

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

逢 甲 大 學 3D 校 園 導 覽 製 作



學 生：湯 竣 杰 (四 甲)
陳 思 鴻 (四 甲)
翁 國 淵 (四 甲)

指 導 教 授：謝 信 芳

中 華 民 國 九 十 三 年 五 月

目 錄

圖 表 目 錄.....	V
第一章 導 論.....	1
1.1 背景.....	1
1.2 動機.....	2
1.3 目標.....	2
第二章 系統概述.....	5
2.1 虛擬校園及 3D 歷史背景.....	5
2.2 3D 圖形化的理念.....	7
2.3 軟體開發工具.....	8
第三章 系統分析與設計.....	9
3.1 系統開發概述.....	9
第四章 系統功能架構與作業流程.....	11
4.1 系統功能架構.....	11
4.2 作業流程.....	13
第五章 開發工具.....	14

5.1 3D STUDIO MAX	14
5.2 3D GAME STUDIO	15
第六章 系統實作解說.....	19
6.1 模型實作解說.....	19
6.1.3 建築物製作.....	21
6.2 路徑導覽實作解說.....	25
6.2.1 校園分區編碼.....	25
6.2.2 設立站點.....	28
6.2.3 終點站.....	29
6.2.4 非終點站.....	29
6.3 導覽流程的程式控制.....	31
6.3.1 從原地到目的地.....	31
6.3.2 半途更改目的地.....	32
6.4 導覽攝影機程式解說.....	34
6.4.1 導覽攝影機視角遠近切換.....	34
6.4.2 導覽攝影機定位方向切換.....	35
6.5 按鈕程式解說.....	37
6.6 聲音程式解說.....	39

第七章 使用者介面說明.....	40
7.1 自動導覽.....	41
第八章 效能評估.....	46
8.1 系統評估.....	46
8.2 功能評估.....	47
第九章 結論.....	48
9.1 組員心得.....	48
9.2 未來展望.....	52
參考資料.....	53

圖表目錄

圖 4.1 初擬架構圖	11
圖 4.2 完整架構圖	12
圖 4.3 作業流程圖	13
圖 5.1 3D GAME STUDIO 引擎架構圖	15
圖 5.2 WED 子系統	16
圖 5.3 MED 子系統	16
圖 5.4 繪圖程式	17
圖 5.5 動作語法	18
圖 5.6 動態函式庫	18
圖 6.1 Model Editor 支援曲面動畫和頂點動畫	19
圖 6.2 紋理貼圖	20
圖 6.3 真實圖與 3D 圖的對應	21
圖 6.4 未加入材質的構造圖	21
圖 6.5 加上材質後的預覽(右上角)	22
圖 6.6 材質選項	23
圖 6.7 忠勤樓架構	23

圖 6.8 忠勤樓 3D 外觀.....	24
圖 6.9 忠勤樓教學大樓和語文大樓的平面圖.....	24
圖 6.10 未編碼的校園圖.....	26
圖 6.11 已編碼的校園圖.....	27
圖 6.12 終點站和非終點站.....	29
圖 7.1 系統畫面.....	40
圖 7.2 是否確定自動導覽.....	41
圖 7.3 選擇內部或外部導覽.....	41
圖 7.4 點選目前位置.....	42
圖 7.5 點選前往位置.....	42
圖 7.6 開始導覽畫面.....	43
圖 7.7 到達目的地顯示畫面.....	43
圖 7.8 視野切換畫面.....	44
圖 7.9 鳥瞰圖(1).....	44
圖 7.10 鳥瞰圖(2).....	45

第一章 導論

1.1 背景

在一所廣大的大學校園裡頭會有各式各樣的建築物。設想，今天你是一位外賓或新生，想要參訪大學的各部門，面對一棟棟陌生的建築物，該從何找起呢？如果在校門口備有一台內建導引系統的電腦裝置，只要你輸入你想去的部門，然後透過螢幕指引你從目前位置走到該部門位置的方向，是不是很方便？於是我們了解到校園導覽系統的需要性，便想利用這次專題發展此系統。由於學長們已經開發了一半，我們就決定延續學長們的系統，來進行系統改良。這套系統主要是以3D軟體製作，對於我們這組來說，即陌生且有趣的，因為我們從未學過3D製作的軟體，但是又喜歡玩3D遊戲。這套系統目前只做到校園外圍導覽和商學大樓導覽，我們希望在這段相當緊迫的日子，可以做到其他主要建築物內的導覽，諸如行政一館、行政二館與資電館等。

1.2 動機

老師之前給了我們幾個題目做選擇，最終覺得類似 3D 遊戲的校園導覽較引起我們的興趣，況且我們都有常玩 3D 的遊戲習慣，對其製作都頗感興趣。之後延續了上學期學長們的專題題目，並加以改良來滿足自己製作的渴望。

1.3 目標

之前學長們所開發這套系統的功能有：

1. 提供多視角導覽。
2. 環境建設真實化。
3. 目的地路徑搜尋功能。

我們希望能夠完成以下三個目標：

1. 改善人物動畫。
2. 改良系統的導覽畫面。
3. 完成校園所有建築物的室內導覽。

前兩項應該不難達成，至於第三項對我們是一大挑戰。相信有學長的

技術支援和老師的指點，我們可以更具信心完成。底下為學長與我們

專題的比較表：

	之前學長們的專題	我們的專題
定向模式	在導覽過時人物的轉動會使拍攝鏡頭也跟著轉動，使用者容易產生混亂。	可採用第三人稱的定向模式作導覽，讓使用者方便記得路徑。
自動導覽	是從地圖上任何一個點開始出發。	採用點對點方式，從一個建築物到另一個建築物。
更改目的地	在自動導覽開始到結束前無法更換目的地。	在自動導覽的過程中可以隨時更換目的地。
校園環境佈置	對建築物外的環境佈置較講究。	對建築物外的環境佈置較粗糙。
鳥瞰模式	沒有實作對整個校園環境的鳥瞰。	實作了對整個校園環境四個角度的鳥瞰。
導覽時拍攝方向	在導覽時是從人物背後拍攝。	在導覽時可以從人物上空拍攝下來..
建築物模型	大略地根據建築物輪廓來	根據真實建築物外形來實

	實作.	作.
自動導覽程式 錯誤	有些建築物點選了卻到達 不了.	可以點選的建築物皆可以 順利到達.



第二章 系統概述

2.1 虛擬校園及 3D 歷史背景

目前在網路上所能知道的虛擬導覽系統，大多以網路服務(Web Service)為主，可分為四種類型。其一是以媒體播放系統結合其他 3D 動畫工具的製作，其二是以 VRML 方式製作，最後兩種則使用 Virtools 和 EON 工具來開發展現的。逢甲大學現有的虛擬校園系統就是利用第一種類型完成，把整個校園分成數個區域，點選所要的區域後就會在線上即時播放預先攝影該區域的環境片段。VRML(Virtual Reality Modeling Language)類似 HTML (HyperText Markup Language) 是描述一個虛擬的空間。例如，文件中有一個真實樣子的佈告欄，靠近它可以看到最新的網路新聞(Netnews)。

Virtools和EON是發展出來的3D／VR軟體。Virtools係由法國開發出來的軟體，目前在台灣由愛迪斯科技公司代理，其最有名的實作案例便是國家歷史博物館的3D文物展覽，使用者只要上網點選文物照片便可以瀏覽各式文物的3D影像。Virtools能製作出3D文物的移動、旋

轉和翻轉動作等視覺效果佳。EON Studio是由瑋特擬真科技公司發展出來的3D／VR軟體，就整體功能上比Virtools來的強，不過可惜的是這套軟體價高二十萬，試用版功能偏少，且每次連上相關作品網站時所需下載的模組零件過多，耗時耗工，單以功能性來說是個強而有力的平台。

我們曾嘗試過P C遊戲的經驗，構思並實作出這套逢甲大學3D校園導覽系統。我們所使用的工具是3D Game Studio再加上一些製作3D特效的補助軟體，它是個免費的軟體，但補助軟體是試用版本。

2.2 3D 圖形化的理念

採用3D介面處理的理由是現今的電腦大多可以支援高載量的3D圖形運算，再加上所有視角以3D畫面來處理不會有任何死角產生，人可以各個角度觀看四周的環境。再加上，現今的P C遊戲軟體都是以3D為主軸，美麗生動的畫面成了玩家購買的基本要素。2D視角的只能固定一個角度觀看，用連續貼圖方式做成環繞一個點的俯視觀察點，線上遊戲RO就是採用這種貼圖技巧。

從以上的解釋可見3D化的好處，在3D模擬的環境下可將建築物模型渲染出真實之質感並且可以設計出更符合我們的設計概念的模型。如果在一間大廈的修繕過程中，能夠先以3D模型來模擬所有建築物間隔與機械系統、空間規劃以及修復過後的大廈狀況。如同現今房屋仲介一樣，那豈不是一件美好的事物；相反的，若是以傳統2D平面建築設計圖，密密麻麻的線條和一些無法解釋的數字，是一般人根本不能了解的，只有專業的設計師才看得懂。

