✧ 簡歷

一、基本資料

姓　　名：歐鎧豪
連絡電話：+886 912 778 955
電子信箱：okh8609@gmail.com

學　　歷：
國立臺灣大學 資訊工程所 （2020 - 現在）
國立臺灣科技大學 資訊工程系 （2016 - 2020）
高雄高工 資訊科 （2013 - 2016）

二、學習表現

- 臺灣科技大學 資訊工程系 **連續 7 個學期**獲得**書卷獎**的殊榮
  並且在**三年半**的時間內 以**第一名**的成績取得**學士學位**
- 高雄高工 資訊科 在學期間**每個學期**「學期總成績」都是**第 1 名**

三、相關作品

(一) iOS 限時聊天 APP

採用 Web API 架構，手機端可設定訊息保存期限，有隱密的聊天過程。

(二) 留言板網站

一款以 ASP.NET MVC 撰寫的網頁留言板，有看板功能、好友功能…等。

(三) 資料申報後台

以 Golang 搭配 Gin 框架，實作出一個可自訂申報欄位的後台系統。

(四) 3D 迷宮

僅透過 OpenGL 提供之 2D 繪圖 API，將迷宮以 3D 的方式呈現。

(五) 3D 雲霄飛車

使用 OpenGL 搭配 Shader，創造出自己的遊樂場。

(六) 第一人稱動作遊戲

使用 Unity 實作第一人稱動作遊戲，玩家可以發動近身攻擊與魔法攻擊。

(七) 第三人稱射擊遊戲

使用 Unity 實作第三人稱射擊遊戲，玩家可以射擊敵人。

(八) 圖像處理

使用 C++實作快速傅立葉轉換，並能夠呈現高通、低通濾波效果。
使用 C++實作一些圖像處理演算法，如：調整對比度、邊緣增強…等。

(九) 人體身形調變

先抓取人體的骨架點並切分人體部位，接著對模型各部位進行胖瘦調控。

(十) 室內 Wi-Fi 定位

蒐集空間中 Wi-Fi 訊號的 RSSI，使用 MLP 來判別目前所處的室內位置。

(十一) 智慧門鈴

樹莓派搭配 LINE Chatbot、網站前後端、人臉辨識來實作智慧門鈴。

(十二) 組態管理與自動化佈署

利用 Jenkins 與 Ansible 來自動化佈署應用程式到樹莓派上執行。

(十三) Automatic Guided Vehicle & Fleet Management System

實作自動導引車（AGV）與車隊管理系統（FMS）。

(十四) 小畫家

一款以微軟 GDI+ API 開發之桌面應用程式，提供簡單的繪圖功能。

(十五) 汽車防盜系統

偵測車輛是否遭到入侵，並即時取得相關訊息（如：GPS、車內影像）。

(十六) 智能電源供應器

利用回授電路，設法使電源的輸出電壓能夠盡量保持穩定。

四、 實務技能

(一) 熟悉且有相關實務開發經驗

C / C++ 與 STL

C# 與 ASP.NET （WebForm、MVC、Web API）

Python、tf.Keras

Golang、Gin

Android APP、Java

iOS APP

Linux 基本操作

(二) 曾經有相關專案的參與經驗

HTML、CSS、JavaScript、jQuery、Bootstrap

Django、網路爬蟲

MySQL、MS SQL

五、 其他事蹟

證券櫃檯買賣中心 資訊部門 實習生 （2018/07 - 2018/08）

擔任臺灣科技大學 電腦研習社 社課講師 共 4 小時

參加 教育部資訊安全人才培育計畫 資安實務攻防研習營 共 11 小時

高雄市 第 55 屆中小學科學展覽會 第二名

教育部 全國高職學生 104 年度專題暨創意製作競賽 佳作

2015 第十一屆全國電子設計創意競賽 資通組 佳作

✧ 相關作品

一、 iOS 限時聊天 APP

(一) 專案簡介

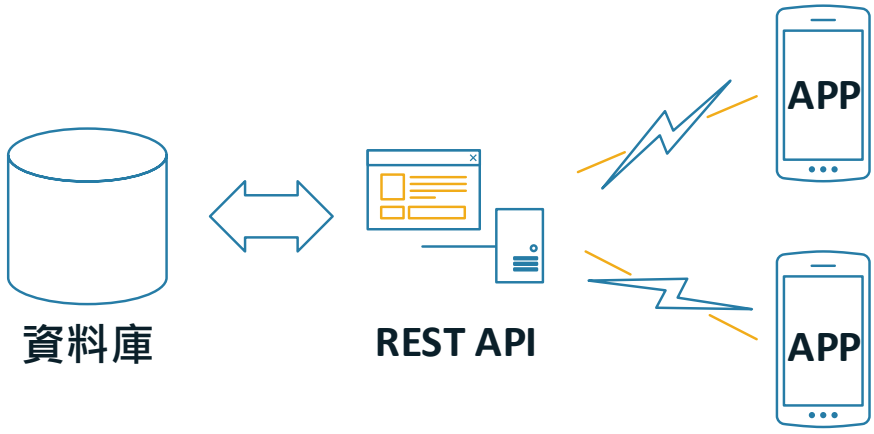
這款聊天軟體可以設定訊息的有效期限，只要時間到期後，這則訊息將永遠消失在這個世界上；達到高隱匿性、低身分識別的文字聊天軟體，讓使用者可以安心的使用，而不必擔心會留下任何紀錄。

(二) 功能說明

1. 帳戶功能：
   使用者帳戶之註冊、登入、登出、編輯帳戶資料。
2. 好友功能：
   可以搜尋好友、加入好友清單（有點類似通訊錄的概念）。
3. 限時聊天功能：
   聊天內容會在設定的時間過後刪除，達到高隱匿性的目的。
4. 即時邀請功能：
   提供邀請連結，讓被邀請方可以不需註冊登入即可使用該通訊功能乙次。

(三) 系統架構

我們的系統架構圖如下圖所示：

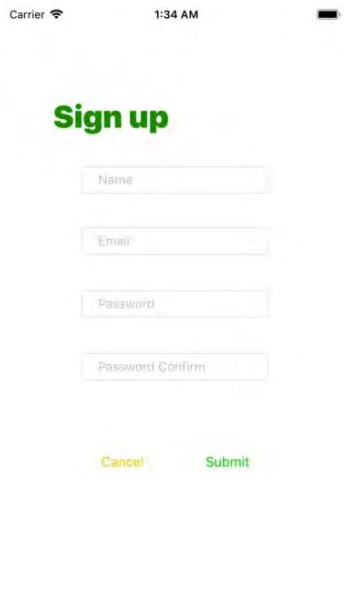


為提高應用**程式的彈性**，本專案採用**分層的方式實作**。我們以 ASP.NET 開發一套 **RESTful API**，其提供對資料庫進行操作的功能。

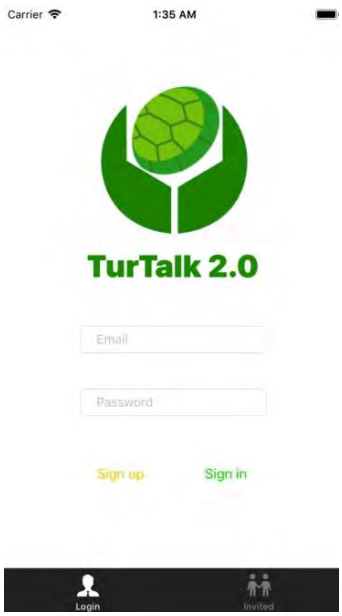
我們開發的手機 APP 端，將**透過 HTTP Method** 對後端之 RESTful API 提出資料的請求，所有關於資料庫的存取是藉由 RESTful API 來完成，而不是直接透過手機端對資料庫進行操作。

