

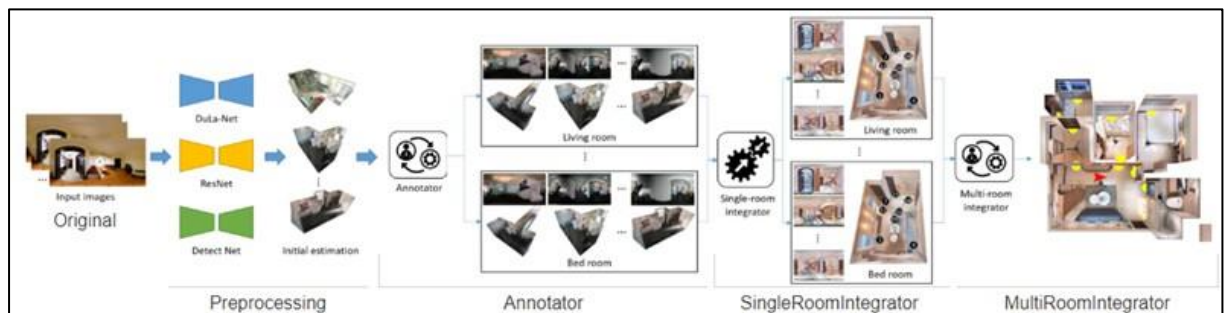
專題

● 3D 實境賞屋

◆ 專案介紹：

改變舊有觀看 2D 照片的賞屋方式變成更實境化的 3D 模型，透過清大朱宏國老師團隊開發的 Annotator、SingleRoomIntegrator 等工具輸出的房屋標記資訊將全景房屋照片貼在 3D model 上。產生 3D 房屋場景。在由我們負責的部分 MultiRoomIntegrator 的 Editor 將每間房屋做半自動拼接組合後，將拼接資訊記錄下來在 viewer 輸出。Viewer 讓攝影機在 model 中轉換站點、穿梭其中。系統架構如下<圖 1>。

此系統讓使用者不用觀看 2D 平面房屋照片，更不用到現場實際觀看，就能有身歷其境之感。房仲業者及消費者能透過手機或電腦操控攝影機位置，隨時隨地皆可實境賞屋，讓過程更方便化與實境化。



<圖 1>系統架構圖

◆ 開發工具：

1. JavaScript
2. WebGL

◆ 主要技術：

1. Three.js library 應用
2. Shader 貼圖應用
3. 3D 場景 Camera controls 應用
4. Panorama、CubeMap 應用
5. Json 資料存取、解析

◆ 遭遇問題及心得：

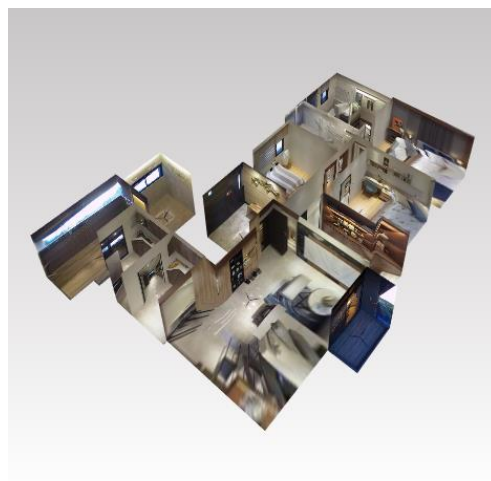
第一次接觸 Three.js 這個 library，從一開始只有產生簡單房間方塊 geometry 再變成將 3D model 輸出並貼上全景圖 panorama，利用 json 檔案傳輸標記資訊，利用不同 camera 來控制場景畫面，更豐富使用體驗。

過程中修改過多次讀取 panorama 方式，因為 panorama 照片原始是 8k 圖片，全部讀到記憶體中非常吃效能，會讓整個網頁 crash 掉，後來修改讀取圖片的流程架構，改用 cube map buffer 概念來讀取圖片，需要用到的才讀入記憶體，不需要則 release。也遇過 IOS 平台 WebGL 支援 texture ID 問題，後來利用合併 texture 的方式，將多張圖片合併成一張 texture 讀入，減少 texture ID 來解決問題。

◆ 成果展示：



<圖 2>TopView 1



<圖 3>TopView 2



<圖 4>DownView

● 拉霸遊戲-沙灘甜心

◆ 專案介紹：

一款具有多種華麗表演的拉霸遊戲，除了主遊戲外還有另外機率觸發的 feature game、free game，皆以不同方式表演呈現。專案開發將原本在 APP 上完整移植到網頁 H5 遊戲，利用粒子系統及動畫系統做出轉場效果、報獎效果等，豐富畫面美術感，吸引玩家遊玩。