2.3 軟體開發工具

1. 3D Game Studio
2. 3D MAX Studio
3. PaintShop Pro
4. ACDsee



第三章 系統分析與設計

3.1 系統開發概述

本系統是以使用者為導向，如果有好的介面設計，那麼能導引使用者做自然律動的交談模式。反之，則會對使用者產生“不友好”的交談方式。

當初在設計的時候，以人的直覺為考量對象。人的直覺是一種最終使用者記在腦中的系統映像。它請求使用者對使用過後的操作描述，透過這種方式便可了解該軟體在被使用時是否能讓使用者容易地操作。太過複雜的操作系統會讓初學者不易上手，也因此設計模型必須要開發成能夠容納包含在使用者模型中的資訊，使系統映像正確地反映有關介面的語法和語義資訊。

一般軟體工程師先定義並且把工作分類，介面設計就可以開始了。在介面設計過程中可以使用以下的方法來完成：

1. 建立工作的目標和計劃。
2. 把每個目標／計劃轉成具體的按鈕。
3. 指出系統的狀態，例如，按鈕在被執行的時候，它看起來會是怎樣的介面？

4. 定義控制機制，例如，當系統狀態改變時，有那些按鈕是使用者可以使用的。
5. 控制機制如何影響系統的狀態。
6. 指出使用者如何由輸出結果得到所要的資訊。

人機介面設計需要理解介面技術、人的直覺、使用者的熟練程度、行為模式以及使用者必須實施的總任務是介面設計中的所有因素。諸如應答時間、命令結構、錯誤處理等設計問題都需被考慮在系統的設計模型中。



第四章 系統功能架構與作業流程

4.1 系統功能架構

本專題我們是針對學長之前的系統來重新實作並改良，但是經過不斷的研商，我們覺得如果沒定下太多功能的話，由於技術尚未了解，會導致不著邊的下場，所以我們決定以幾項簡單的基本功能。其架構圖如下：

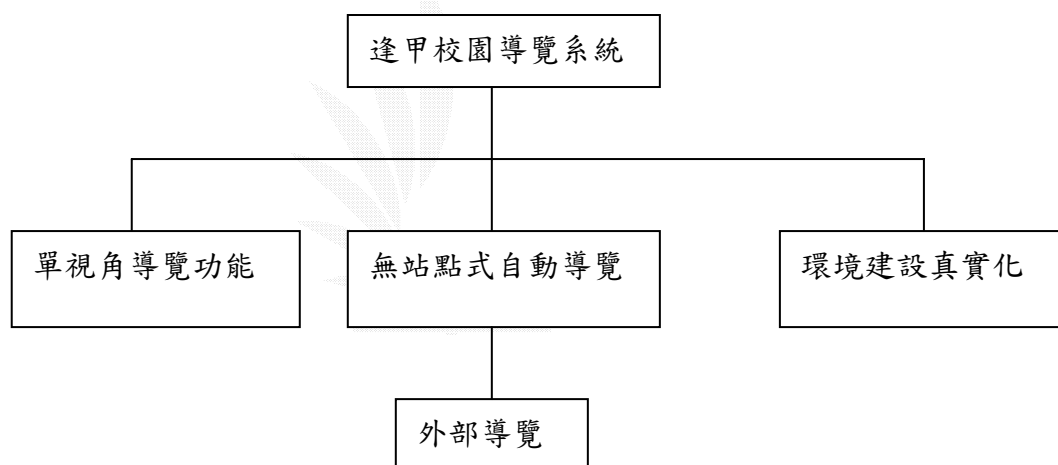


圖 4.1 初擬架構圖

對於這單純的功能來看，實在無法登大雅之堂。後來，基於技術層面的逐漸理解與掌握，再加上老師的鼓勵和提供了寶貴建議，我們便大膽的增加下列功能：

1. 多視角導覽功能---解決了單一視角的非彈性攝影單向拍攝，使用者可根據需要切換視線角度。

2. 站點式的自動導覽---可讓導覽更具有彈性，方便擴充成可半途切換目的地之導覽功能。
3. 半途更改目的地---這項功能可以讓使用者在使用自動導覽的功能時採用的子功能，使用者可以很方便地隨時隨地更改目的地，而無須每次都從頭開始導覽。
4. 按鈕選項---改善使用者介面,提高易用性。
5. 熱鍵功能---提高方便性。
6. 疑難排解---讓初使用者解決使用時容易產生的困惑。
7. 內部導覽---可在某個建築物內導覽。