此設計方式的**優點**在於，如果我們未來想在**不同的平台上做出同一套的功能，我們不必再從頭撰寫程式**，只需了解如何調用該 RESTful API。

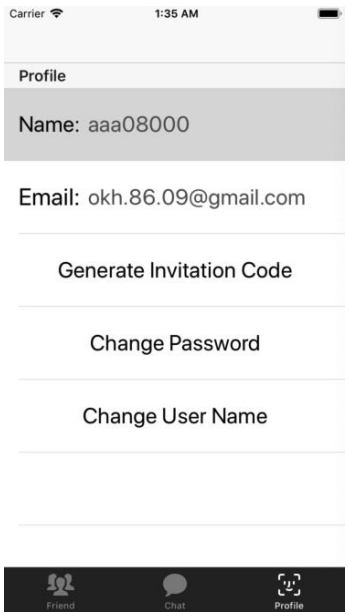
(四) 成果展示



▲ 帳戶註冊

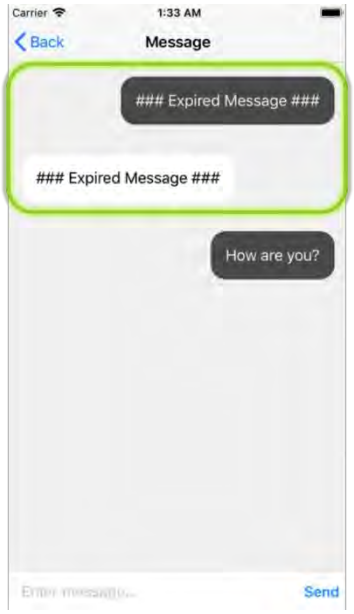


▲ 帳戶登入登出



▲ 編輯帳戶資料

▲ 好友功能



▲ 限時聊天功能

▲ 即時邀請功能

二、 留言板網站

(一) 專案簡介

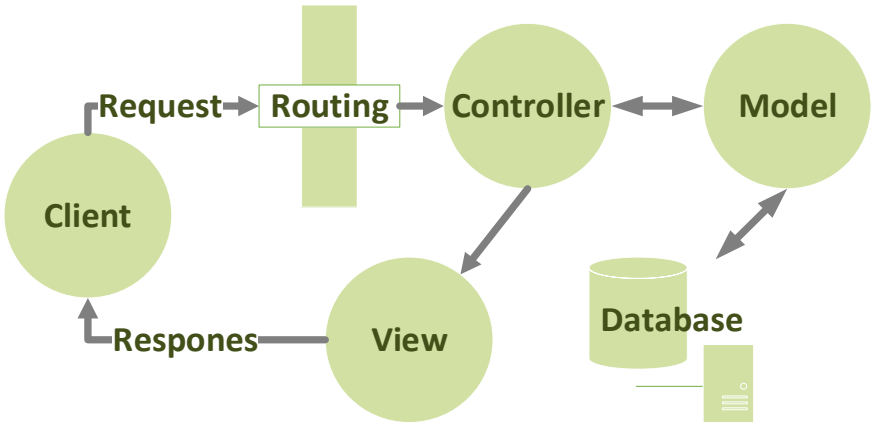
這是一款多功能型的留言版網站，除了基本的發布文章與留言功能之外，它還具有刊登廣告、家族、動態頁、看板、抽卡等功能。

(二) 功能說明

1. 帳號管理功能：
   帳號註冊、登入登出，以及編輯帳戶資料。
2. 社交功能：
   好友功能、動態頁功能，建立、參與家族功能，抽卡功能。
3. 廣告功能：
   以自己發布的文章之熱門度為籌碼，來張貼廣告。
4. 看板功能：
   可以新增看板，且可在看板中發布文章。

(三) 系統架構

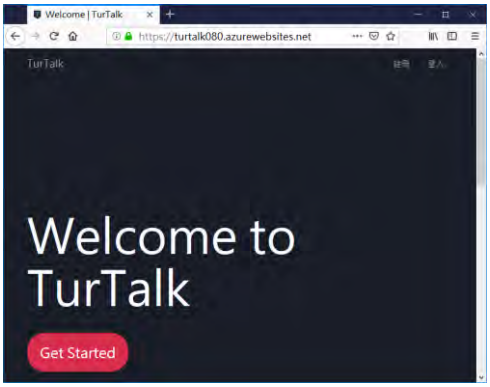
本專案使用 **ASP.NET 之 MVC 專案**實作，其中每個元件之間的交互關係如下圖所示：



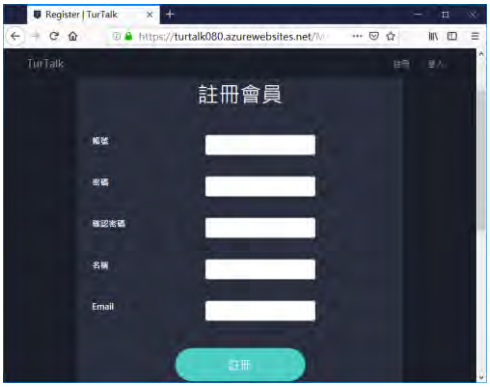
▲ 系統架構圖

當 Client 發送 Request 到 Server 後，會先根據發出請求的 URL 經過 Routing 選擇相應的 Controller；此 Controller 透過 Model 與資料庫取得資料後，將資料填入相應的 View 中，再回傳給 Browser，讓畫面得以呈現在 Clinet 的面前。

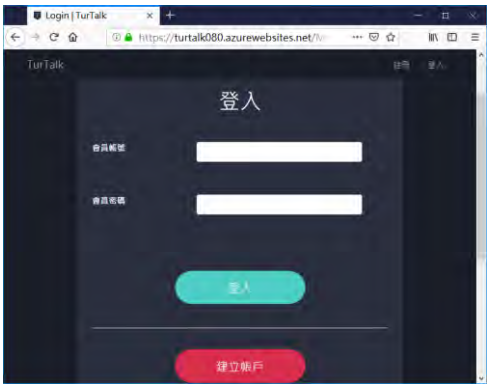
(四) 成果展示



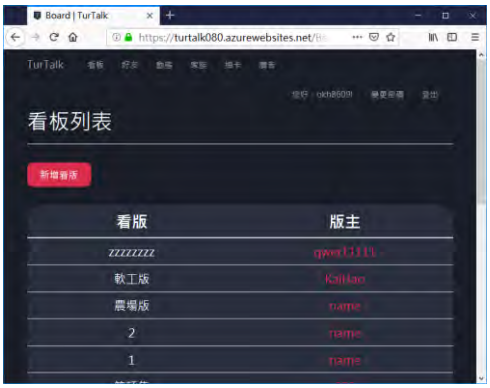
▲ 起始頁面



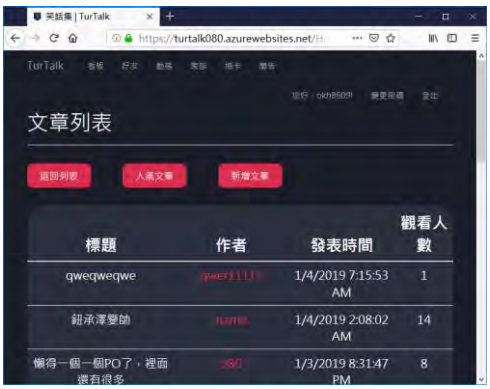
▲ 註冊頁面



▲ 登入頁面

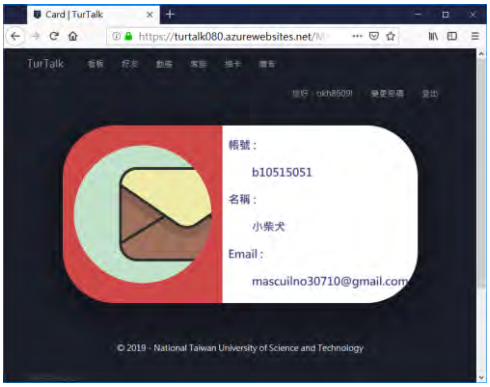


▲ 看板列表



▲ 文章列表

▲ 文章內容



▲ 抽卡功能

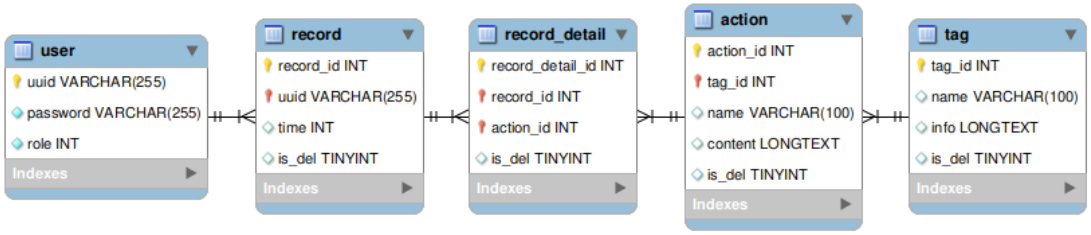
▲ 個人頁面

三、 資料申報後台

(一) 專案簡介

本專案以 Golang 搭配 Gin 框架，實作出一個可自訂申報欄位的後台系統。管理者在佈署完畢後，可以透過 RESTful API 給出未來這個系統將要申報的欄位項目，往後的所有申報紀錄都必須遵循該欄位的設定。

這個專案比較困難的地方在於資料庫的設計（如下圖所示），由於欲申報的項目是佈署後才決定的，所以我們勢必要以另外一個表格來記錄必須申報的欄位有哪些(如下圖中的 `tag` 資料表)，接著所有的紀錄條目都會去關連到這個表格。



此外，這套系統有實現相關帳戶註冊與登入的功能，並有做到權限管理的功能（某些特定的 API 需要管理員權限才能呼叫）。在驗證方面，我們使用 JWT 來進行驗證；在密碼保存方面，我們使用 PBKDF2 來保障其安全性。