◆ 開發工具：

1. Cocos Creator
2. TypeScript

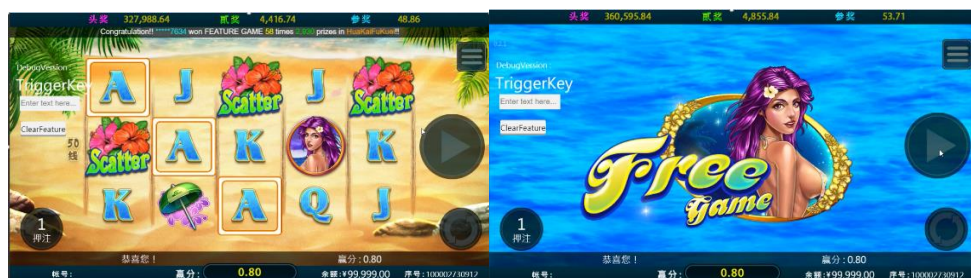
◆ 主要技術：

1. Particle system 應用
2. Animation system 應用
3. Spine 骨骼動畫應用

◆ 遭遇問題及心得：

使用動畫系統做出海浪轉場動畫，依照現有的兩張海浪美術圖必須做出海浪不停在跑的感覺，調整時機點及飛入畫面方式來完成。因為 Cocos Creator 的粒子系統有點難調整，後來利用調整粒子 life time、start size 並結合動畫系統來做出光線彈射的效果。學會動畫系統及粒子系統的使用能夠使遊戲更豐富。

◆ 成果展示：



<圖 1>遊戲主畫面

<圖 2>海浪轉場



<圖 3>光線彈射

<圖 4>排球彈跳動畫

● 棋牌遊戲-骰寶

◆ 專案介紹：

一款多人連線下注遊戲，遊戲過程包含下注、開骰、派獎等階段，每階段皆會做不同表演，利用美術圖及動畫系統結合做出桌子閃亮動畫，及使用 spine 動畫加上 action system 的淡入淡出效果做出開骰動畫，讓遊戲過程更加有趣、畫面效果更呈現更完美。

◆ 開發工具：

1. Cocos Creator
2. TypeScript

◆ 主要技術：

1. Action system
2. Particle system
3. Spine
4. MVC model

◆ 遭遇問題及心得：

開發時須遵照 MVC 及動態機架構開發，在遊戲階段轉換過程中，時常會沒有清空 view 裡面的 component，導致下次在進入同狀態時，畫面顯示會有所錯誤，後來利用 state initial 及 release 階段做畫面 reset 處理來解決。

◆ 成果展示：



<圖 1>遊戲主畫面



<圖 2>開骰畫面



<圖 3>提示動畫



<圖 4>桌面閃亮畫面(開獎)

● 條碼應用軟體

◆ 專案介紹：

透過 openCV 開發的條碼應用軟體，透過圖象切割後進行條碼辨識，中間使用過 homography、binarization、enhancement 做處理，做中進行定位等等的技術，丟到最後面辨識的區塊去解出條碼。

◆ 開發工具：

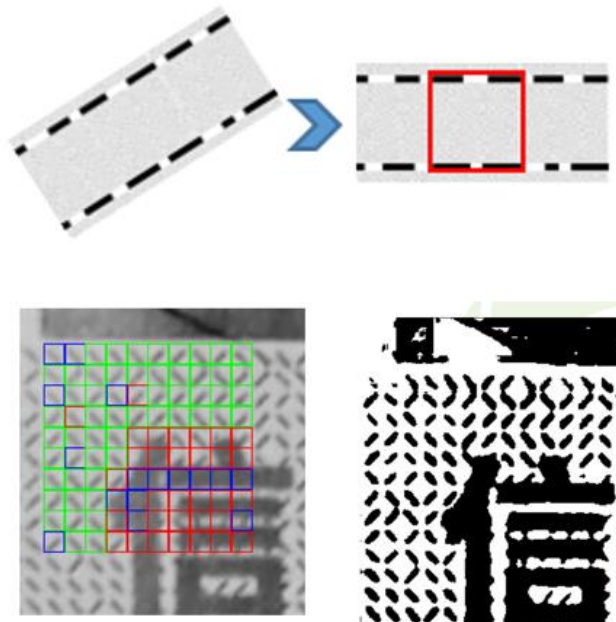
1. C++
2. OpenCV

◆ 主要技術：

1. 影像處理
2. 圖象切割辨識

◆ 遭遇問題及心得：

開發過程與團隊合作進行，討論各 component 定義，如何進行切割處理及辨識，遇到困難部分主要是圖象模糊，導致定位等切割位置問題，並且在圖象 homography 時也會有 pixel 跑掉等問題。



