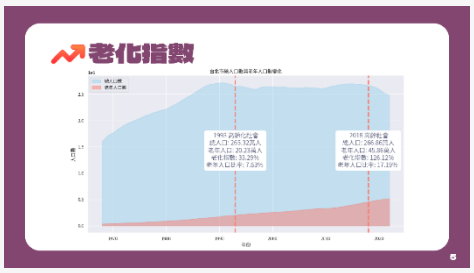
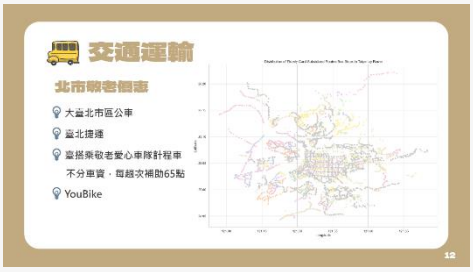
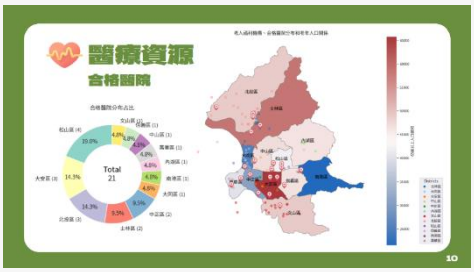
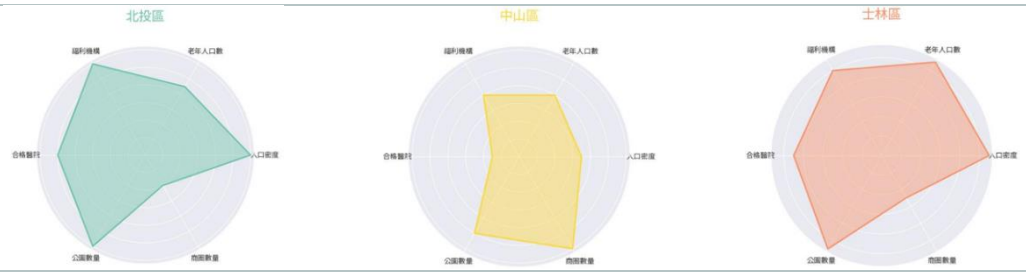
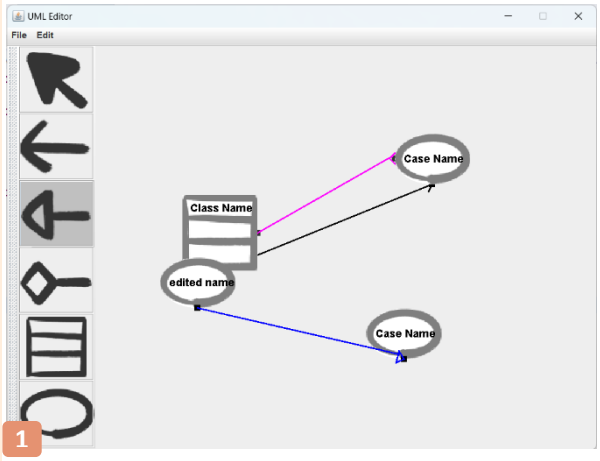
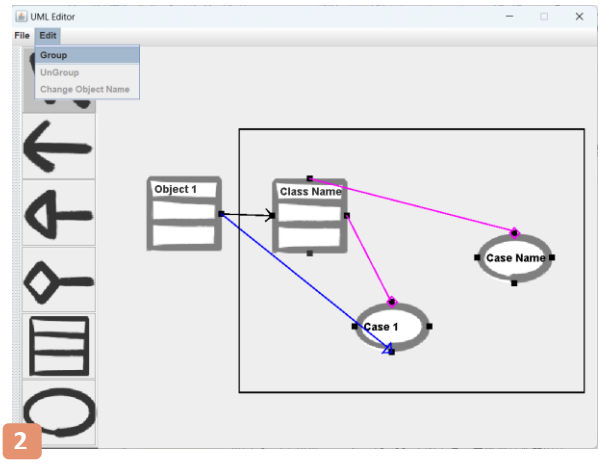
資料視覺化

開發語言	Python（pandas 套件）
簡　　介	使用政府公開資料，從台灣老化趨勢分析台北市適合養老的行政區，從醫療資源、交通運輸再到安居娛樂將結果視覺化的視角挑出平均總分最高、綜合評估最適合養老的去處。 **（使用堆積圖、時間序列分析圖、熱力圖、氣泡圖、甜甜圈圖、雷達圖）**
成　　果	   