其功能架構圖如下：

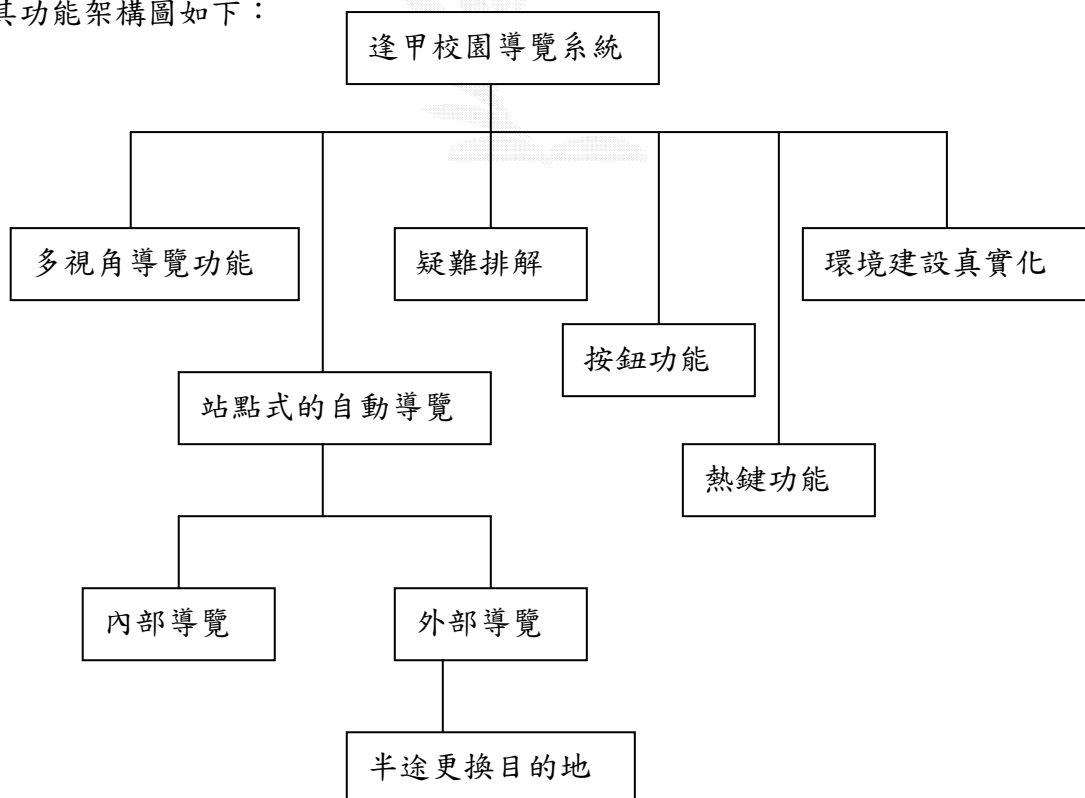


圖 4.2 完整架構圖

4.2 作業流程

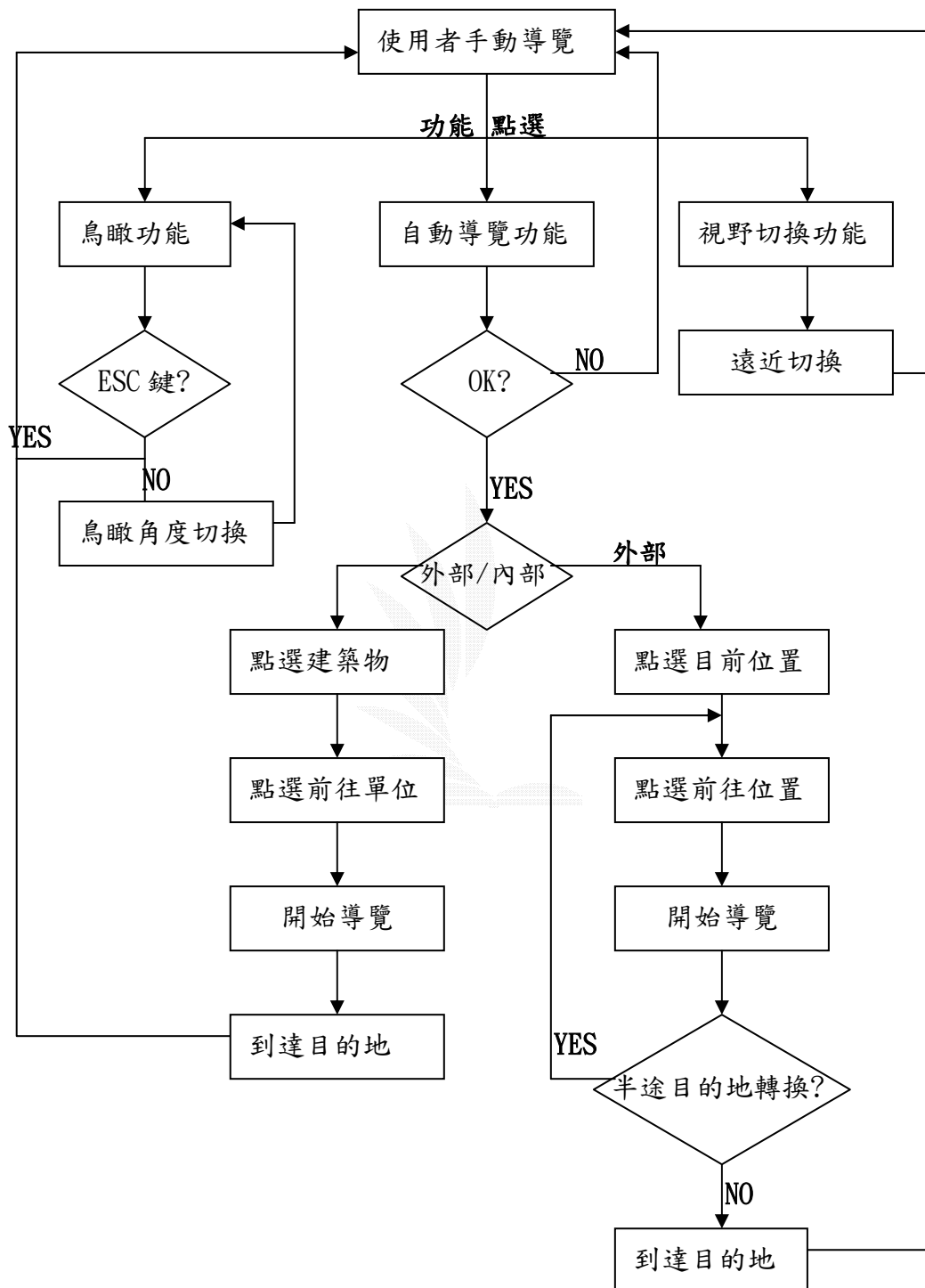


圖 4.3 作業流程圖

第五章 開發工具

這次的專題主要以 3D GAME STUDIO 軟體工具來實作。它是一套 3D 遊戲製作引擎，內含有簡易的 3D MODEL MAKER 與 C-SCRIPT MAKER. 前者就是簡化了的 3D STUDIO MAX。

5.1 3D STUDIO MAX

3D STUDIO MAX 是一套最普遍的 3D 動畫製作軟體，其應用範圍非常廣泛，舉凡建築透視動畫、電影電視廣告設計以及遊戲製作，都可以找到它的蹤跡。多樣的外掛程式，是 3D STUDIO MAX 的特色之一。

3D STUDIO MAX 是很棒的模型製作軟體，容易上手使用，最吸引人的是它與軟體 3D GAME STUDIO 相容。之前學長主要使用它來實作建築物的模型，結果發現當放進 3D GAME STUDIO 執行時，有時會造成某些畫面模糊不清。於是之後學長就告訴我們直接用 3D GAME STUDIO 來實作建築模型，經過一番考量後，我們採納了學長的建議。

5.2 3D GAME STUDIO

前面談過，它含有簡化的 3D STUDIO MAX 與 C-SCRIPT MAKER，然而這套系統引擎是如何運作的呢？請看圖 5.1：

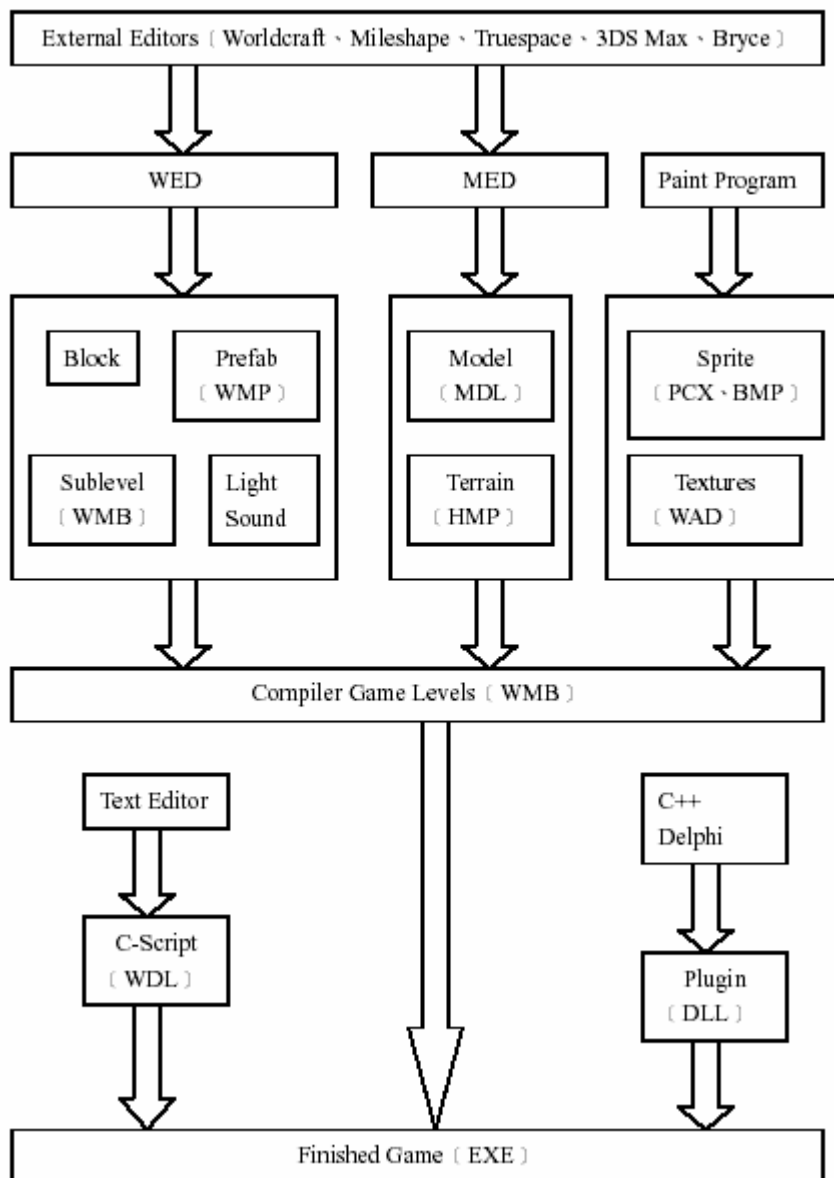


圖 5.1 3D GAME STUDIO 引擎架構圖

External Editors 用來編譯 WED 和 MED 的額外編譯器，包括了 Worldcrat、Milescape、Truespace、3DS Max 和 Bryce 等著名軟

體。如圖 5.2，WED 代表的是整個世界環境建設，一個世界的建構可能包括有很多大大小小的區塊、許多的燈光音效、或者包含著另一個子世界〔子環境〕。

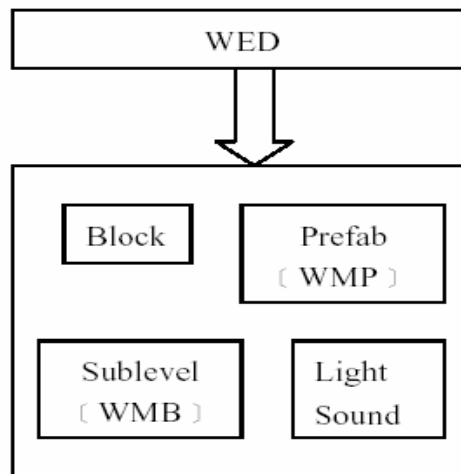


圖 5 2 WED 子系統

如圖5.3，MED代表的是模型建造，包括人物模型、環境建築物模型、材質貼圖。

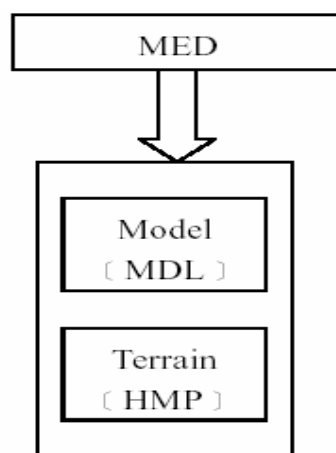


圖 5 3 MED 子系統

如圖5.4，Paint Program 指的就是繪圖軟體，利用繪圖軟體去做出不同的材質提供貼圖之用，當然3D Game Studio裡面也有一些預設好的貼圖材質可以直接呼叫引用。