課堂相關作品

● 多媒體實習-小頭歷險記遊戲

◆ 作品介紹：

一款單機互動打擊殞石遊戲，利用 kinect 攝影機偵測人體骨架位置，手臂控制武器攻擊迎面而來的各種殞石，限時內獲得高分獲勝，若打到危險殞石則扣血量。雙手比出勝利姿勢釋放技能，清除畫面物品。利用 Particle System 及 Sky Box 讓畫面效果融入太空主題。

◆ 開發工具：

1. Unity
2. Kinect 深度攝影機

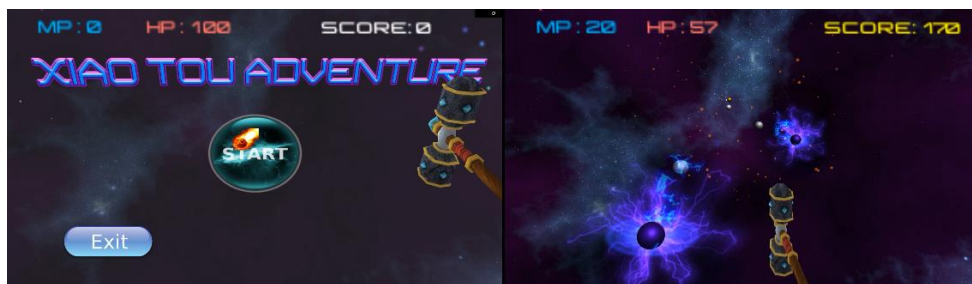
◆ 主要技術：

1. Kinect API 應用
2. Unity 開發及程式應用

◆ 遭遇問題及心得：

在隨機產生殞石 prefab 的參數調整上，透過試玩幾次來做調整到較好的參數，隨著關卡做不同的變化，讓難度有所改變。不會讓遊戲呈現單一的情況。Kinect 原本偵測座標軸與遊戲不同，導致控制武器錯誤，最後將軸向調整為相同方向解決。

◆ 成果展示：



<圖 1>遊戲開始畫面

<圖 2>遊戲主畫面



<圖 3>施放技能 1

<圖 4>施放技能 2

● 遊戲程式設計-3D 機關迷宮

◆ 作品介紹：

一款收集道具避開機關的迷宮小遊戲，利用光線效果及 skybox 營造詭譎氣氛，飛鏢、地刺等機關增加遊戲難度，角色可施放技能無敵。組合粒子效果來呈現技能。

◆ 開發工具：

1. Unity

◆ 主要技術：

1. Particle system
2. Animation system

◆ 遭遇問題及心得：

學會利用 SpotLight、PointLight 來做光影效果，調整光線的大小及強度等等來做出符合遊戲主題氣氛的感覺，利用 SpotLight 作為類似角色的手電筒感覺，讓畫面中能見度降低，提高遊戲性。

◆ 成果展示：



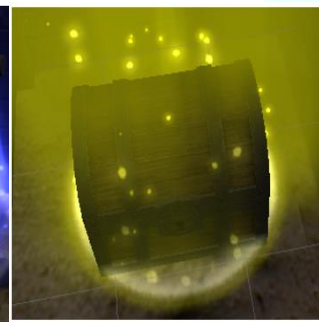
<圖 1>遊戲地圖



<圖 2>遊戲主畫面



<圖 3>無敵技能



<圖 4、5>地圖物件

● 遊戲程式設計-狐狸打排球(連線遊戲)

◆ 作品介紹：

一款雙人連線對戰遊戲，兩邊個別操控角色進行排球對戰。利用 Area effector、Point effector 等 2D 物理來增加遊戲性，黑洞會吸引排球，箭頭則會給予排球推力，讓遊戲增加難度與趣味性。利用動態產生 prefab 做出殘影效果，讓畫面更加炫酷。