(二) API 介紹

本專案所有功能的呼叫，均以 RESTful API 實作，以下列出幾個主要的 API 來作範例說明與展示：

- `POST /auth/user/verify`
  - 功能說明：驗證使用者帳號，並取得 JSON Web Tokens
  - 參數說明：'uuid=帳號'、'password=密碼'
  - 回傳資料：{"token":"JSON Web Tokens"}
- `POST /api/action`
  - 功能說明：新增可申報的項目（必須在 HTTP Header 中帶上 token）
  - 參數說明：'tag_id=該項目所屬的欄位 ID'、'name=項目名稱'
  - 回傳資料：{"action_id":該項目的 ID }
- `POST /api/record`
  - 功能說明：新增申報紀錄（必須在 HTTP Header 中帶上 token）
  - 參數說明：{"actions":[該筆申報紀錄的所有項目 ID]}
  - 回傳資料：{"record_id": 該筆申報紀錄的 ID }

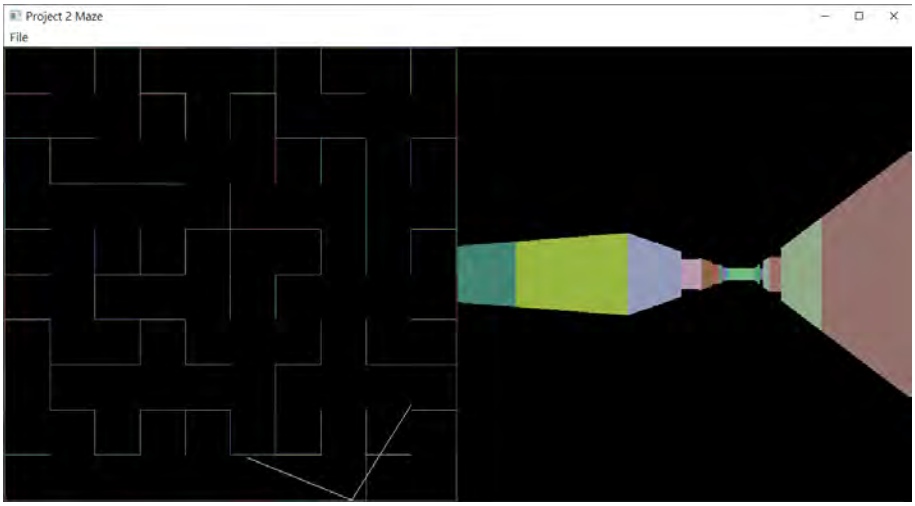
四、 3D 迷宮

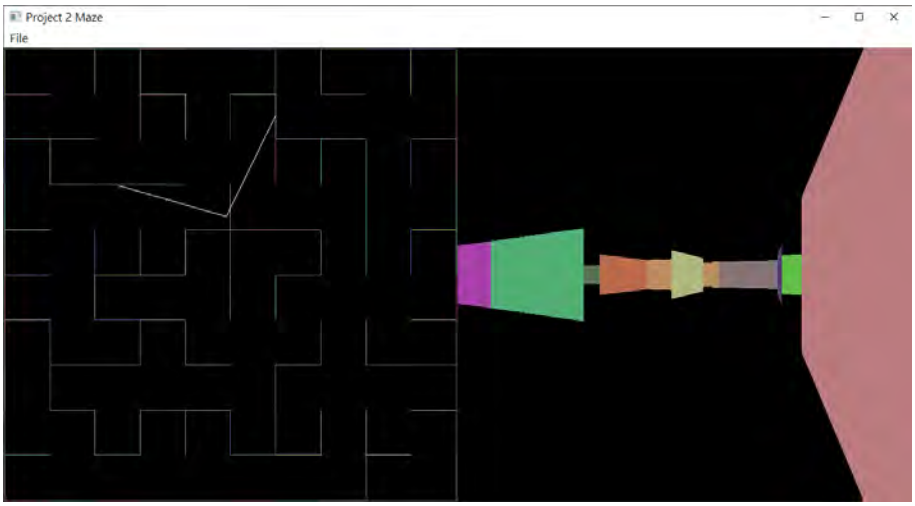
(一) 專案簡介

本專案僅透過 OpenGL 提供之 2D 繪圖 API，來搭建具有 3D 立體感的畫面。其困難點在於必須自己實作電腦圖學相關之座標轉換。

首先，我們要將畫面中物體之世界座標（World Space）轉換為以相機為原點之相機座標（Camera Space）。接著，基於效能的考量，我們會利用 Cell and Portal Visibility Algorithms 來將看不見的物體剔除。最後，我們將看得見的物體，投影到畫面上做呈現。

(二) 成果展示





(三) 原始碼

https://github.com/okh8609/CG_Project2_Maze

五、3D 雲霄飛車

(一) 特色功能

- 軌道車

  利用數學曲線計算軌道，並且可以讓火車沿著軌道行走。
- OBJ 載入

  自己實作 OBJ 檔案讀取，並可將頂點載入 shader 中。
- 煙火粒子特效

  設定每個粒子的大小與衰退速度，達到粒子爆開、再慢慢消逝的效果。
- 水波 shader

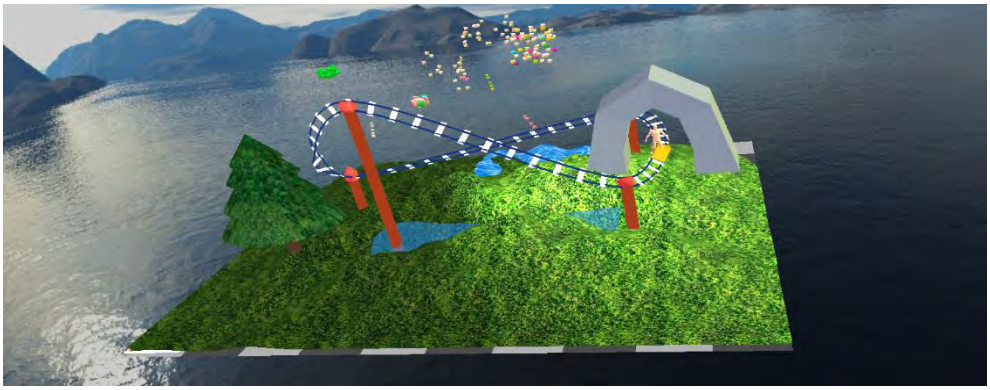
  透過 sin 波與海浪貼圖來模擬出水波的效果。
- 日出日落 shader

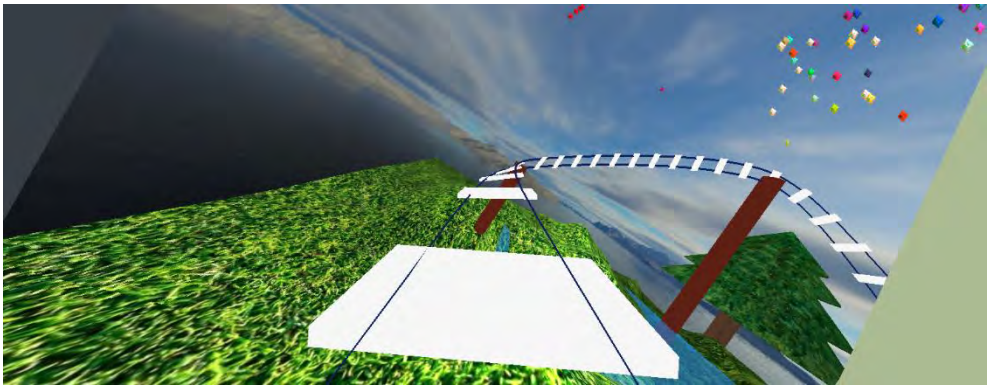
  利用 Phong lighting model 技術與時間參數，讓大地的光影隨時間變化。
- Skybox

  將所有物體包裹於一個六面貼圖的正方體中，使得場景不會過於單調。
- 樹木貼圖

  在同一個模型中，貼上兩張貼圖（樹皮與樹葉）。

(二) 畫面呈現





(三) 相關連結

介紹影片：https://youtu.be/hA1q2IDi1Ik

原 始 碼：https://github.com/okh8609/CG_Project3_RollerCoaster

六、 第一人稱動作遊戲

(一) 專案介紹

使用 Unity 搭配動畫器、粒子特效、碰撞器與相機鏡頭的切換，來實作一個第一人稱的動作遊戲；其中，主角可以對敵人發動近身攻擊（用劍砍），與遠距離發動魔法攻擊。

(二) 畫面展示

近身攻擊如下圖所示：

遠距離魔法攻擊如下圖所示：

近身被攻擊如下圖所示：

七、 第三人稱射擊遊戲

(一) 專案介紹

使用 Unity 搭配粒子特效、碰撞器、燈光效果與相機鏡頭的切換，來實作一個第三人稱的動作遊戲。在這個專案中特別加了小地圖，可以讓玩家看到自己與敵人的位置，增加玩家對遊戲的掌握度。另外，我還加入了一個升降機關，可以增添遊戲的樂趣。