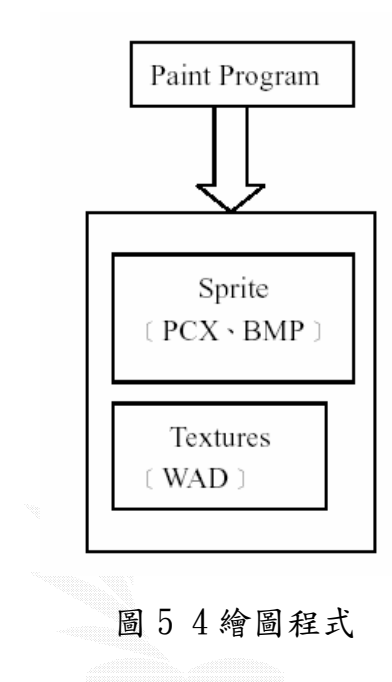


圖 5 4 繪圖程式

接著下來我們只要在編譯時將整個世界存成WMB，WMB 就是一個世界的大小。到這邊大家應該有發現一件事，那就是WED 和WMB 的世界，在WED 的定義中也包含了另一塊WMB，這也就是說我們可以將一個已經做好完整的世界 [WMB] 再包裝到另一個世界裡頭。

在完成WMB的製作後，為了要做出在世界裡面的所有動作，C-Script 和其他程式編輯器在這時後被運用上。如圖5.5，C-Script 可以用來製作所有物件的動作語法。這邊的動作可包含有人物的行走、陽光的照射角度、樹葉隨風飄逸、雲朵的飄動、門的開關、電梯的垂直移動等等。

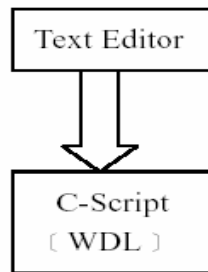


圖 5 5 動作語法

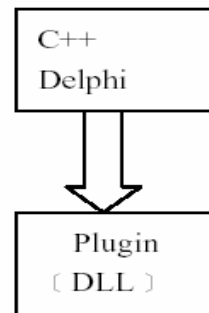


圖 5 6 動態函式庫

最後，把三者〔WDL、WMB、DLL〕編譯成一個可執行檔，到這裡就已經完工了。

第六章 系統實作解說

6.1 模型實作解說

GameStudio 是一個可以製作 3D/VR 遊戲的軟體，透過和我們 Web3D 軟體相似的建模和貼圖、音效、互動等設定，可以方便地建立簡單的遊戲場景，反過來說明，遊戲方面的應用也是我們 Web3D 技術的應用之一。GameStudio 軟體裝設以後，產生兩個可執行程式：

6.1.1 Model Editor 模型編輯器

可以為所製作的 VR 遊戲製作三維模型並進行紋理和動畫的設定。

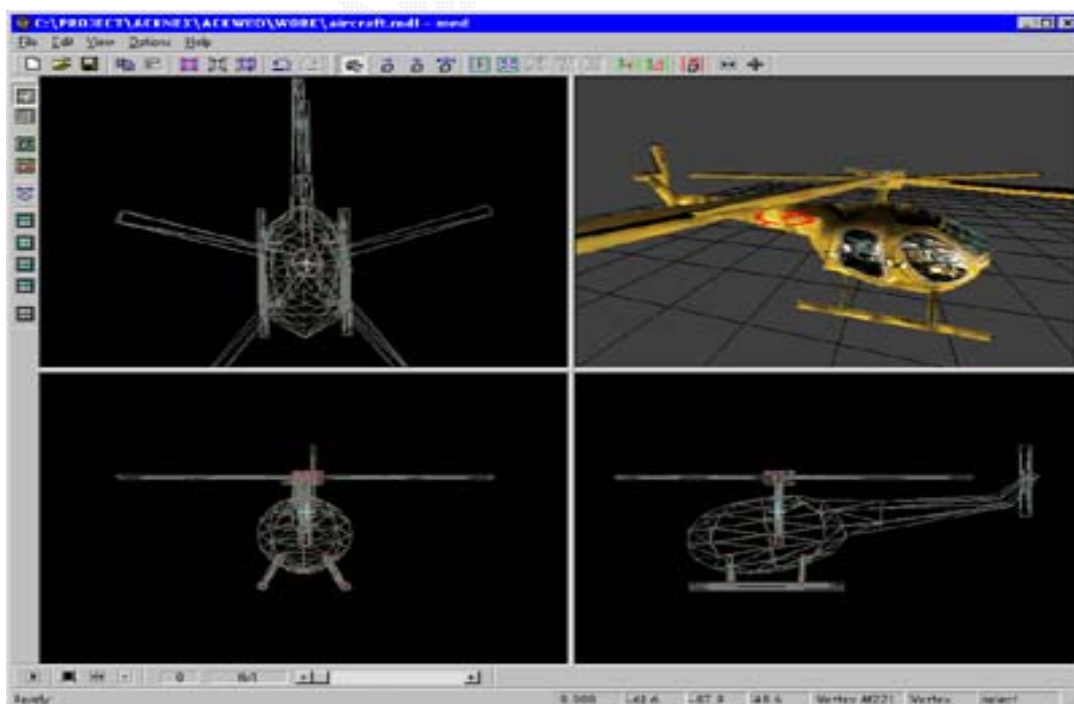


圖 6.1 Model Editor 支援曲面動畫和頂點動畫

6.1.2 Level Editor 層次編輯器

在 Level Editor 層次編輯器中，可以設定場景的燈光、霧效；可以導入 Model Editor 模型編輯器編輯的模型對象，然後對模型進行移動、旋轉、縮放；可以導入音效；可以對其進行紋理貼圖；可以進行視角的編輯；可以加入或者導入互動腳本。