◆ 開發工具：

1. Unity

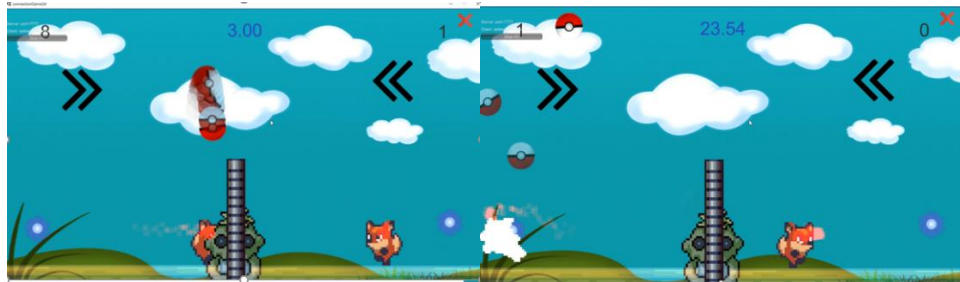
◆ 主要技術：

1. Unity 連線 API
2. 2D 物理
3. shader

◆ 遭遇問題及心得：

第一次做連線小遊戲，在兩個 client 會發生畫面不同步問題，導致對戰遊戲無法順利進行，後來使用 Network Manager API 解決此問題讓連線畫面可以同時更新。

◆ 成果展示：



<圖 1>遊戲主畫面

<圖 2>角色碰到球閃爍



<圖 3>排球殘影

<圖 4>狐狸紅眼(shader)

● 遊戲程式設計-淡水阿嬤的海盜桶(手機遊戲)

◆ 作品介紹：

一款手機派對破冰小品遊戲，利用手機 G-sensor 來做為遊戲主要控制，改變武器大小或更換武器皆由手機 sensor 做互動，並且做出環景效果符合海盜桶主題。

◆ 開發工具：

1. Unity

◆ 主要技術：

1. Gyro sensor API
2. Accelerometer sensor API
3. Multi touch

◆ 遭遇問題及心得：

設計時為了將手機 sensor 與遊戲做關聯，後來決定用 gyro 陀螺儀來作為環景效果，accelerometer 加速感測器作為更換武器的控制方式。並利用 multi touch 的方式來改變武器大小。

◆ 成果展示：



<圖 1、2>遊戲主畫面



<圖 3>排行榜紀錄

● 遊戲程式設計-荒廢城市(FPS 遊戲)

◆ 作品介紹：

第一人稱射擊遊戲，逃避隨機生成的狼群，狼群會自動追逐玩家，限時內玩家須透過有限子彈擊殺狼群，並在固定範圍內避開障礙物，時間結束存活者獲勝。

◆ 開發工具：

1. Unreal

◆ 主要技術：

1. Animation tool
2. Skeleton assets

◆ 遭遇問題及心得：

第一次使用 Unreal 開發遊戲，學會利用 Blueprint 已經寫好的程式庫來做判斷處理，少去自己寫程式的開發時間，並且學會利用不同的 user input 來做角色 control。

◆ 成果展示：



<圖 1>遊戲主畫面



<圖 2>遊戲場景



<圖 3>角色受傷反饋畫面

● 遊戲企劃與設計原理-Boooooom

◆ 作品介紹：

一款多人遊玩派對遊戲，地圖兩邊不同隊伍控制角色丟出炸彈、觸碰砲台或佔領旗幟等可以佔領地圖區域，限時內佔領較多區域隊伍獲勝，多種地圖物件皆為不同效果，兩種以上物件組合出不同攻擊策略，豐富遊戲互動性和趣味性。以趣味的背景音樂及Q版畫面來呈現一款派對同歡趣味遊戲。

◆ 開發工具：

1. Unity

◆ 主要技術：

1. Shader 改變地圖顏色
2. User controls

◆ 遭遇問題及心得：

課堂上從發想、分工、設計、實做到展示，都必須有四人團隊共同努力，在設計遊戲時，必須構想到遊戲機制、遊戲目標等等，花了好幾天的時間來規劃，團隊中每個人都必須做出貢獻，互相幫助努力才能有所成果。

◆ 成果展示：



<圖 1>遊戲進入畫面



<圖 2>遊戲場景



<圖 3>遊玩畫面



<圖 4>勝利畫面

● 基礎電腦圖學-3D 機器人影像處理

◆ 作品介紹：

利用 fragment shader 對影像做處理，將畫面轉黑白、灰階、夜視鏡效果及各種不同 filter 效果，並且套用 skybox 在場景中。控制機器人做出行走動作。

◆ 開發工具：

1. OpenGL

◆ 主要技術：

1. Vertex shader
2. Fragment shader
3. Skybox

◆ 遭遇問題及心得：

初次學習 shader 來做影像處理，一開始只會套用灰階公式來做處理，上網學習人家的 shader 寫法。將 RGB 值乘上數值再留下 G 值，並且在畫面上加上線條，做出類似夜視鏡掃描效果，如下〈圖 4〉。

