 後要開啟 Ubuntu 的 File Manager 除了要安裝 Xming 之外還需要安裝 nautilus 才能開啟。

安裝 nautilus 指令

```
sudo apt install nautilus
```

在舊版的 cd2020pj1 更新後，啟動 wsgi.py 時會因為缺少 `pydrive` 而跳出錯誤，所以要補安裝。

```
pip3 install pydrive
```

若在成功啟動後，網頁還是出現錯誤，可以嘗試去更新 `mako`，更新 `mako` 前驅要先到 `python3` 的模組目錄下將 `mako` 移除，再安裝 `1.1.3` 版本。

```
cd /usr/lib/python3/dist-packages
sudo re -r mako
sudo pip3 install mako==1.1.3
```

安裝完成後嘗試重新啟動 `wsgi.py`，看是否能正常連上網頁。

使用指定網路位址連線

需要先在虛擬主機裡測試是否能成功開啟 wsgi.py 並用網頁連上，在測試用電腦和虛擬主機對連。電腦與虛擬主機連線前需要設定網路的網卡設定，網卡設定是在當前有對外網路連線功能的網卡進行設定，對該張網卡案右鍵，內容，IPv4 和 IPv6 網路協定只需要使用 IPv6 的網路協定就可以。選到 IPv6，內容(或快速點擊 IPv6 選項兩下即可開啟”內容”)。使用指定的 IPv6 位址 DNS(圖 12)。

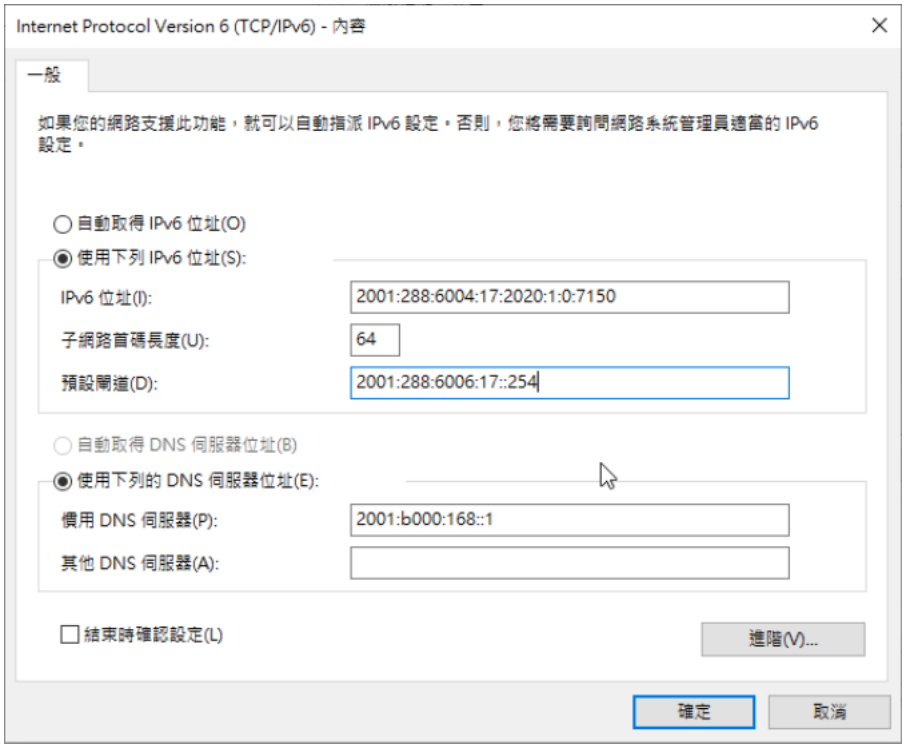


圖 12 網卡設定_IPV6 和 DNS

設定好後，需要將防火牆關閉才能成功連到用虛擬主機開的 wsgi.py(啟動的虛擬主機和連線的電腦同一台才需要設定網卡和關閉防火牆)。

檔案上傳

為了上傳 ttt 檔到虛擬主機的 pj1 專案目錄下並遠端控制小車移動，在 static 的目錄下 uploadform.js 的資料格式增加 ttt 格式。

```
/*
設定可以上傳的檔案副檔名
呼叫執行 axuploader.js, 將檔案以批量直接存至伺服器硬碟
嘗試同時將檔案透過 Google Drive API 存至對應的雲端硬碟
用於將各上傳檔案名稱存入資料庫中, 檔案刪除時從資料庫中移除
*/
function sendToServer(files){
    var req = new XMLHttpRequest();
    var result = document.getElementById('result');
    req.onreadystatechange = function()
    {
      if(this.readyState == 4 && this.status == 200) {
        result.innerHTML = this.responseText;
      } else {
        result.innerHTML = "working...";
      }
    }

    // 利用 POST 將上傳檔案名稱數列送至 server, 準備存入資料庫
    req.open('POST', '/saveToDB', true);
    req.setRequestHeader('content-type', 'application/x-www-form-urlencoded;charset=UTF-8');
    req.send("files=" +  files);
}

$(document).ready(function(){
$('.prova').axuploader({url:'fileaxupload', allowExt:
['jpg','png','gif','7z','pdf','zip','flv','stl','swf','ttt'],
finish:function(x,files)
    {
        // 這裡要利用 sendToServer 函式將 files 數列傳到 server, 再由 python 納入資料庫
        sendToServer(files);
        alert('All files have been uploaded: '+files);

    },
enable:true,
remotePath:function(){
return 'downloads/';
}
});
});
```