實驗方法與結論

方　　法	依據主題資料特性分別挑選適合圖表進行分析
	老化趨勢：透過**堆積圖**和**時間序列分析**，台北市自 1992 年進入高齡化社會，並在 2022 年達到超高齡社會，老化指數持續攀升至 166.1，為**六都最高**。**熱力圖**呈現人口密度，探討台北市各區域老年人口分布，發現人口總數與老年人口總數非正比關係，如北投區老年人口相對較高。
	交通運輸：整合臺北市敬老愛心卡補助搭乘公車捷運路線，以網格方式點出所有公車站，以探討公共交通的便利性，使用**散點圖**來呈現分佈情況，分析指出整個台北市的交通便利性，無論居住地區，老年人生活都可以輕鬆地使用大眾運輸工具。
	醫療資源：使用**地圖**和**甜甜圈圖**來呈現合格醫院和福利機構的分佈情況，並分析各行政區的醫療資源情況，觀察各地區老年人口密度和醫療資源的關聯性。
	安居娛樂：整合北市商圈綠地及台北市行政區老年人口資料，探討老年人口密集地區的綠地分佈情況，使用**熱力圖**和**氣泡圖**來呈現公園的分佈情況，並統計各行政區的公園數量，例如北投區、士林區和中山區的公園分佈情況。
實驗結果	**綜合分析**：使用**雷達圖**呈現各方面表現，依需求選擇適合居住的地方（例：北投、中山、士林區） 

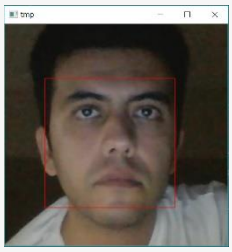
物件導向程式設計 — UML Editor	
開發語言	Java
成　　果	依照規格書開發符合需求的 UML 編輯器，識別類別、建立關聯、設計 UML 圖表、Design Pattern 和程式實現物件導向設計方法。
實驗方法與結論	
方　　法	1： **需求分析**：分析系統需求，確定功能和特性。 2： **類別識別**：辨認出必要的類別，包括基本物件、連結物件和 composite 物件。 3： **關係建立**：確定類別之間的關聯關係，包括聚合、組合、繼承等關係。 4： **UML 圖表設計**：畫出 UML 圖表，包括類別圖、用例圖和序列圖，視覺化系統結構和行為，**分析父類別（super class、interface）所需必要屬性與功能，再將特異部分於各子類別中實現**。 5： **測試與驗證**：完成 code 進行單元、整合測試，確保系統符合需求並具有正確的功能。 **依 Scenarios 找出物件之間繼承、多型、組合關係，分類並從各 Super Class 開始** 1： **Object** 基本物件為 class（方形）、 useCase（橢圓）、三種 line 的 super class，包含共同屬性與功能，如：物件深度、物件名稱、四邊連接 port 與 paint()等，其餘特異部分由 sub class 完成，利用 **多型（Polymorphism）**讓基本物件可以個別定義各自功能內容根據實際的需求去執行。 2： **Mode** 其中 baseMode 為 super class 集結所有左方六個功能（選取、連結線、建立基本物件），同理使用多型與繼承概念完成各自功能。 3： **UI** 包含 Canvas（Paint 所有物件）、ToolBar（六個功能）、MenuBar（群組物件、重新命名物件、取消群組）。
實驗結果	 ▲ 圖 1：建立物件、三種功能連結線  ▲ 圖 2：群組物件、更換物件名稱
結　　論	透過專題了解物件導向核心概念是＂去耦合＂，避免牽一髮動全身的義大利麵式 coding。設計 UML Diagram 釐清 class 彼此關係，充分體會到除了繼承讓程式碼可以重用外，多型的概念讓程式碼更彈性。 https://github.com/NaomiLinBlog/UML-Editor