(二) 畫面展示

玩家射擊敵人的畫面如下圖所示：

亦有實作鏡位切換，可以使用第一人稱來進行遊戲：

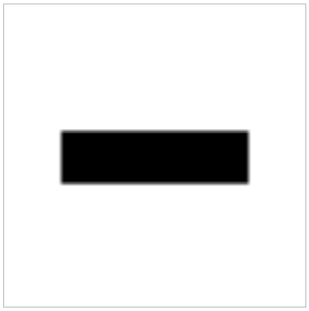
八、 圖像處理

(一) 傅立葉轉換實作

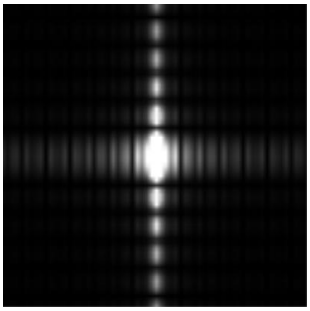
這是工程數學的課程專案，各位審查委員也許會看到其他組別有類似的作品，但是我是直接以指標的方式對內部記憶體做操作，時間（計算速度）與空間（耗費記憶體）上的表現都較其他組別優異許多。

1. Discrete Fourier Transform, DFT 實作

透過離散傅立葉轉換，將圖像從空間域轉換到頻域上，可看出其頻率的分布情況。（圖片大小：64 pixels × 64 pixels）



▲ 原圖



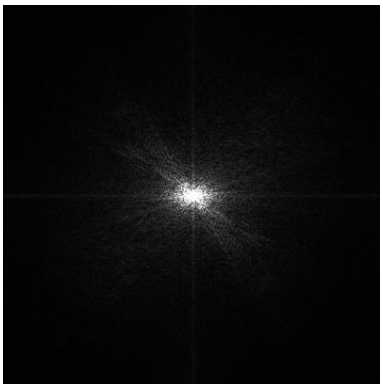
▲ 頻譜圖

上述方法，若遇到尺寸大一點的圖片，會耗費分鐘等級以上的時間，因此實作以下的經過加速後的演算法。

2. Fast Fourier Transform, FFT 實作

基於 DFT 演算法，並利用 Divide and conquer 的策略，減少重複的計算，可以更快速地取得頻率的資訊。（圖片大小：512 pixels × 512 pixels）

▲ 原圖

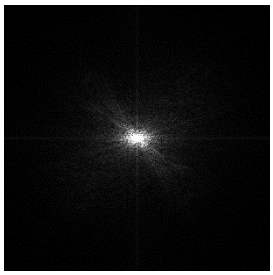


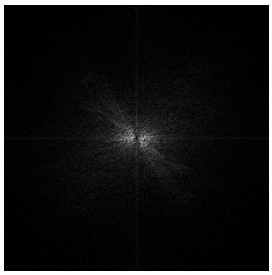
▲ 頻譜圖

3. High-pass filter, HPF 實作

有了頻譜圖之後，將低頻的訊號去除，留下高頻的訊號，即留下影像中物體的邊緣。


▲ 原圖


▲ 頻譜圖

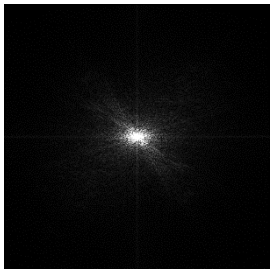

▲ 濾波後

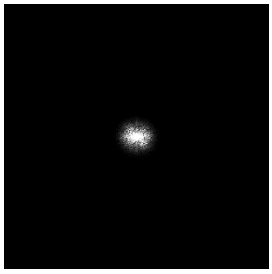

▲ 逆運算還原

4. low pass filter, LPF 實作

有了頻譜圖之後，將高頻的資訊去除，留下低頻的資訊，整張影像看起來會是模糊的感覺。


▲ 原圖


▲ 頻譜圖


▲ 濾波後


▲ 逆運算還原

(二) Histogram equalization 實作

將圖像中灰階值最小變為 0，灰階值最大者變為 255，其他則根據出現機率平均分配，可以調整圖像對比度的方法。


▲ 原圖


▲ 調整對比後

(三) Edge Enhancement Filter 實作

原本的圖片減去高斯模糊後的圖片會留下高頻資訊（影像的邊緣）。

▲ 原始圖片　　▲ 高斯模糊　　▲ 邊緣

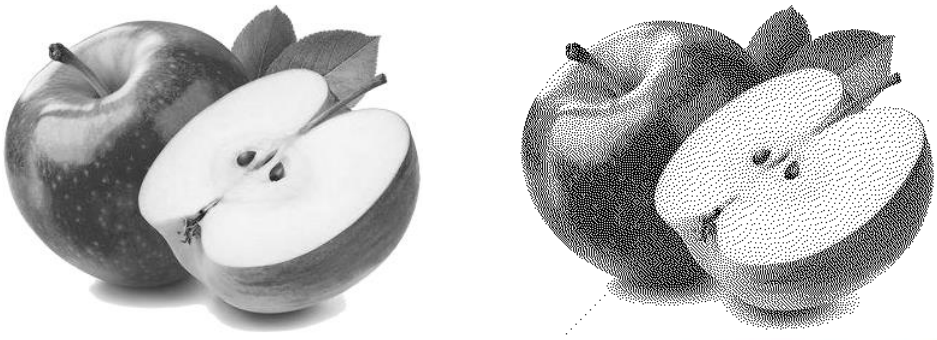
原始影像再加上影像的邊緣資訊，即可達到邊緣增強的效果。

▲ 原始圖片　　▲ 邊緣　　▲ 邊緣增強

(四) Floyd-Steinberg Dithering 實作

雖然我們可以在電腦螢幕上呈現灰階的影像，但是在列印成品時，印表機還是只能透過細小的黑點，來達到視覺上的灰階的目標。

此演算法的中心思想在於，做影像二值化的時候，會利用旁邊的點，來補足自己產生的誤差，產生視覺上的灰階的效果（實際上圖片僅由黑白兩色組成）。



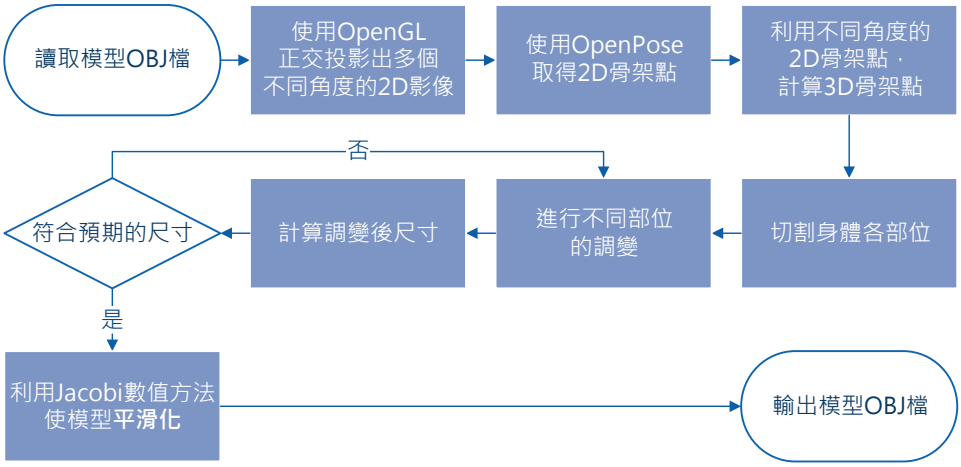
▲ 原始影像　　▲ 運算結果

九、 人體身形調變

(一) 專案說明

本專案可以對 3D 的人體模型進行各部位的調整，我們可以讓某個人體模型變胖或是變瘦，讓使用者得以看見自己不同體態的樣貌。我們會先設法取得人體的 3D 骨架點，接著利用人體骨架點的座標為依據，切分出人體各個部位，進而對人體進行調變的操作。