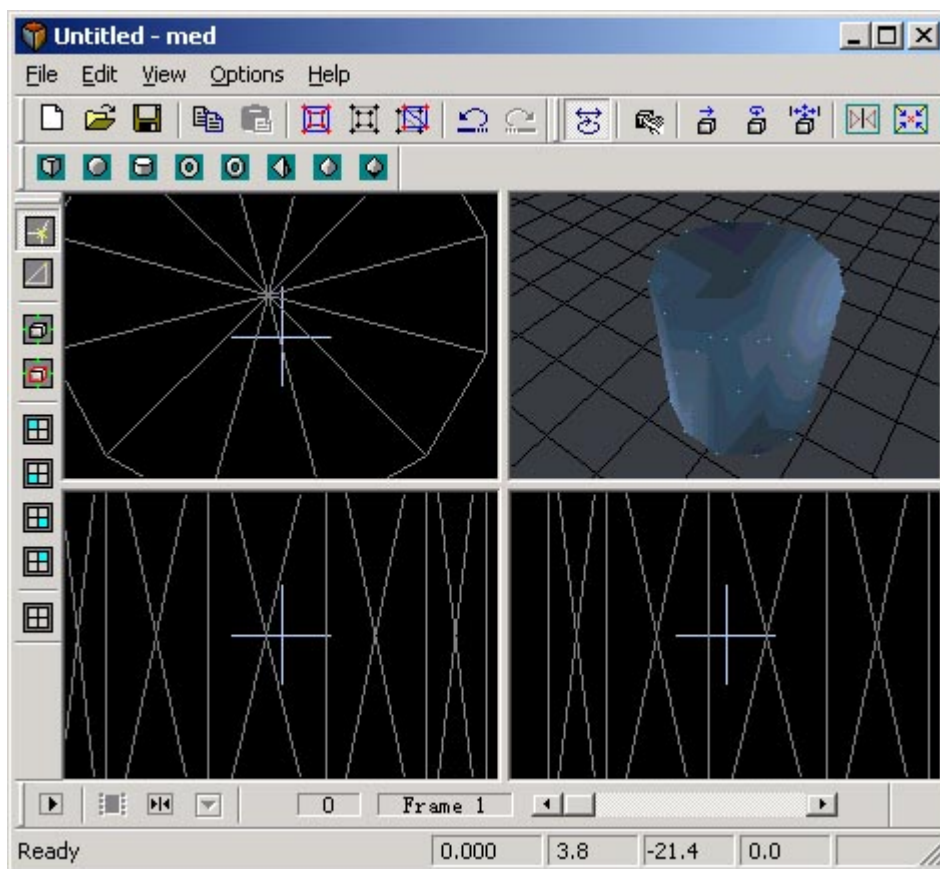


圖 6.2 紋理貼圖

6.1.3 建築物製作

在本專題上我們是用 3D 呈現出接近真實的景像，故我們會在要繪出的建築物的周邊拍攝，已直接與相片比對的方法來令所要繪出的建築物細緻度提高。

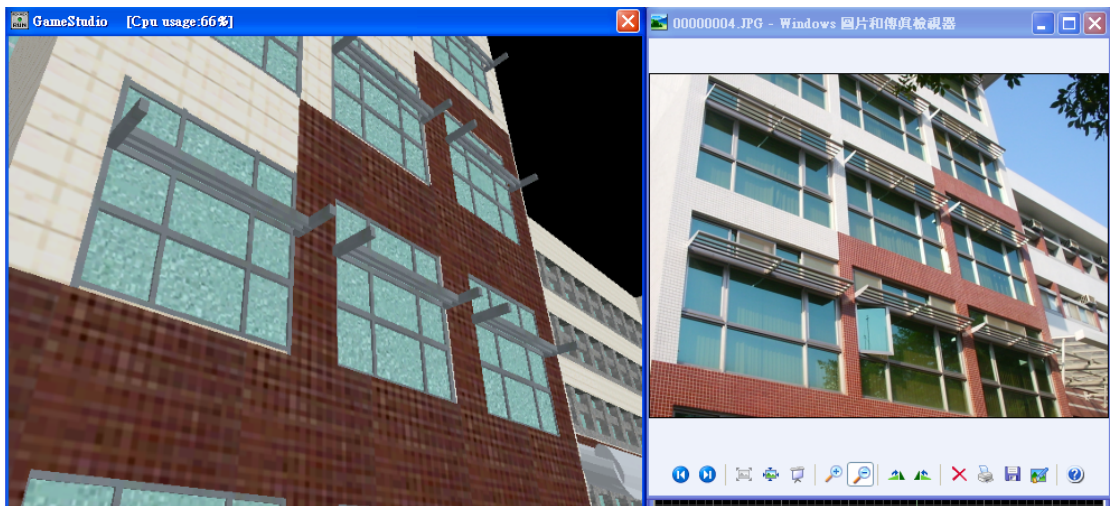


圖 6.3 真實圖與 3D 圖的對應

建構模型如同真實建築該建築物般，先由底部開始。我們先用忠勤樓為例：

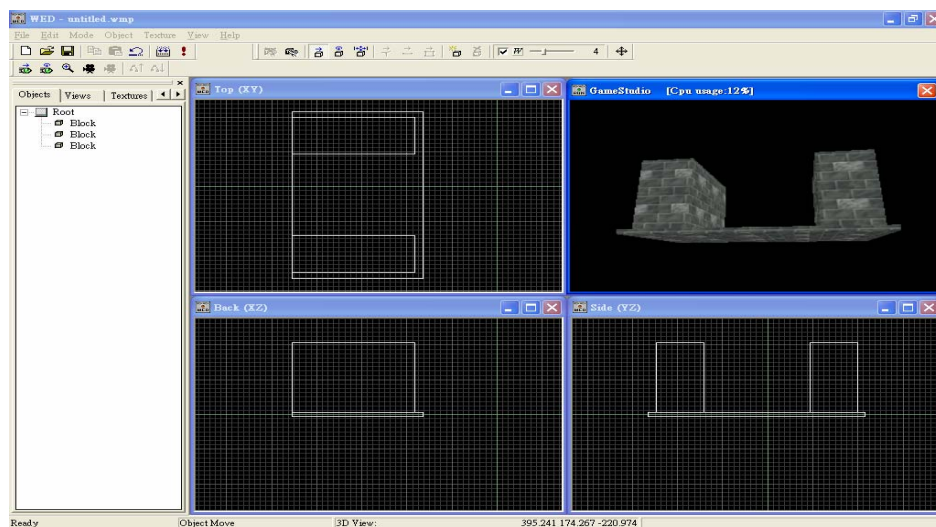


圖 6.4 未加入材質的構造圖

圖 6.4 是已插入主要的兩棟大型樓層但卻還沒加入材質，沒有指定的基本材質如右上角小視窗般的樣子。

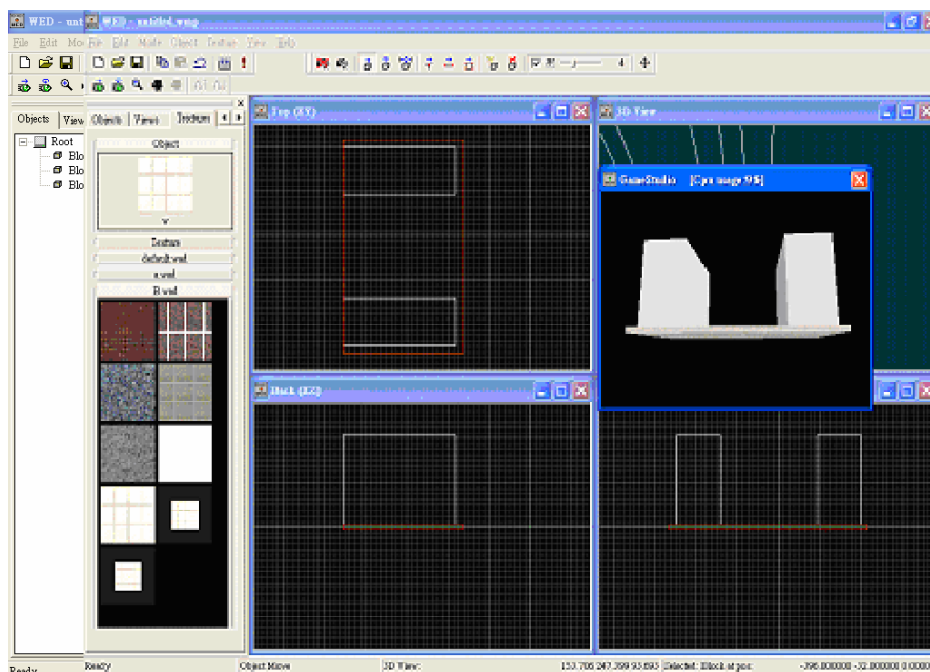


圖 6.5 加上材質後的預覽(右上角)

而加入左旁列表的材質則令原本平淡的圖型更加和原本的建築相似。但內附的材質有限，且大部份不是我們所希望的樣式，故要製作材質。在為了更似目標建築物，我們以數位相機直接把該建築物的外牆拍攝下來，再加工製作成適合使用的 96X96 或 64X64 的 BMP 圖片檔。