影像辨識 — 真偽人臉辨識

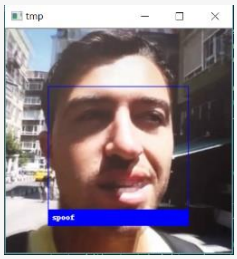
開發語言	Python	訓練模型	SeNet-154（DNN）

成　　果

使用 DNN 和遷移學習，藉由 SeNet-154 做基底 model，辨識出圖片中人臉的真假，達到94.09%的準確率。



▲ **成果一：框出圖中人臉**



▲ **成果二：顯示真臉或假臉**

實驗方法與結論

方　　法

成果一：直接使用 dlib 的套件。辨別圖片中否有人臉。讀取圖片，再查詢圖片中的臉部，查詢出臉部的編碼後，便根據以紅色方框將臉部框出，並呈現出來。

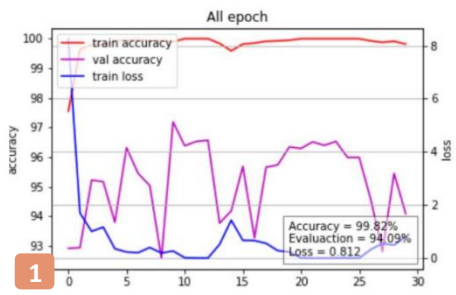
成果二：**判斷為真人臉或偽造影像人臉**。綜合評估和訓練挑選 Loss 較低的模型，將預測結果與實際標籤做對比。預測正確使用藍色方框標示如右圖 1，反之以紅色方框標記。主要分為：

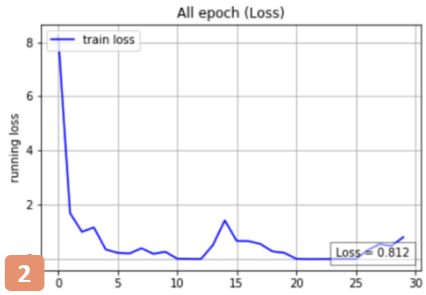
A . Training 使用 real 1223 張和 spoof 7076 張，將正規化後的資料打亂順序並導入 SENet-154 模型訓練。在每次的迭代過程計算 Loss 值並將梯度更新。

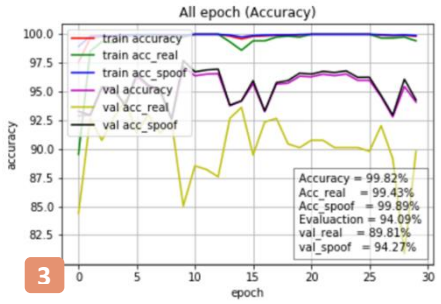
B . Evaluation 將預測每張照片的成果（real or spoof）與原先所屬標籤進行比對，評估準確率。

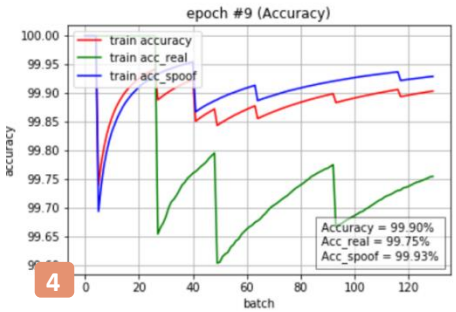
C . Detection 綜合訓練和評估挑出了最後 detection 的模型。如果照片原標籤等於預測結果時會使用藍色方形框住人臉表示預測正確，反之預測錯誤則是用紅色方框。

實驗結果









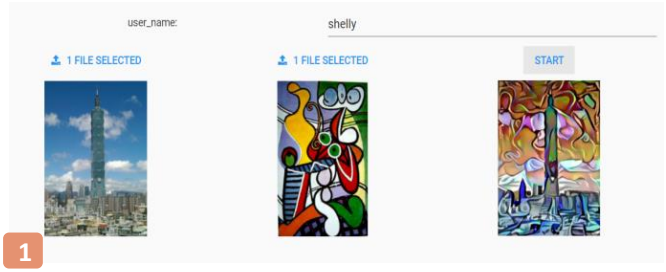
結　　論

從圖 2 可看出從 epochs=12~20 之間震盪，應選擇其他範圍的 epoch 數值（如 5~10 或 20~25），Loss 會較小。圖 3 可看出當 epoch 越大，real 的 Accuracy（黃綠線）上下震盪幅度越大。圖 4 後段 real 之 Accuracy 上下大幅度震盪（綠線）。推測因為其 real 的資料太少，所以模型偏好去猜 spoof。

卷積神經網路 + 資料庫設計 — 模擬風格轉換平台	
開發語言	Python、PHP、JavaScript、CSS、HTML
應用環境	YOLOv3、Tkinter、Apache、MySQL workbench
訓練模型	**VGG19**
成　　果	製作個人風格照片的平台。設置轉換來源照片與風格圖片，系統連結資料庫將變更後的成品儲存於獨立網站。