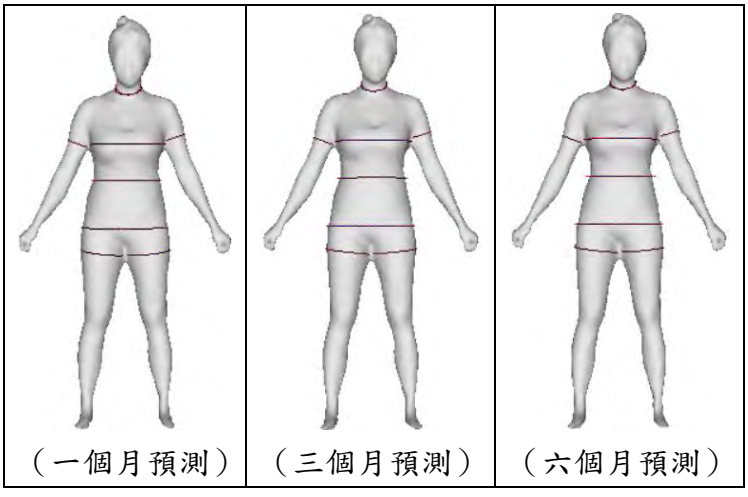
其工作流程如下圖所示：



(二) 專案結果

以下為實驗用的人體模型示範結果輸出，我們可以成功地將人體模型自然地調整為希望的尺寸。

▲ 人體模型



▲ 人體模型 身型調變結果

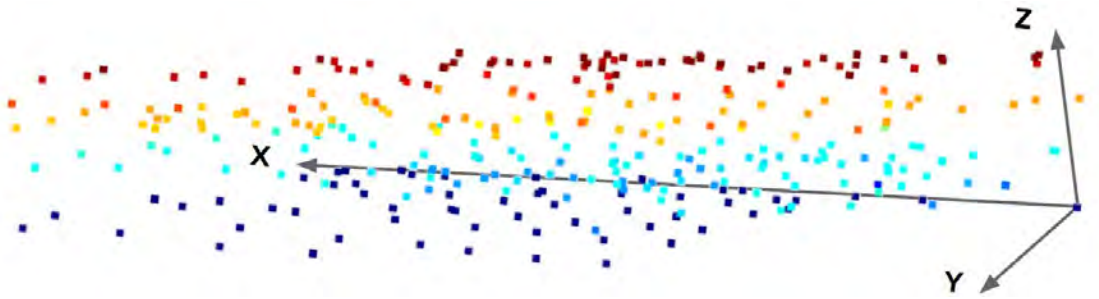
十、 室內 Wi-Fi 定位

(一) 專案簡介

本專案利用了空間中分布的 Wi-Fi 訊號強度，來判別目前所處在的室內位置。

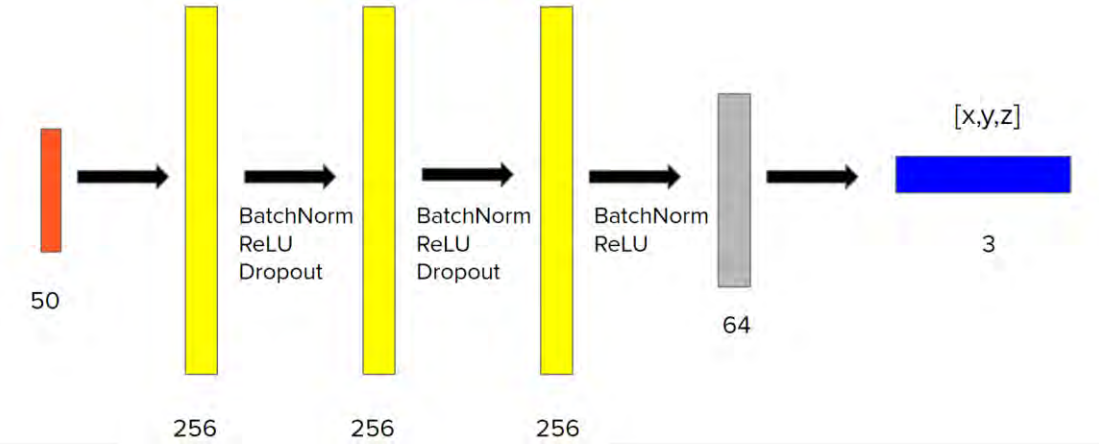
(二) 資料蒐集

本專案在教室中蒐集了 300 個點，其各點在空間中的散佈如下圖所示，此外我們同時使用 5 台樹莓派（Raspberry Pi）同步蒐集來提升效率。



(三) 模型架構

我們僅使用了一個簡單的 MLP (Multilayer Perceptron)來作為預測模型，其架構如下圖所示：



(四) 實驗結果

我們以預測值與原資料之間的歐氏距離來作為目標函數，實驗結果如下列所示，平均驗證誤差僅有 3.5 公尺。

Training Loss：0.87 m

Validation Loss：3.5 m

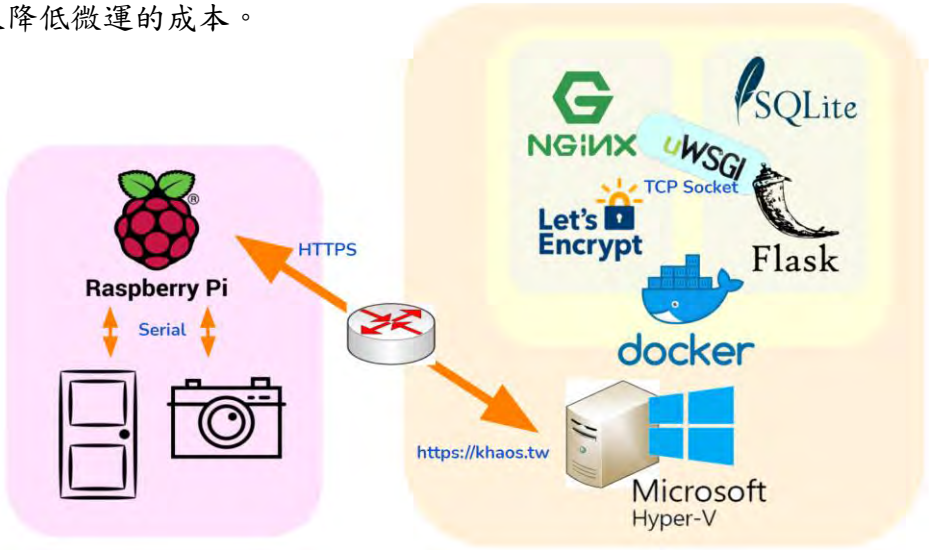
十一、 智慧門鈴

(一) 專案簡介

當有人按下電鈴按鈕時，會利用樹莓派（Raspberry Pi）搭配鏡頭模組拍下訪客的身影，並透過 LINE Chatbot 即時傳送影像到管理者的手機與後端管理網頁，讓管理者可以遠端開鎖（以電磁閥作為示範）。此外，後端亦結合人臉辨識系統，可以令已知的使用者自由通行。