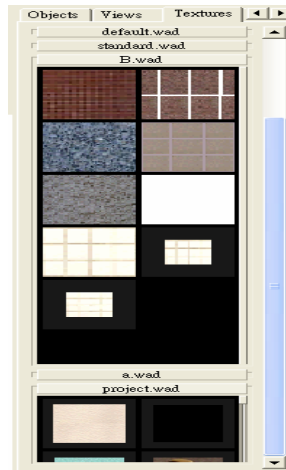


圖 6.6 材質選項

製作一個 3D 模型就像用積木去拼裝一樣。

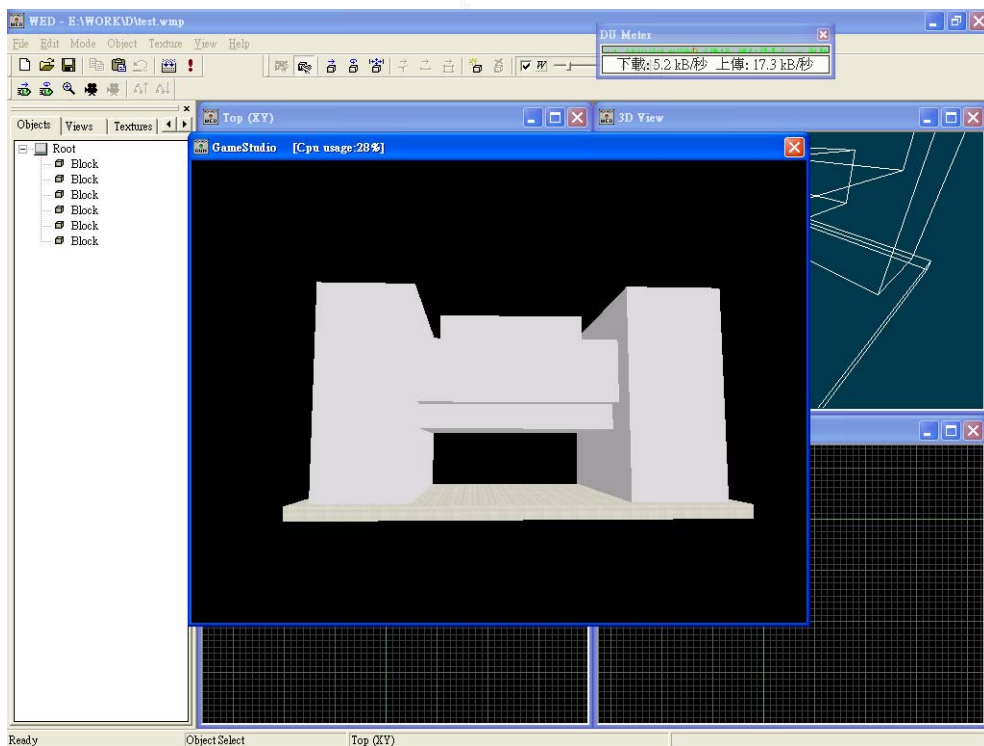


圖 6.7 忠勤樓架構

必要時還需要切割出相對的位置，如門、打通等。門窗的製作則用玻璃材質埋在已切割出來的位置上，一塊一塊地拼上，必要時

可以直接作窗邊貼在外牆上已達效果。

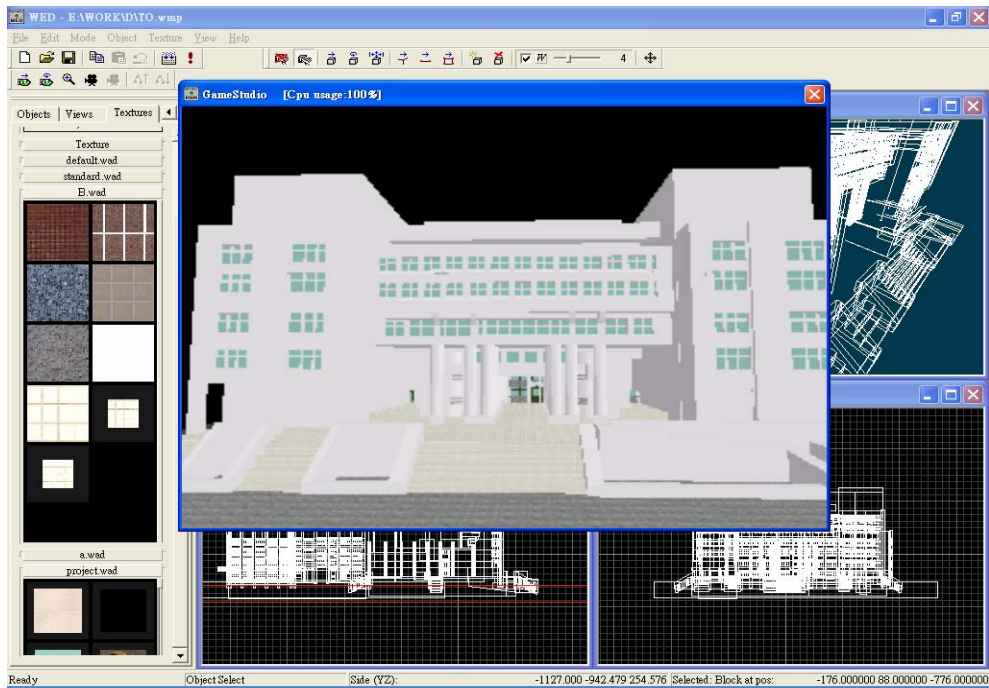


圖 6.8 忠勤樓 3D 外觀

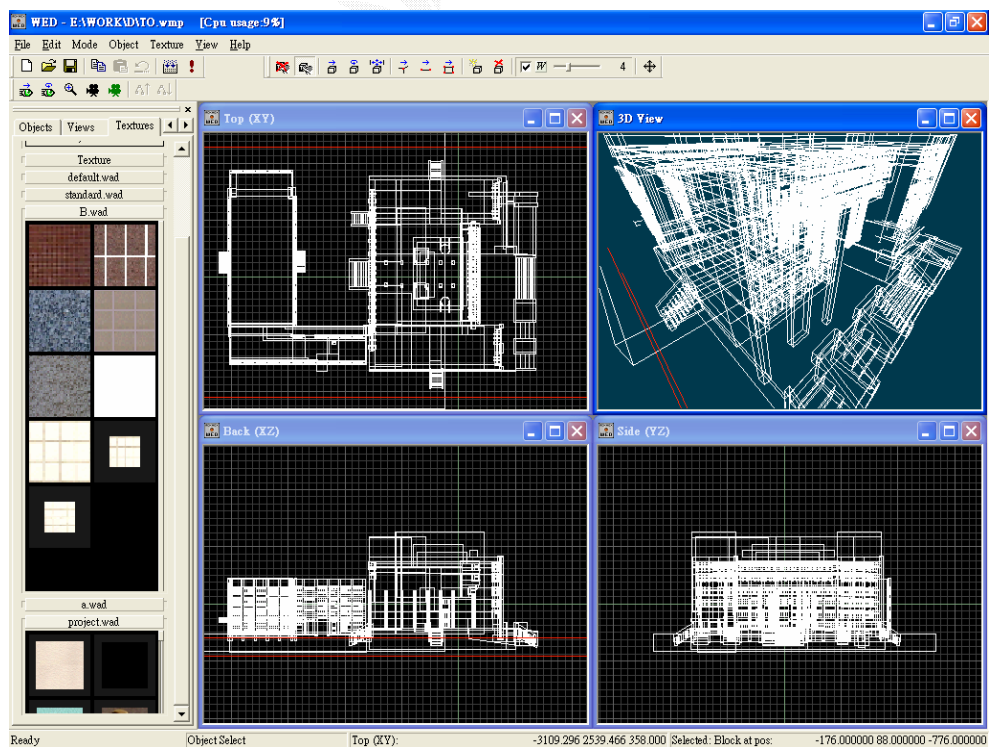


圖 6.9 忠勤樓教學大樓和語文大樓的平面圖

6.2 路徑導覽實作解說

這個部份起初我們參考了學長的程式，但是他的路徑選擇方法相當模糊，最後我們決定自己重新撰寫。

6.2.1 校園分區編碼

在構思的時候，我們想到說正在實用時使用者不可能拿著導覽系統到處跑，所以我們決定不打算實作半途轉自動導覽的部份。我們的方法是—如果選擇自動導覽的話，會先出現選單，點選目前使用者的位置，然後再點選要前往的位置。整個導覽的過程就是從使用者位置走到目的地。至於使用者該如何知道其目前位置，只要查看每個資訊站機器旁所給予的名字就可以了。

起初我們想用最簡單的構想—從每棟建築物拉一條路徑到其他的建築物。這個想法會造成地圖上的路徑很凌亂，沒有彈性，沒有擴充的餘地。於是我們就想到學長所說的設立站點，不過我們還用了一個方法讓程式判斷比較好寫，就是先將地圖做個編碼的工作，每棟建築物我們用兩個元素的陣列來表示它。

比如商學大樓跟資電館設定的第一個代號都是 1，第二個代號分別為 1 跟 2，那麼商學館的陣列代號就是 [1, 1]，資電館的陣列代號則為 [1, 2]，兩者同時與資訊站 A 相連。由此，假設使用者想從某一建築物

走到資訊管(使用者點選資訊管後會有一個目的地陣列[1,2])，那麼路
程中經過的資訊站只要判斷其陣列的第一個代號如果是1，就朝資訊站
A 的方向走去，一直走到資訊站 A 後，才利用第二個代號來判斷如果是
1，就去商學大樓，2 的話就去資電館。當然，如果幾棟建築物彼此連
去同一個資訊站，它們的陣列第一個代號必須相同。圖 6.10 是校園地
圖，圖 6.11 是編碼後的校園地圖(不是每個地點都實作)。