◆ 成果展示：



〈圖 1〉Skybox



〈圖 2〉Fragment shader



〈圖 3〉Fragment shader



〈圖 4〉夜視鏡掃描效果

● 基礎電腦圖學-2D 小飛機遊戲

◆ 作品介紹：

一款 2D 飛機躲避殞石遊戲，利用連續圖做出飛機飛行動畫、殞石轉動動畫、碰觸爆炸效果。並且使用 shader 做畫面處理，當撞到越多次數的殞石，會有一圈黑框慢慢往內靠近，讓可視範圍縮小，增加遊戲難度。另外也使用粒子效果作畫面裝飾。

◆ 開發工具：

1. OpenGL

◆ 主要技術：

1. Fragment shader
2. 連續圖動畫

◆ 遭遇問題及心得：

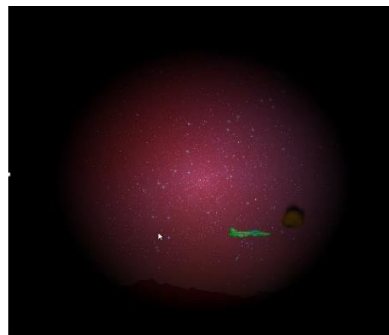
開發過程中發想是否能夠使用 shader 與遊戲做互動效果不只是做裝飾美化而已，後來決定利用整個畫面漸層黑掉的感覺來做受傷動畫，畫面越來越小並轉紅。

◆ 成果展示：



<圖 1>遊戲主畫面 1

<圖 2>遊戲主畫面 2

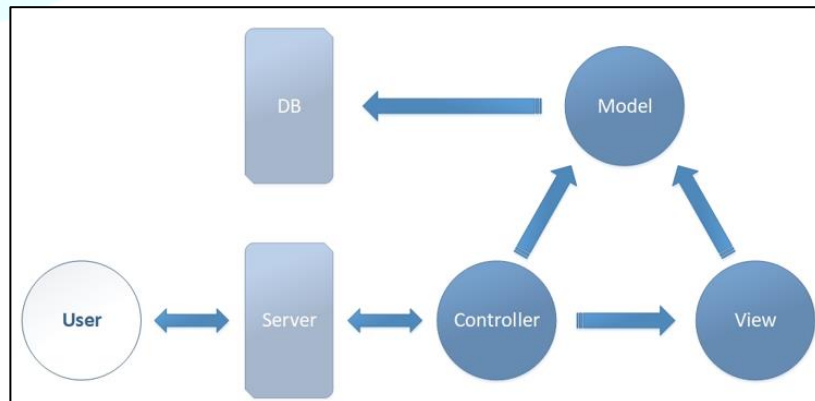


<圖 3>shader 畫面黑掉效果

● 軟體工程- T-Page

◆ 作品介紹：

簡單頁面的看板網站，類似 BBS 站，有家族及好友系統，並且可以在看板分類下發文、按讚等等，看板內會顯現熱門文章及分類，皆連結 Database 來做資料更新，介面簡單方便使用者操作。系統架構圖如下<圖 1>。



<圖 1>系統架構(MVC Model)

◆ 開發工具：

1. Cocos Creator

◆ 主要技術：

1. MySQL
2. TypeScript
3. MVC Model

◆ 遭遇問題及心得：

使用 MVC 架構開發系統，設計上必須分開三個部分開發，在之後維護或增加功能比較好做處理，也不會程式碼都全部擠在同一個 script 裡面。在開發上也較能分工合作，與團隊共同開發。

◆ 成果展示：



<圖 2>看板列表



<圖 3>管理者頁面

● IOS 程式設計- 阿捏 Cal 健康(APP)

◆ 作品介紹：

APP 供使用者輸入身高、體重及年齡等基本資訊，給予使用者建議每日攝取熱量，計算出基礎代謝率。並且能讓使用者輸入每日飲食資料，從資料庫中抓出熱量，並記錄回資料庫中，供使用者查詢歷史紀錄。

◆ 開發工具：

1. iPhone Xcode

◆ 主要技術：

1. iPhone Xcode 程式開發
2. 資料庫存取
3. APP UI 設計

◆ 遭遇問題及心得：

首次開發 IOS APP，學習如何使用 xcode 開發，利用裡面 table view、data picker、pick view 等工具來做畫面設計，並讀取資料庫裡資訊顯現出歷史紀錄，畫面美術設計上略顯不足，功能上可正常使用。專案開發之後必須找美術專業人員合作，才能做出更完整、更精緻的軟體。

◆ 成果展示：



<圖 1>個人資料頁



<圖 2>每日熱量



<圖 3>歷史紀錄