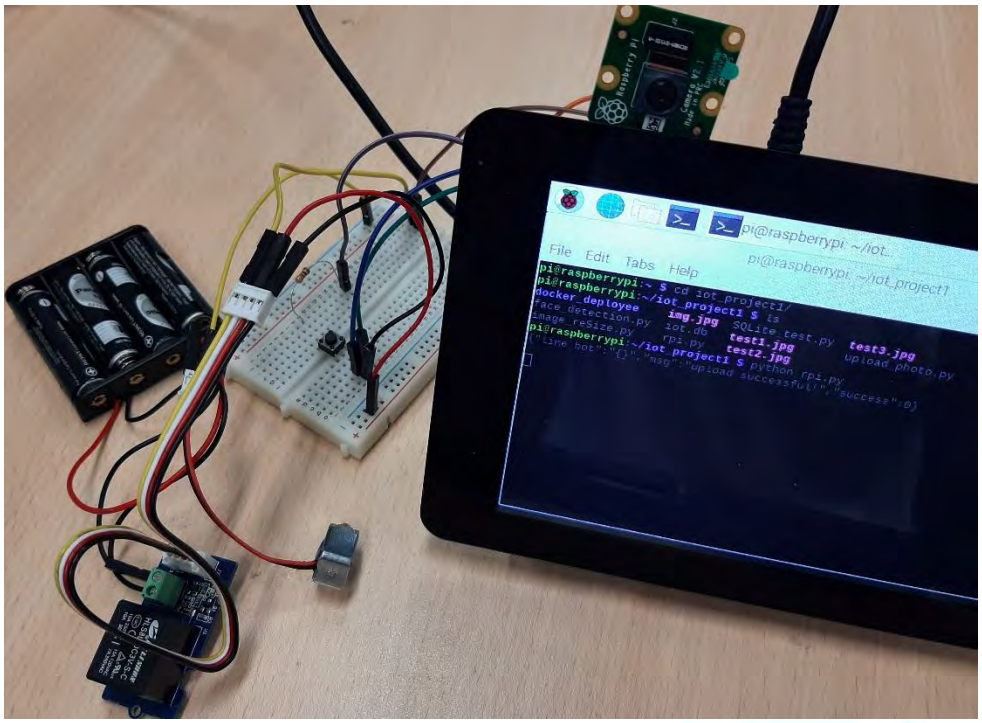
(二) 系統架構

本專案之系統下構圖如下圖所示，後端搭配虛擬機器與 Docker 來佈署，可以大大降低微運的成本。



(三) 成果展示

硬體部分的配置如下圖所示：



LINE Chatbot 的操作成果如下圖所示：

後端管理網頁呈現如下圖所示：

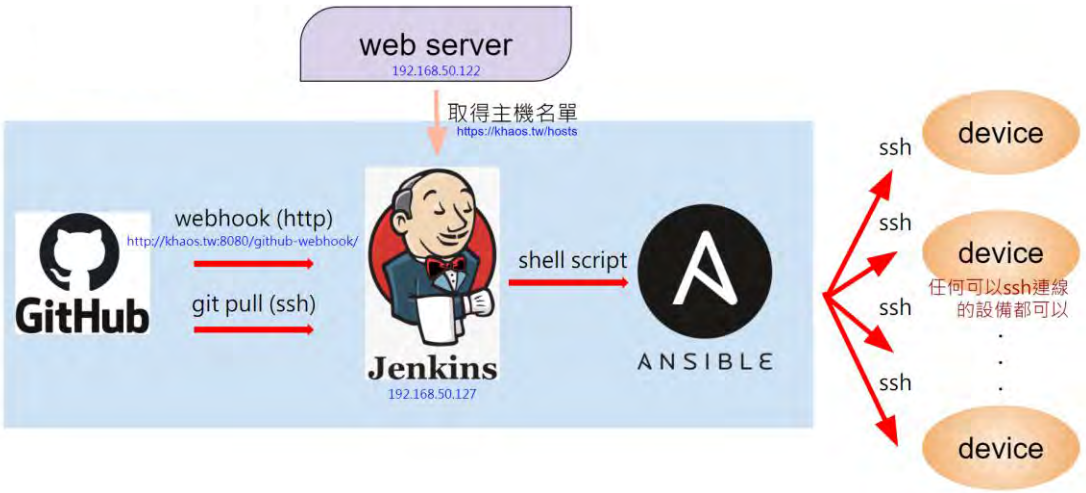
十二、 組態管理與自動化佈署

(一) 專案簡介

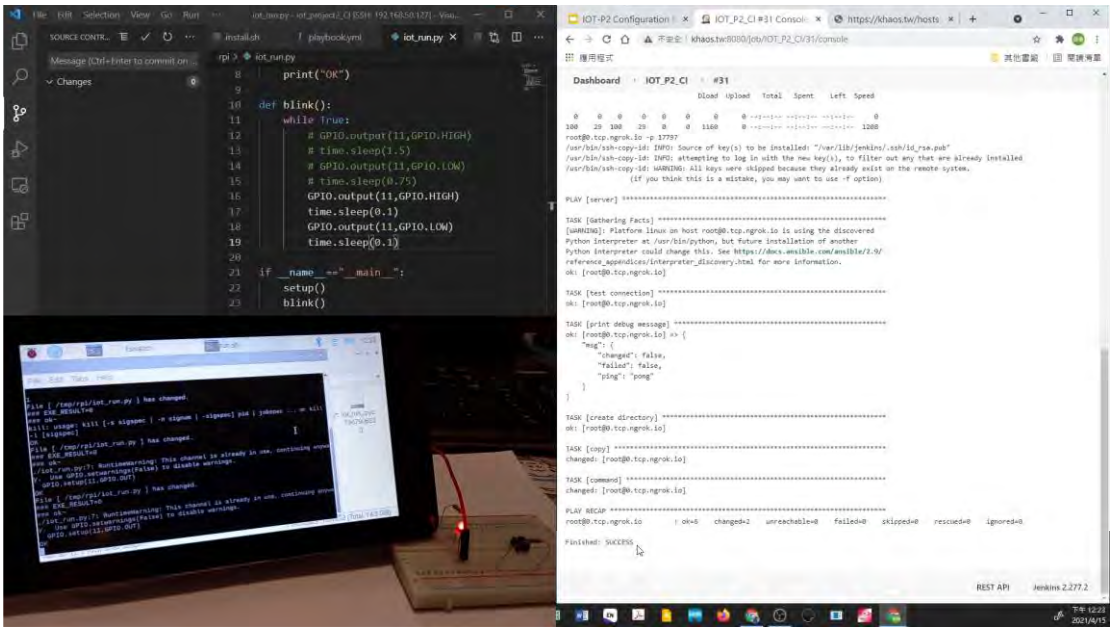
本專案利用 GitHub 的 webhook 來驅動 Jenkins，並使用 Ansible 來佈署相關應用程式到樹莓派（Raspberry Pi）上執行。而樹莓派（Raspberry Pi）上亦有對應的監控程式（利用 fswatch 搭配 shell script 達成），可以自動編譯與執行平台相依的新的程式碼；若新的程式碼無法正常運作的話，會自動回覆上一個版本的應用程式。

(二) 系統架構圖

該專案的系統架構圖如下圖所示：



實驗成果畫面截圖：



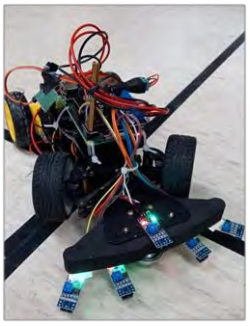
十三、 Automatic Guided Vehicle & Fleet Management System

(一) 專案簡介

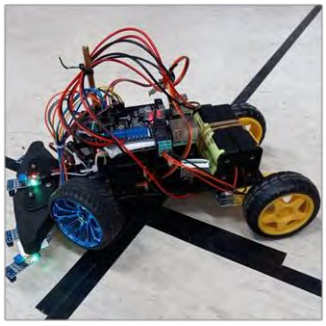
本專題要實作自動導引車（AGV，Automated Guided Vehicle）與其車隊管理系統（FMS，Fleet Management System），需要在指定場域中開發出 FMS 來對 AGV 下達指定的路由命令，讓 AGV 得以遵循指定的路徑行走，並即時將 AGV 之真實位置顯示於 FMS 上。

(二) Automatic Guided Vehicle 成果展示

本專題硬體配置完成圖如下圖組所示，其分別為不同視角拍攝之外觀呈現。值得注意的是圖中的藍色的輪子是沒有用到的，僅為當初實驗性質之配置。



▲ 府視圖

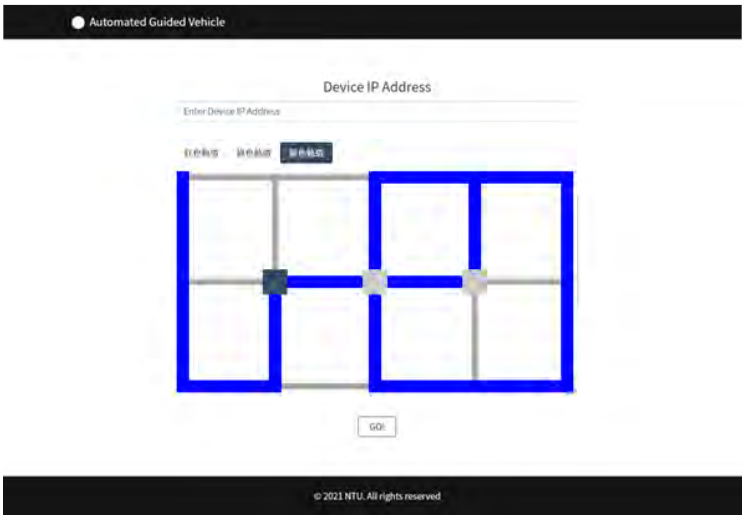


▲ 側視圖

▲ 仰視圖

(三) Fleet Management System 成果展示

本專題之車隊控制系統頁面如下圖所示，其中地圖的部分是以 HTML 的 canvas 元素搭配點擊事件完成，可以直接點選地圖中的深藍色方塊來選取中繼站點以完成路由選擇。