圖 6.10 未編碼的校園圖

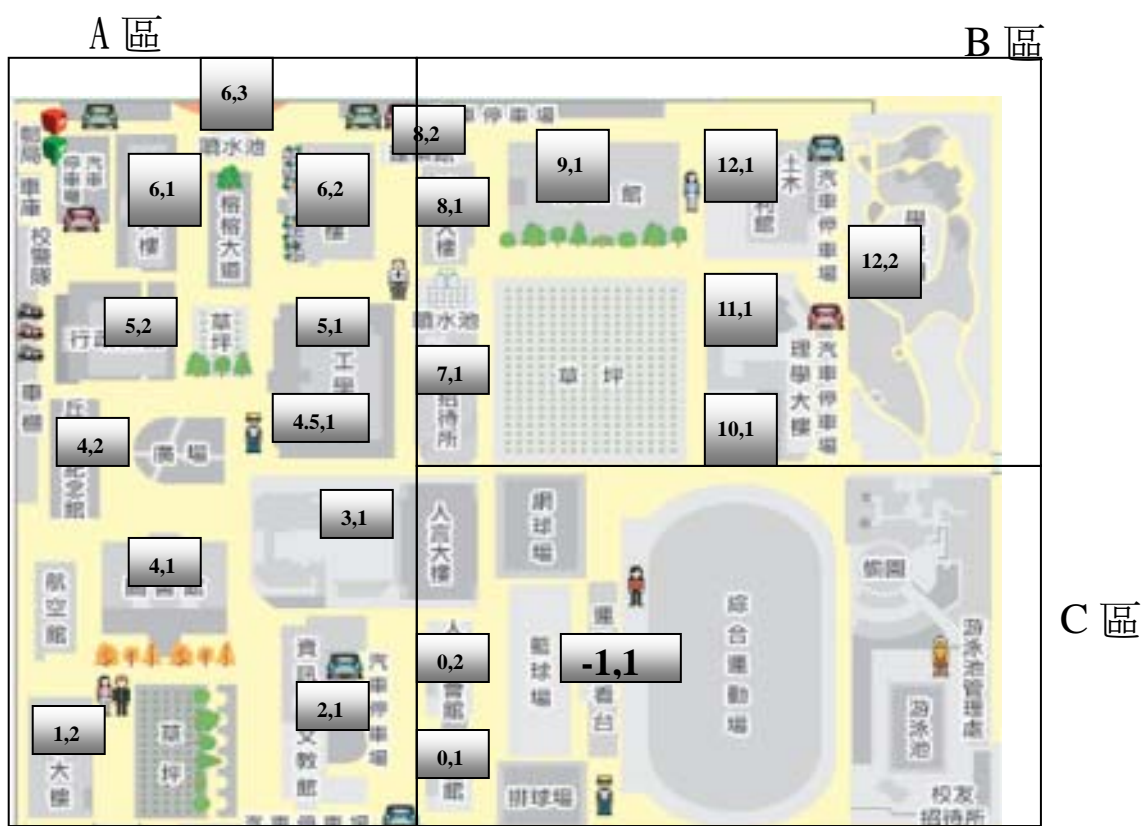


圖 6.11 已編碼的校園圖

由上圖可知，A 區的第一個陣列代號的範圍是 1、2、3、4、5、6，而 B 區的第一個陣列代號的範圍是 7、8、9、10、11、12，然後 C 區則是 -1 及 0。編碼的宣告設定程式如下：

```
//-----建築物代號-----
var tumushuili[2] = 12,1 ; //土木水利館
var xuesiyuan[2] =12,2 ; //學思園
//.....
var lixuedalou[2] = 11,1 ; //理學大樓
//.....
var jiaotangxiao[2] = 10,1 ; //教堂
//.....
var yuleguan[2] = 9,1 ; //育樂館
//.....
```

```
var yuwendalou[2] = 8,1 ; //語文大樓
var jiaoxuedalou[2] = 8,2 ; //教學大樓
//.....
var diyizhaodai[2] = 7,1 ; //第一招待所
//.....
var xingzhengguan1[2] = 6,1 ; //行政館
var zhongqinlou[2] = 6,2 ; //忠勤樓
var xiaomenkou[2] = 6,3 ; //正門口
//.....
var gongxueyuan[2] = 5,1 ; //工學院
var xingzhengguan2[2] = 5,2 ; //行政二館
//.....
var lianhefuwu[2] = 4.5 , 1 ; //聯合服務中心
//.....
var tushuguan[2] = 4,1 ; //圖書館
var qiufengjياتang[2] = 4,2 ; //丘逢甲紀念堂
//.....
var renyandalou[2] = 3,1 ; //人言大樓
//.....
var zidianguan[2] = 2,1 ; //資電館
//.....
var shangxuedalou[2] = 1,2 ; //商學大樓
//.....
var diantongguan[2] = 0,1 ; //電通館
var renwenshehui[2] = 0,2 ; //人文社會館
//.....
var yundongchang[2] = -1,1 ; //綜合運動場
```

6.2.2 設立站點

在實作中站點又分終點站和非終點站，終點站其實就是代表某建築物的位置，非終點站則是尋找這些終點站的站點。看了圖 6.12 就會很清楚了。

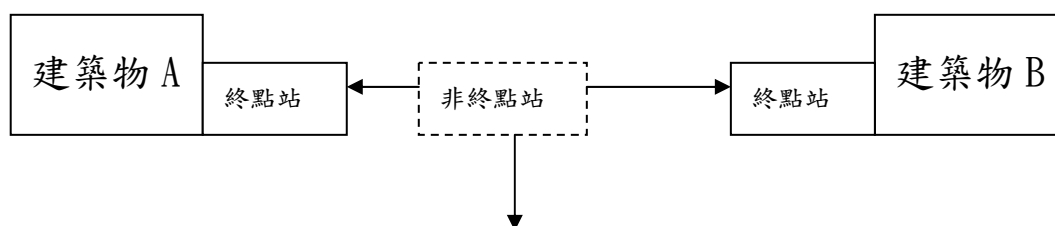


圖 6.122 終點站和非終點站

6.2.3 終點站

終點站的程式比較簡單，每個終點站的程式都一樣，就只是呼叫處理到達目的地後要做的事情的函數。比如從自動轉手動導覽，發出到達的聲音等。這個函數的程式如下：

```
function arrive_building()
{
    whamm_handle = snd_play(arrive_sound, 100, 0); //發出聲音
    ME=player;
    patrolpath=1; //關閉自動導覽變數
    playwalk=0; //開啟手動導覽變數
    player_walk(); //執行手動導覽
    arrive_reset();
    arrive_panel.visible=on;
}
```

6.2.4 非終點站

非終點站的內容就比終點站的複雜了。它必須：一、知道目的地編號是本身這個站點的哪個方向。二、知道目的地是否就是在下一個站點(終點站)。三、必須紀錄下一個站點的編號(非終點站的站點有設編號)，此編號用來處理使用者半途更改目的地所要做的事。在此只列出

其中一個非終點站的程式內容：

```
function choose_which_path_for_event12()      // L 資訊站的內容
{
    if(go_position[0]==2) { //如果目的地就在下一個終點站
        betw_station[1] = 0 ;
        str_cpy(start_to_next, "path_stop12"); //與 12 號區連結的站
        點

        str_cpy(what_path, "path_zidian"); // 走向資電館
        str_cpy(arrive_path, "from_zidian");
        back_position[0]=2;
        back_position[1]=1;
    }
    else {
        //去下一個非終點站
        if(go_position[0]>=9 || go_position[0]<=0 ||
            go_position[0]==3) {
            betw_station[1] = 15;
            str_cpy(what_path, "stopL_to_stop0");
        }
        else {
            betw_station[1] = 11;
            str_cpy(what_path, "stopL_to_stopK");
        }
    }
}
```

這個函數會等到人物觸發事件時才被呼叫執行，其動作是：如果目的地就與它連結的下一個終點站，那麼就直接走去然後結束導覽；如果不是，就判斷 go_position[0]在此非終點站的哪個方向，然後朝那個方向的非終點站走去，以此類推繼續導覽。

6.3 導覽流程的程式控制

6.3.1 從原地到目的地

當使用者一開始點選目前位置後，back_position 這個陣列就會被 ASSIGN 該位置的編碼值，然後根據編碼值將人物跳到地圖上屬於該建築物編號的地點(某個 x, y, z 軸). 程式片段如下：

```
you = ptr_for_name ("warlock_md1_3757");  
your.X =796;  
your.Y = 3088;  
your.Z = -260;
```

之後使用者點選了目的地，go_position 這個陣列就會如同 back_position 一樣被 ASSIGN 該目的地的編碼值，作為站點判斷之用。開始導覽時，人物會跟隨在地圖上拉好的路徑來走，而且會碰到地圖上的特殊物件，這些物件就是資訊站，碰觸物件就會掃描並觸發事件，其掃描與碰觸程式片段如下：

```
//掃描事件  
result = content(my.x);  
if (result == CONTENT_PASSABLE) {  
    temp.pan = 100;  
    temp.tilt = 100;  
    temp.z = 150;  
    result=scan_entity (my.x, temp);  
}  
  
//其中一個觸發事件  
action touch_event_15 事件名稱  
{
```

```
my.ENABLE_scan = ON;  
my.ENABLE_detect = ON; // sensible for scans  
my.EVENT = what_event_15; //呼叫函數名稱  
}
```

觸發事件會呼叫所要做的事情，由於這些事件都是所謂函數，這些函數會被呼叫來實作如前所述的事件處理(請看 6.2.2 設立站點)。

每次觸發事件都要尋找正確路徑，而掃描然後進入該路徑的程式片段如下：

```
temp.pan = 360;  
temp.tilt = 180;  
temp.z = 1000;  
result=scan_path(my.x, temp); //掃描周圍路徑  
enter_path(); //找到屬於該名稱的路徑然後沿著它來走
```

6.3.2 半途更改目的地

使用者可以在自動導覽的過程尚未結束時臨時更改前往的目的地，使用者不必說因為選錯目的地而得從新點選，這樣不但可以提高使用效益，還可以增進親和力。

起初我們的構思是把演算法設定成：每次更換目的地後，必須等待人物角色經過了下一個站點才做偵測，但是這樣一來就有可能會走冤枉路了。於是改變初衷，經過一番討論了以後，終於想到了一個可以避免走冤枉路辦法。