CMS

CMS 是一種網站後台的管理系統(也是一種內容管理系統)，我們要將 CMS、Google API 和 Oauth2 結合，允許相同 Domain 下的帳號連進網頁。

Google API 和 Oauth2

- 使用 Google + Domain API 為@ gm.nfu.edu.tw 用戶設置登錄名
  https://console.developers.google.com
- 設置 OAuth 2.0 客戶端 ID
  使用 https://github.com/authomatic/authomatic 允許用戶使用 Google 或 Github 帳戶登錄。
- 獲取您的 oauth_gm.txt 文件內容：
  1. 登錄到您的@gm 帳戶
  2. 到 https://console.developers.google.com
  3. 接受許可並創建 Google API 項目(圖 13)
  4. 啟用 Google +Domain API(圖 14)
  5. 在 Google + Domain API 憑證下，按 CONFIGURE CONSENT SCREEN 按鈕
  6. 選擇內部或外部用戶類型並添加新的應用程序名稱(圖 15)
  7. 在“ API 和服務”下，進入“憑證”頁面
  8. 為 Web 應用程序(圖 17)創建“ OAuth 2.0 客戶端 ID”(圖 16)類型的憑據
  9. 授權的 JavaScript 來源：https://localhost:8443(圖 18)
  10. 授權的重定向 URI：https://localhost:8443/login/google/(圖 18)
  11. 將自己的的客戶 ID 和客戶密碼保存到 oauth_gm.txt 中再存到指定位置

圖 13 新建 Google API 專案

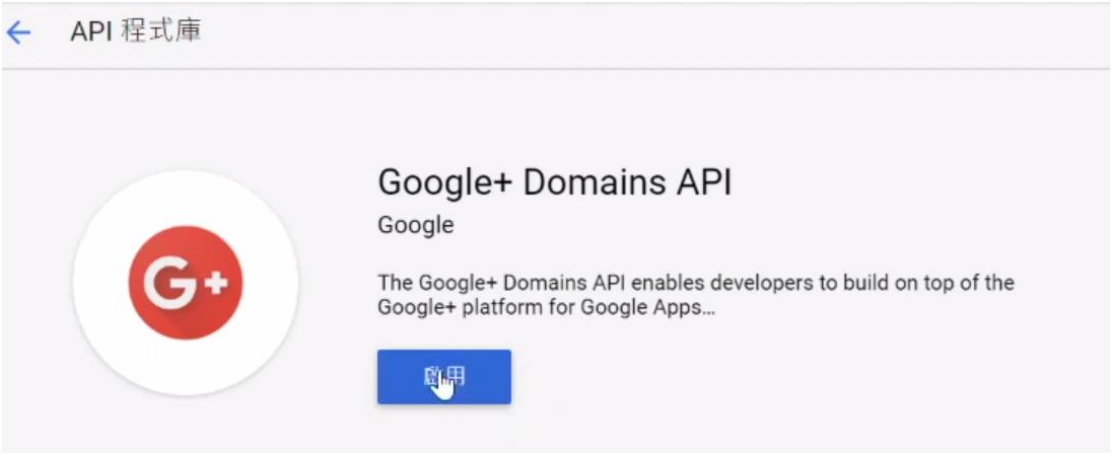


圖 14 啟用 Domains API

圖 15 Oauth2 用戶類型驗證

圖 16 建立 Oauth2 憑證

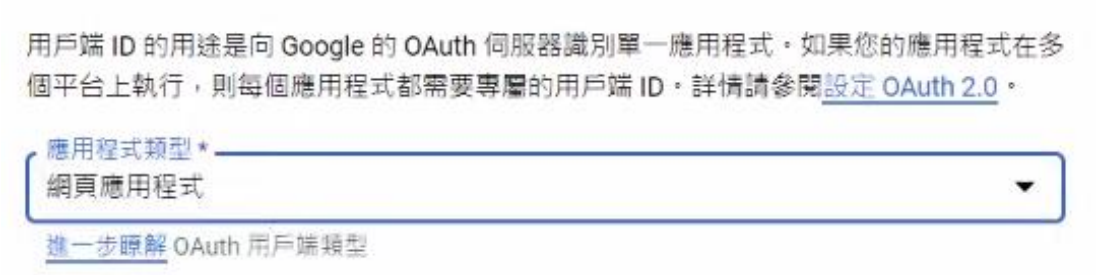


圖 17 應用程式類型

圖 18 URL