(四) 相關連結

本專案之成果展示影片如下列網址所示：

https://youtu.be/i42_GLMrgAw

本專案之完整程式碼公開於以下程式碼儲存庫：

https://github.com/okh8609/IOT_Project5_AGV-and-FMS

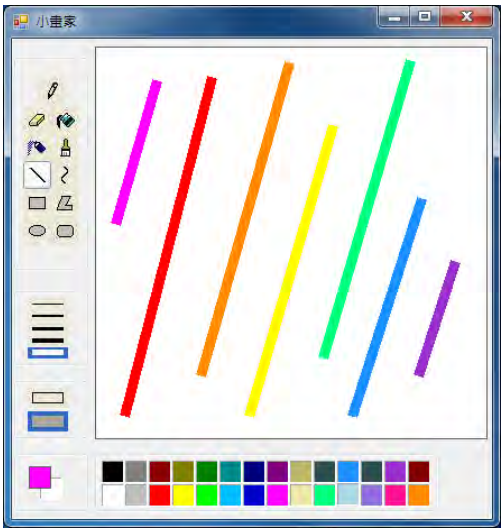
十四、 小畫家

(一) 專案簡介

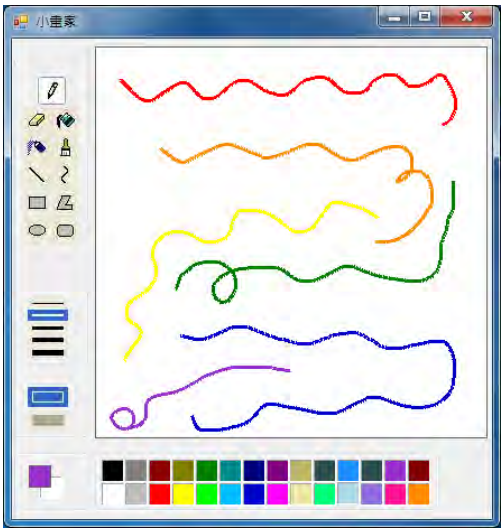
利用微軟提供的 **GDI+** API 開發一套簡易版的小畫家，支援以下功能：

1. 繪製**直線**，**鉛筆**功能，繪製**多邊形**，繪製**貝茲曲線**
2. **筆刷**與**橡皮擦**功能，**噴槍**功能
3. 繪製**矩形**、**橢圓形**、**圓角矩形**
4. **填色**功能