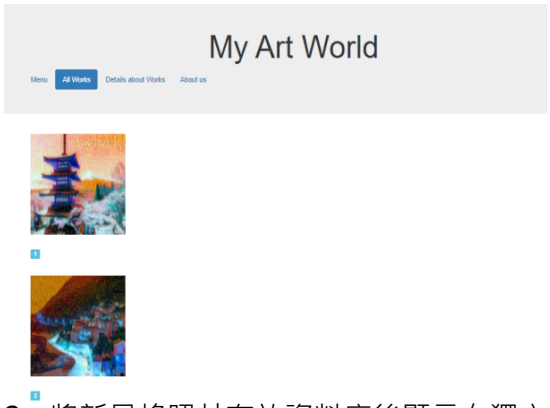
專題內容

實作方法



▲ 圖 1：轉換來源照片 + 風格圖片 → 新風格照片

(使用 Tkinter 製作圖形介面平台)



▲ 圖 2：將新風格照片存放資料庫後顯示在獨立網頁

- Style transfer :
  1： 神經風格轉移通過定義兩個損失函數來完成，這些損失函數試圖最小化內容圖、風格圖和生成圖之間的差異。
  2： 計算一個來自生成圖和風格圖的風格矩陣。風格矩陣被用來尋找卷積層的特徵圖之間的相關性。
  3： 得到生成圖像的梯度後可以通過使用 Gradient descent 來評估模型。
  4： 將每個迭代中生成的圖像儲存在資料夾中。
  5： Tkinter 製做轉換界面上，如圖 1。

- WampServer 個人伺服器建立資料庫存放變更後的照片 ：
  1： 當所有迭代完成時，觸發按鈕將最終生成圖上傳至資料表
  2： 資料表更新後將自動顯示作品於網頁上，如上圖 2

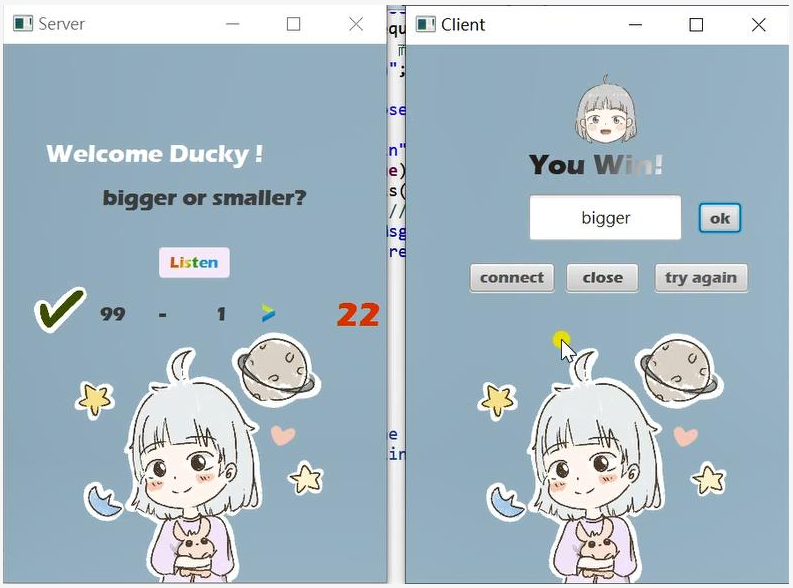
Socket 程式設計 — 數字骰寶

專題架構	Client – Server （Socket）
專題主體	Java、JavaFX、Scene Builder（使用者介面）
簡　　介	Server 與 Client 互相通訊並使用 multi-thread 技術： **Server →** 在設定隱藏數字後依序接收 Client 來訊並作出相對回應 **Client →** 告知 Server 運算符號、數字、大於小於符號，賭看看比隱藏數字大或小

影片 QR code
6:00 分 DEMO 開始

專題內容

遊戲簡介



▲Server 端　　▲Client 端

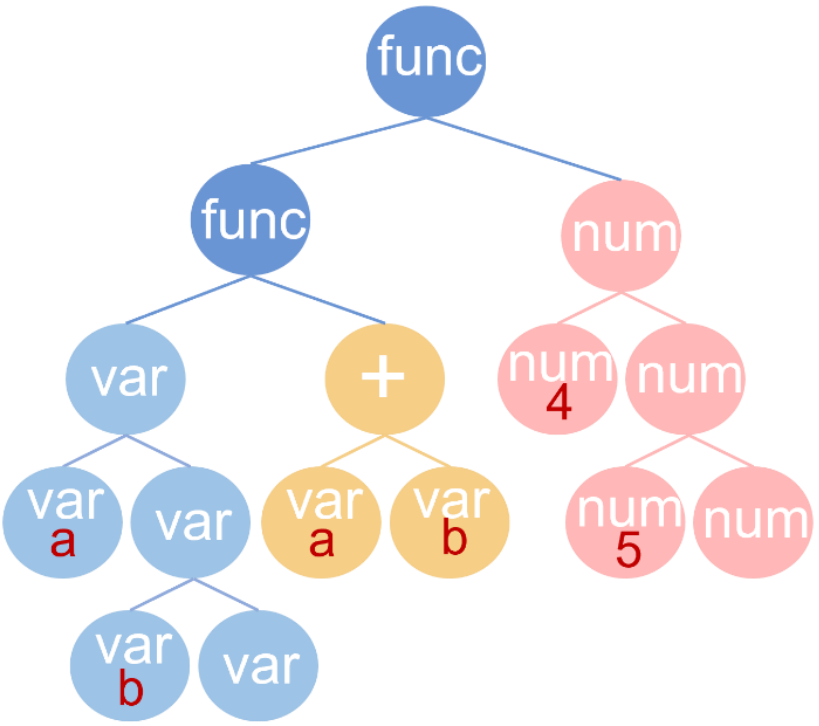
設計內容

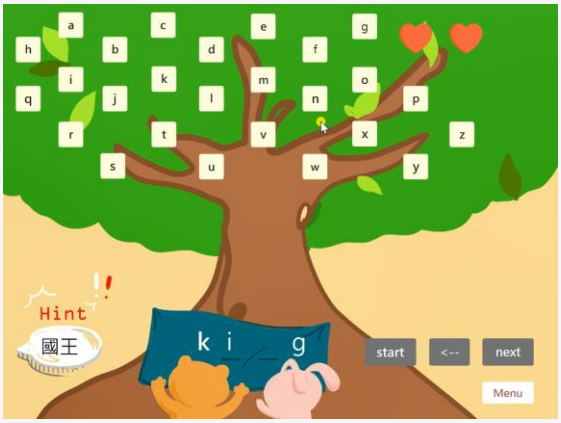
- **Socket應用設計:**

利用Server端與Client端利用Socket互相傳遞訊息，
Server接受連線請求後即開始遊戲，需要先輸入玩家名稱
[1.] 在連線啟動的同時會隨機產生目標數字flag(最後要比較的數字)
[2.] 此時Server會再隨機決定第一個數字A，接著Server與Client會透過一來一往來引導Client
[3.] 去讓Client決定要對數字A做加或減一個數字的動作
[4.] Client選擇完運算符號會需要決定要進行加或減的數字B(第二個數字)
[5.] 接著Client輸入數字後會接著被要求猜這個運算式(A +or- B)的結果會大於或小於目標數字flag,Client若猜對了(運算式成立)就表示贏了

連　　結

https://drive.google.com/file/d/1kST_WgFrgVT5wZhntLFzspdBuqGGpytk/view?usp=sharing

Compiler 設計 — Mini LISP	
開發環境	Ubuntu、VMware
開發語言	C++、Lex、Yacc
成　　果	一個 Compiler 從解析 prefix-notation 指令到能理解語法並輸出結果 1： 能夠判斷語法、列印、邏輯運算、數值運算 2： 能夠定義變數、函數、命名函數
專題內容	
實作方法	1： 使用 Lex 解析指令切割出不同形式的 Token (number、var、operator...) 2： Yacc 依據語法與給定 token stream 從底層節點 Bottom-Up 逐步建構 Abstract Syntax Tree (AST) 3： 完成 AST 後從 root 開始 traverse 各節點得到當前結果 4： 解析 Prefix-notation 指令 如：(Print-num ((function (a b) (+ a b)) 4 5)) 透過 Mini-Lisp grammar 分析後可得知其定義了 Function，需傳入兩參數 a、b，並將 a + b 傳回父節點 指令中設定將 4 、 5 傳入此 Function 得到 4+5=9，最後印出結果
範　　例	 ▲ 依指令建構之抽象語法樹 AST (Print-num ((function (a b) (+ a b)) 4 5))

Java 程式設計 — 挑戰單詞樹 Spelling Tree		
開發環境	Eclipse + Scene Builder（使用者介面）	
使用語言	Java、JavaFX	
備　　註	第一個程式專題	
簡　　介	使用 multi-threading 技術同步更新答題情況與生命值，設有三個等級，依照中文翻譯提示在大樹中找出對應的字母的單語拼音遊戲	影片 QR code
專題內容		
遊戲簡介	 ▲封面  ▲關卡內容 利用課堂中學習的程式基礎結合邏輯設計從零開始製作出這個小遊戲是設計給初學美語的小朋友，內容簡單明瞭，畫面清晰。	
設計內容	• **使用 JavaFX media：** 在開始前可將遊戲背景音樂開啟，並選擇三個選項之一（basic、medium 和 high level 分別代表三個字母、四個字母與五個字母的拼音） • **使用 multi - threading 跨執行緒 操控動畫與生命值：** 1： **動畫→**利用 Thread 同步偵測累積答對次數，達三次時「Excellent！」以動畫形式滑出 2： **生命值→**同步偵測答題情形，若未在提示下正確解答，將透過參數同步更新判定失去一條生命 • 遊戲進行中提供中文 Hint，依提示點選樹上字母來完成拼音，途中可修改答案 • 在三條命還沒用完的情況下完成該等級的所有問題即等級通關，可回主頁面進行下一次挑戰或選擇離開。	
連　　結	https://drive.google.com/file/d/1yuj3k_zVyJjgiQJaw1dJwygGBUWfGZZ0/view?usp=sharing	