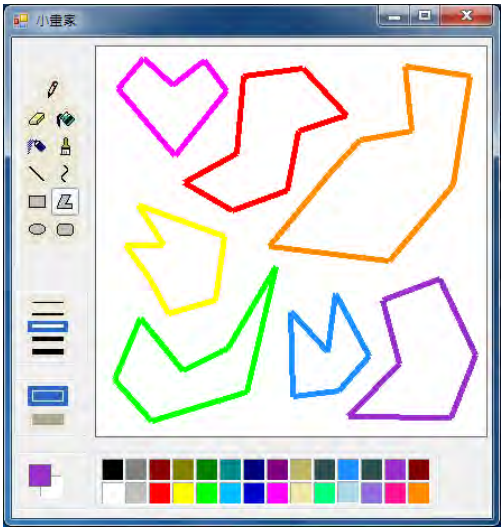
(二) 成果展示



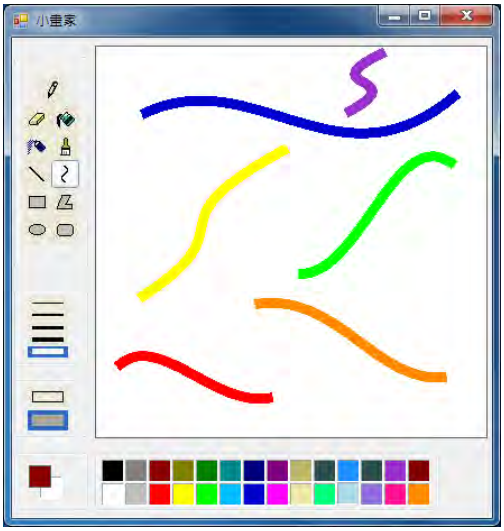
▲ 直線



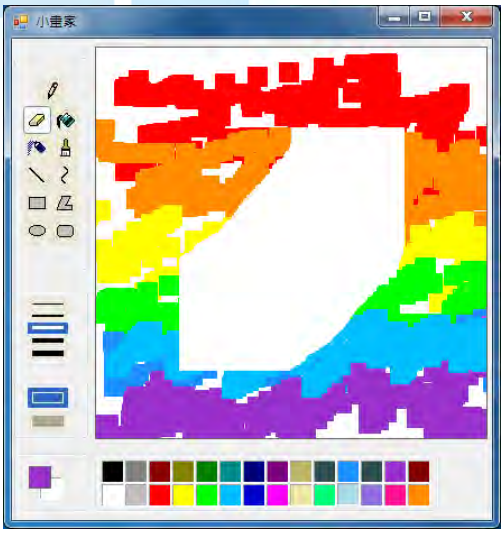
▲ 鉛筆



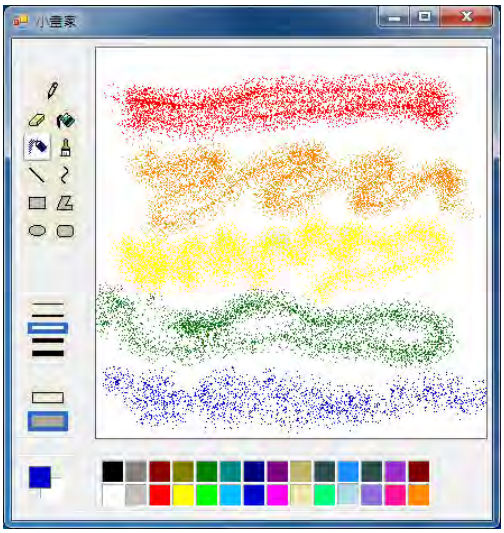
▲ 多邊形



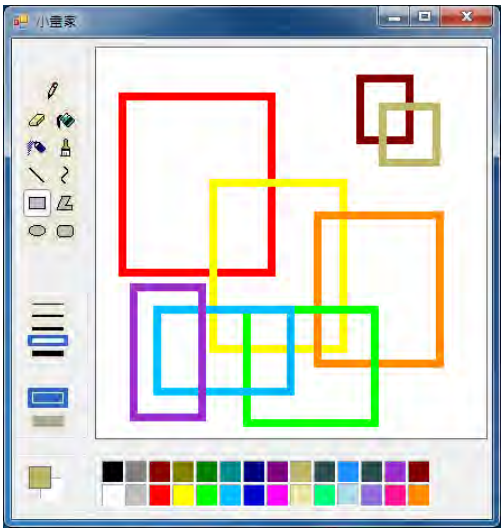
▲ 貝茲曲線



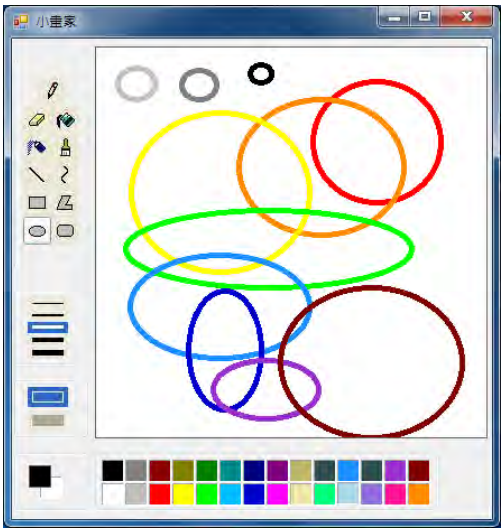
▲ 筆刷與橡皮擦



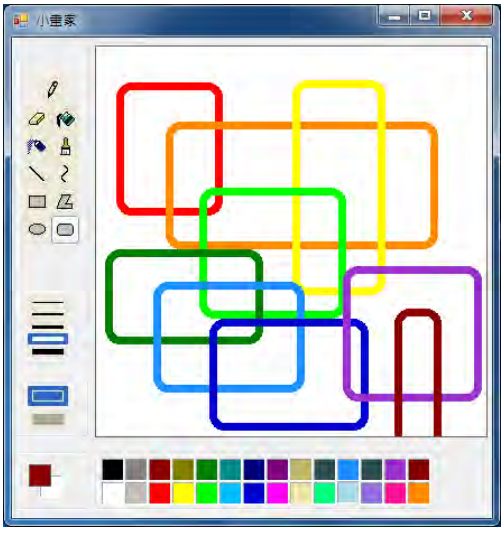
▲ 噴槍



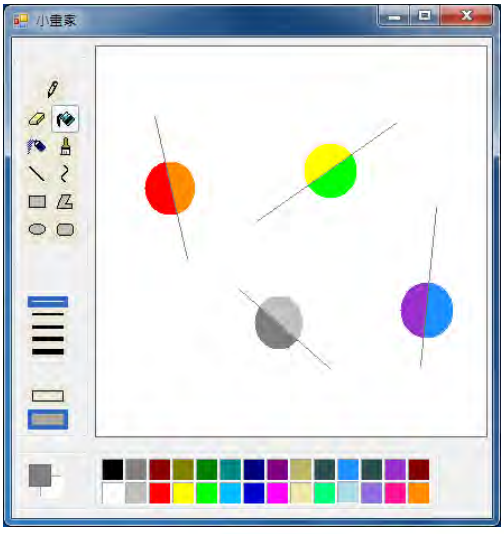
▲ 矩形



▲ 橢圓



▲ 圓角矩形



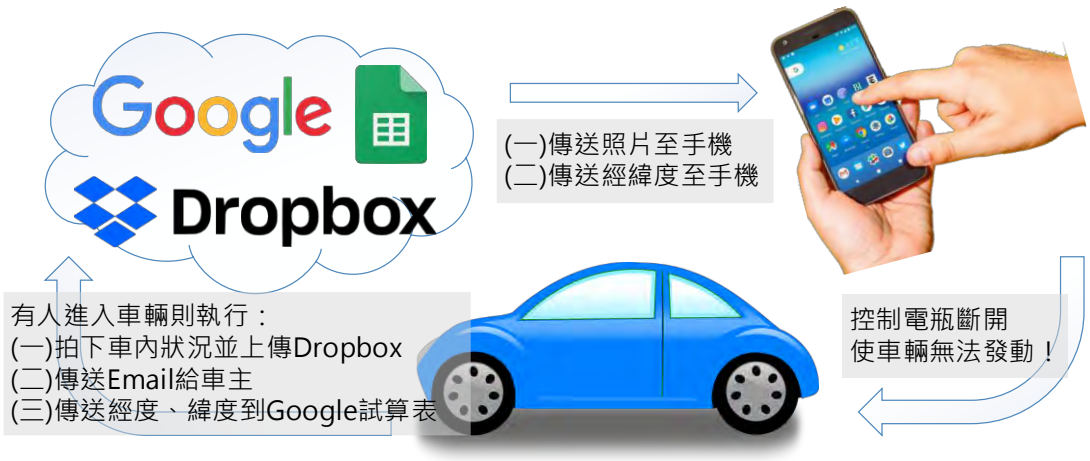
▲ 填色（將半圓填入不同顏色）

十五、 汽車防盜系統

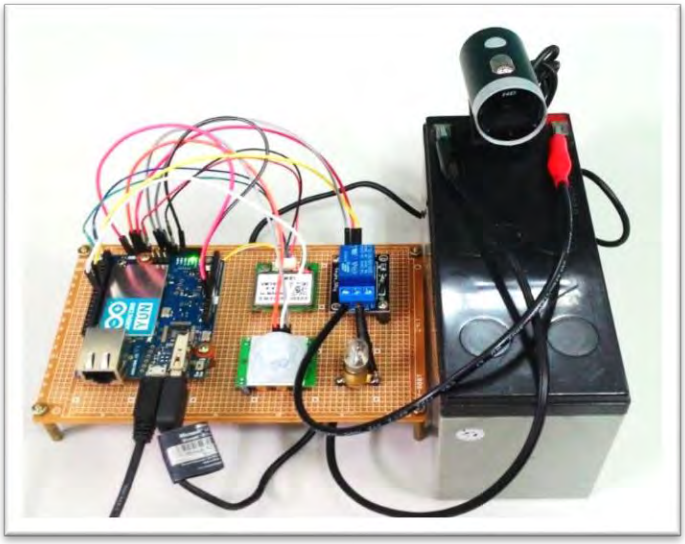
(一) 專案簡介

若紅外線偵測到有人入侵車內時，會開啟網路攝影機拍下車內狀況上傳雲端，並且傳送 E-MAIL 通知車主偵測到異狀；車主可以遠端斷開電瓶使車輛無法發動，並且可獲得車輛之 GPS 位置，以便尋獲車輛。

(二) 系統概念圖



(三) 硬體電路設置

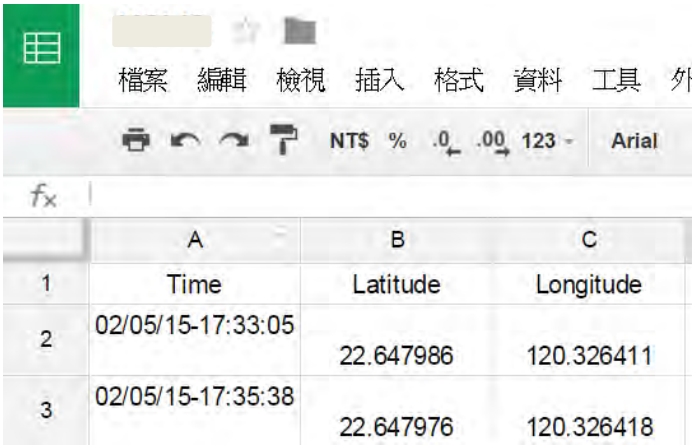


(四) 成果展示

1. 紅外線偵測到動作後，將照片上傳 Dropbox

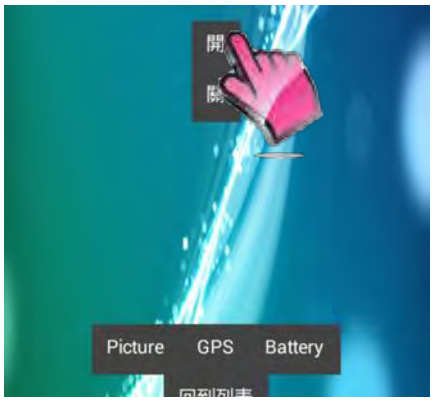
2. 紅外線偵測到動作後，以 Gmail 傳送警告訊息

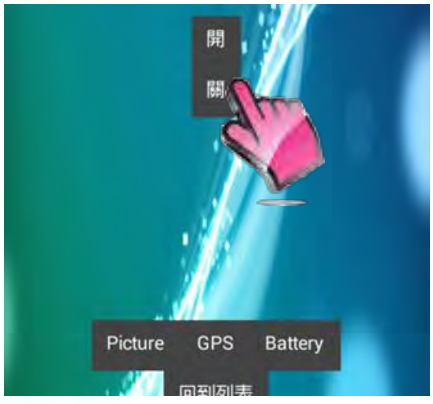
3. 紅外線偵測到動作後，將正確的經緯度上傳 Google Docs



	A	B	C
1	Time	Latitude	Longitude
2	02/05/15-17:33:05	22.647986	120.326411
3	02/05/15-17:35:38	22.647976	120.326418

4. APP 控制車輛與電瓶的連結，按下「開」與「關」，進行電壓測試。





十六、 智能電源供應器

(一) 實驗動機

在日常生活中，將電力系統（電源）接上電器（負載）後，觀察輸出電壓的變化，**發現其電壓略為下降。**例如：啟動吹風機或微波爐的瞬間，家中燈泡會閃爍。

基於以上現象，我們便著手探討如何利用**回授電路，使電源的輸出能盡量保持穩定，不受負載效應的影響。**換言之，我們希望使得電源供應器之電壓調整率 (Voltage Regulation，VR) 趨近於 0%。

(二) 功能說明

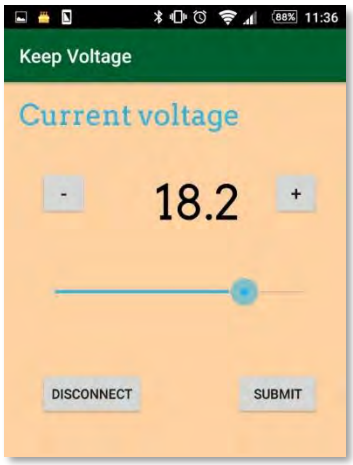
1. 設計**手機 APP** 並以**藍芽**為媒介賦予 Arduino 指定輸**出電壓值**
2. 利用**可調式電源模組**與 Arduino，設計穩定輸出的可調式電源供應器
3. 利用 Arduino **類比**接腳**測量電壓**，並顯示於 **LCD 上**
4. 藉由回授電壓資訊為依據，驅動**步進馬達**，校正輸出電壓

(三) 系統概念圖

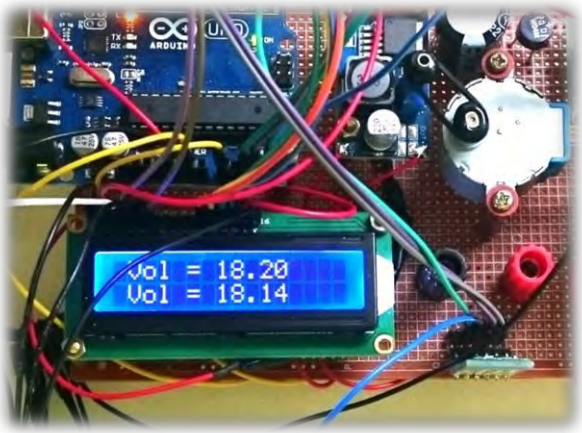
▲ 系統概念圖

(四) 成果展示

可以透過手機 APP 指定電壓，輸出電壓與使用者所期待的電壓差距極小。（右下圖中 上排為指定的電壓，下排為目前實際的輸出電壓）。



▲ 以 APP 指定電壓



▲ 電壓輸出顯示