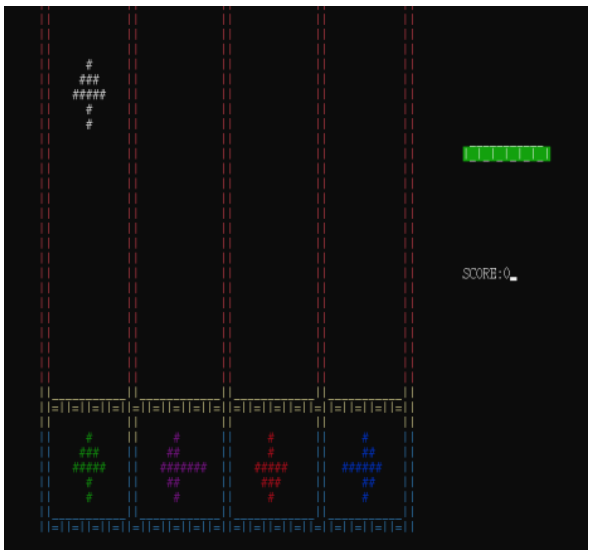
組合語言 — 跳舞機

開發語言	Assembly X86	
簡　　介	• 在箭頭落下到限制範圍內時，在鍵盤上按下對應的上下左右獲得分數，反之則扣分 • 其中設有紅蘿蔔補血與小怪獸陷阱增加趣味性	
技　　術	1： 使用**巨集**提升執行速度 2： 將座標值累加儲存於暫存器後不斷更新定位，使畫面動起來 3： **善加運用內建函式** → 綁定鍵盤、彩色符號、座標定位、隨機選取函數 4： 主要在 cmd 黑窗中呈現	影片 QR code

專題內容

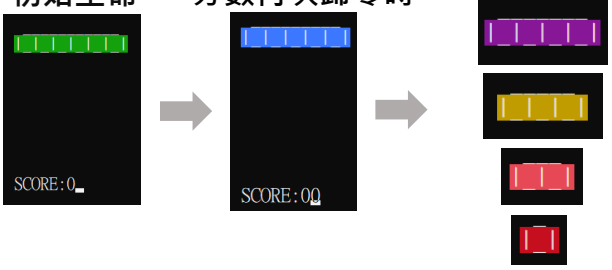
遊戲介面

▲起始畫面（運用屬性巨集設定座標與彩虹色彩 eg : WriteConsoleOutputAttribute）

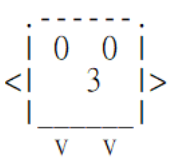


▲ cmd 遊戲進行畫面（綁定鍵盤、隨機選取巨集決定下一個箭頭、暫存器更新座標）

初始生命　　分數再次歸零時

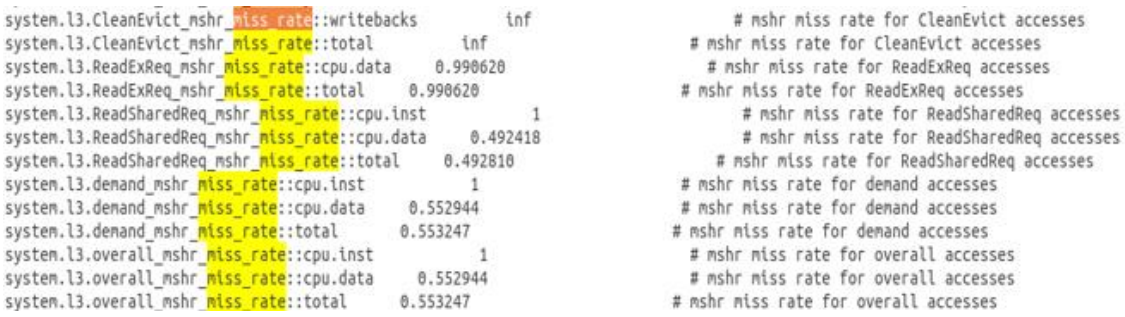
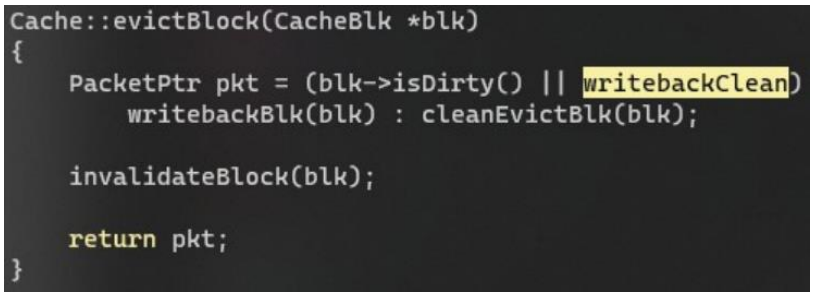


▲補血紅蘿蔔



▲小怪獸陷阱

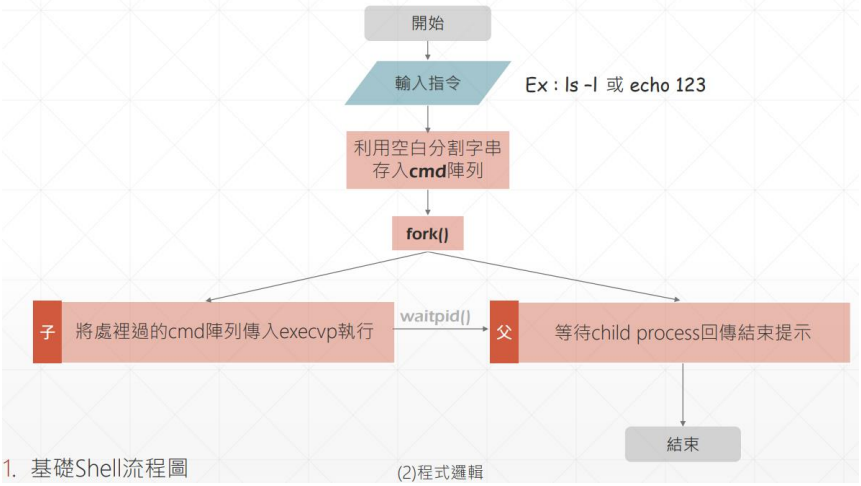
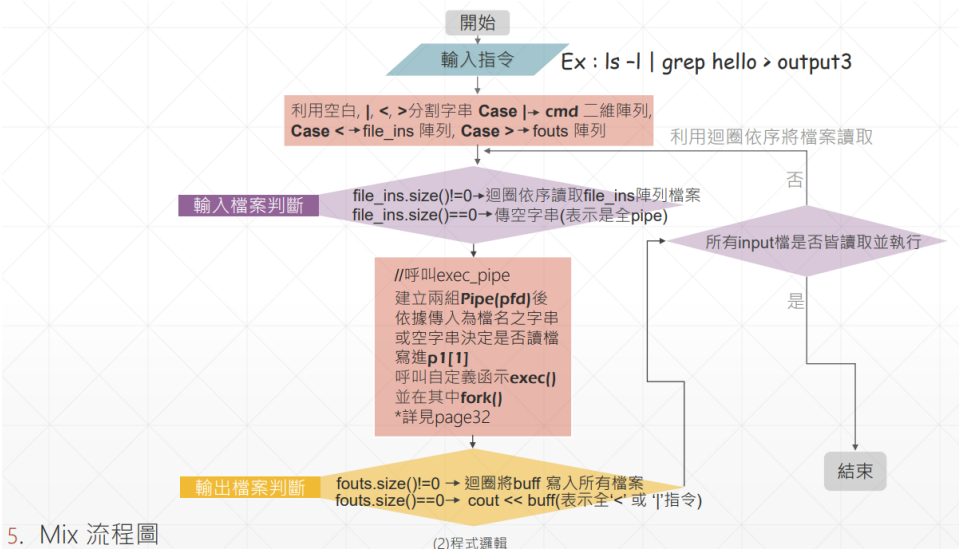
連　　結	https://drive.google.com/file/d/11jZlDhqstYUK1bqdi95iPQJe8OCzGzqy/view?usp=sharing

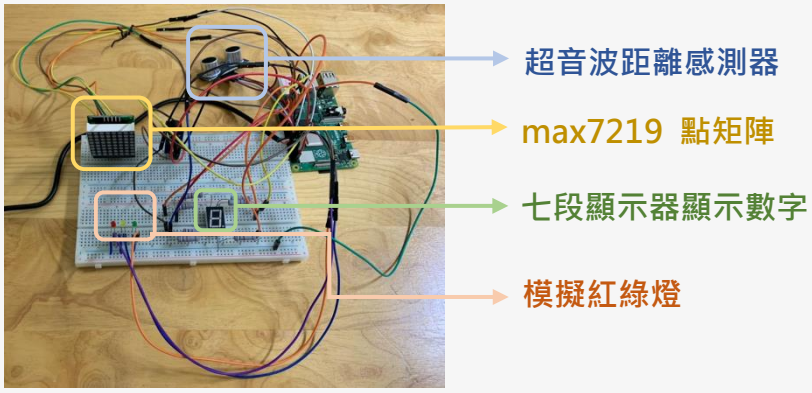
模擬記憶體設計	
開發環境	Ubuntu、VMware
模擬程式	NVmain + GEM5 聯合編譯
成　　果	1： 了解 CPU 和 Memory 基本架構關係，更改原本 GEM5 內建，在 L2 cache 的基礎上實作 L3 cache。 2： 調整參數觀察 set-associate 和 full-way 的 miss rate，理論上 full-way 的 miss rate 較低，表示其 hit rate 更高。 3： 置換不同的 Cache Policy 觀察效能(LRU→RRIP)。 4： 觀察 base.cc 和 cache.cc 的程式碼，參數 writeback_clean 控制了是否啟用 write back policy，若設置為 true 則為 write through policy，會直接把 data flush 進 memory。
專題內容	
實作頁面	 ▲ RRIP policy miss rate  ▲ 找到關鍵參數即可切換 Writeback / Writethrough
成　　績	91/100(91%)


```
Cache::evictBlock(CacheBlk *blk)
{
    PacketPtr pkt = (blk->isDirty() || writebackClean)
        writebackBlk(blk) : cleanEvictBlk(blk);

    invalidateBlock(blk);

    return pkt;
}
```

撰寫 Linux Shell Script	
應用環境	Ubuntu、VMware
背　　景	• 基於父子程序執行關係、pipe 串接、parse 指令 • 使用 C++在 Linux 系統下完成
成　　果	1： 基礎 Shell：能夠執行各種 Linux 的指令 2： Pipe (|)：能夠執行如下指令 ls | head -3、 ls | head -3 | tail -1 |<other cmd> | ... 3： Redirection to file (>)：能夠連續重導入 4： Redirection from file (<)：能夠連續重導出 5： 綜合所有功能
專題內容	
實作方法	 ▲ 基礎 Shell 流程圖  ▲ 綜合所有功能流程圖
DEMO 成績	100/100(100%)

超聲波 LED 輔助倒車系統		
課程名稱	資電跨系實驗	
專題主體	Python + Raspberry pi	
開發元件	樹莓派、超音波距離感測器、蜂鳴器、點矩陣	
成　　果	• 透過超聲波距離感測器偵測物體靠近程度，由近至遠亮起紅、黃、綠 LED 燈 • 同時伴隨蜂鳴器發出不同頻率警示聲 • 數字顯示器顯示剩餘距離	影片 QR code
專題內容		
設計理念		
實作內容	• 迎接駕駛**提示燈**：max7219 點矩陣模擬駕駛剛進停車場的時候，顯示字串”Welcome To Parking Lot” • **超聲波感測器**：超聲波感測器 + 七段顯示器顯示數字 當物體距離剩餘 50 公分時，七段顯示器即亮起 5，剩餘 40 公分時，即顯示 4，以此類推顯示 3、2、1。 • **模擬紅綠燈**：超聲波感測器 + led 燈*3 利用紅、黃、綠三種顏色來提醒駕駛人目前距離大概還有多遠， 剩 50~30 公分會亮綠燈，剩 30~10 公分會亮黃燈，小於 10 公分時就會亮紅燈提醒駕駛要停下了。 • **蜂鳴警示**：蜂鳴器 + 超聲波感測器除了有亮燈的提示，為了預防有一些視線不佳的情況，所以加裝蜂鳴器發出聲音警示駕駛人，當距離剩越小，聲音的頻率變 高，小於 50 公分會發出 Me，小於 40 公分會發出 Fa，小於 30 公 分會發出 So，小於 20 公分會發出 La，小於 10 公分會發出 Si。	
連　　結	https://drive.google.com/drive/folders/1emE3fesyD3OYNeuDESZYzVC2JXEkdjoF?usp=sharing	