我們想到的演算法是:在每個站點設立一個編號,用 BETW_STATION 陣列來表示站點與站點之間的路徑。比如說站點 2 和站點 3 之間的路徑就是用 [2, 3] 來表示,然後每個站點放置一個‘單節點’路徑(就是一個點),每個‘單節點’路徑都設不同的名稱。其控制流程是:當使用者走到路徑 [2, 3] 的時候改變了目的地後,程式便會判斷其所更改的目的地是比較靠近站點 2 還是站點 3。如果是站點 2,就透過 `str_cpy(what_path, "path_stop2")` 得到其‘單節點’路徑名稱,並掃描此路徑名稱位於何處,再前往之。由於該‘單節點’路徑跟站點是放在一起的,當人物角色前往後會同時觸發該站點的觸發事件來繼續尋找前往下一個站點的路徑。以下是站點 2 和站點 3 之間改變目的地時會呼叫到的函數:

```
function change_road_betw23()
{
  if(go_position[0]>4) {
    str_cpy(what_path, "path_stop2");
    rescan_path();
  }
  else {
    str_cpy(what_path, "path_stop3");
    rescan_path();
  }
}
```

函式 6.1

6.4 導覽攝影機程式解說

6.4.1 導覽攝影機視角遠近切換

導覽攝影機視角切換是指使用者可以以第一人稱或第三人稱的方式來導覽校園。遠近則是指說使用者在第三人稱時可以調整遠或近的拍攝距離在此，第三人稱的導覽攝影機拍攝方向是固定的，不會像第一人稱般人轉向哪裡就拍哪裡。這裡利用了 `camera_angle_change` 來表示目前切換的拍攝距離，判斷了目前拍攝距離以後就得判斷目前是拍那個方向(當不是第一人稱時)來做導覽攝影機的定位(此變數是 `camera_stable`)。另一個主要的變數是長度為 3 的 `chase_camera_dist[]` 陣列。第一個元素代表導覽攝影機與人物的前後距離，第二個元素代表左右距離，最後一個代表高度距離，調整這三個變數可以改變拍攝遠近距離跟拍攝方向。其控制的程式如下：

```
function change_cam_angle()          //camera 遠近視線更換
{
    whamm_handle = snd_play(whamm, 100, 0);
    person_3rd=3;

    camera_angle_change=(camera_angle_change+1)%2 ;

    if(camera_angle_change==0) {
        chase_camera_dist[2]=3000;
        ctemp=2000;
    }
}
```



```
        if(camera_angle_change==1) {
            person_3rd=0;
        }

        if(camera_stable==0) {           // check camera 方向
            chase_camera_dist[0]=-1*ctemp;
            chase_camera_dist[1]=0;
        }
        if(camera_stable==1) {           // check camera 方向
            chase_camera_dist[1]=1*ctemp;
            chase_camera_dist[0]=0;
        }
        if(camera_stable==2) {           // check camera 方向
            chase_camera_dist[1]=0;
            chase_camera_dist[0]=1*ctemp;
        }
        if(camera_stable==3) {           // check camera 方向
            chase_camera_dist[1]=-1*ctemp;
            chase_camera_dist[0]=0;
        }
    }
}
```

函式 6.2

6.4.2 導覽攝影機定位方向切換

導覽攝影機定位方向切換是指使用者可以切換拍攝方向，然而我們為了讓使用者不失去方向，於是在銀幕的上下左右會指明正門口的位置。這裡我們利用 camera_stable 這個變數來代表導覽攝影機拍攝的方向，這個方向是一直固定的，除了使用者自行切換。當 person_3RD=0 代表第一人稱，這時就沒有定為方向的拍攝，其控制的程式如下：

```
function camera_stable_switch()      //camera 拍攝方向函數
{
    if(person_3rd!=0) {      //如果不是第一人稱,則作定位切換
        whamm_handle = snd_play(whamm, 100, 0);
        camera_stable=(camera_stable+1)%4 ;

        if(camera_stable==0) {      //往下走是正門口
            chase_camera_dist[0]=-1*ctemp;
            chase_camera_dist[1]=0;
            //改變指明正門口位置的 PANEL
            show_door_direction1.pos_x=330;
            show_door_direction1.pos_y=570;
            show_door_direction2.visible=off;
            show_door_direction1.visible=on;
        }
        if(camera_stable==1) {      //往右走是正門口
            chase_camera_dist[1]=1*ctemp;
            chase_camera_dist[0]=0;
            //改變指明正門口位置的 PANEL
            show_door_direction2.pos_x=770;
            show_door_direction2.pos_y=280;
            show_door_direction1.visible=off;
            show_door_direction2.visible=on;
        }
        if(camera_stable==2) {      //往上走是正門口
            chase_camera_dist[1]=0;
            chase_camera_dist[0]=1*ctemp;
            //改變指明正門口位置的 PANEL
            show_door_direction1.pos_x=330;
            show_door_direction1.pos_y=70;
            show_door_direction2.visible=off;
            show_door_direction1.visible=on;
        }
        if(camera_stable==3) {      //往左走是正門口
            chase_camera_dist[1]=-1*ctemp;
            chase_camera_dist[0]=0;
            //改變指明正門口位置的 PANEL
            show_door_direction2.pos_x=10;
```

```
        show_door_direction2.pos_y=280;  
        show_door_direction1.visible=off;  
        show_door_direction2.visible=on;  
    }  
}  
}
```

函式 6.3

6.5 按鈕程式解說

要以圖形作為按鍵, 首先必須定義一個 Panel, 這個 Panel 可以作為圖形的貼版, 顯示在畫面上, 而貼在此 Panel 的圖形便可用來當做按鈕. Panel 內有一些屬性可以利用程式來設定, 達到各種貼版顯示效果, 程式如下:

```
Panel go_panel  
{  
    flags = REFRESH, transparent;  
    pos_x = 300; pos_y = 150;  
    mouse_map = maus;  
    bmap = go;  
    BUTTON=0, 0, go, go, go, check_and_go, null, null;  
}
```

演算法 6. 1

底下我們來看看 Panel 的幾個主要屬性:

- Flags 屬性

flags 有四個參數, 分別是 REFRESH, VISIBLE, TRANSPARENT 和 OVERLAY.

(1)REFRESH: 如果沒有設定這項參數, 則此 PANEL 只有在位置改

變時才會重畫(redraw)。

(2)VISIBLE：這項設定使得 PANEL 可以顯示在畫面上, 如果

這裡沒有預設, 則之後可通過

PANEL_NAME. VISIBLE=ON 來將其顯示。

(3)TRANSPARENT：這項設定使得按鈕呈現半透明化顯示。

(4)OVERLAY：這項設定使得按鈕顏色為 0(純黑色)的部份不顯

示。

- Pos_x, Pos_y 屬性

用來將 PANEL 顯示在畫面上 X 軸與 Y 軸的位置。

- mouse_map 屬性

設定當滑鼠移到此 PANEL 時變換其滑鼠圖示。

- Bmap 屬性

代表此 PANEL 的圖示。

- BUTTON 屬性

將此 PANEL 作為按鈕貼版使用. 其參數如下：

x, y- 按鈕圖示顯示在此 PANEL 內的 X, Y 軸位置。

Bmapon - 當按鈕被按下時顯示的圖示。

Bmapoff - 當按鈕沒被碰到也沒被按下時顯示的圖示。

bmapover - 當滑鼠碰到按鈕時顯示的圖示。

`functionOn` – 當按鈕被按下時呼叫的函數名稱。

`functionOff` – 當按鈕沒被碰到也沒被按下時呼叫的函數名稱。

`functionOver` – 當滑鼠碰到按鈕時呼叫的函數名稱。

6.6 聲音程式解說

為了讓系統使用起來更有親和力，我們便附加了音效功能. 比如剛開始進入系統時會發出 ‘歡迎使用導覽系統’ 的聲音，以及按鈕點選也會發出聲音等等。然而在 3D GAME STUDIO 裡已經有內建程式做了許多聲音處理的功能，所以實作起來非常簡單如下：

```
sound whamm = <click.wav>;  
var_nsave whamm_handle;  
whamm_handle = snd_play(whamm, 100, 0);
```

函式 6. 2

第一行是宣告一個 `sound` 物件來代表聲音檔，第二行是用來設定一個控制變數以避免當載入時被不合法的 `VALUE` 所覆蓋。第三行則是播放聲音，第一個參數是聲音物件名稱，第二則是聲音大小(0-100)，最後是 `STEREO` 平衡值，0 代表左右聲量平衡。

第七章 使用者介面說明

本專題運用3D Game Studio 開發完成，現在開始逐一介紹在系統中的操作介面。

圖7.1是使用系統時的畫面，其上方有五個按鈕，從左到右分別是自動導覽、視野切換、攝影定向切換、疑難排解以及鳥瞰功能。下方有一個標示寫著‘正門方向’代表鍵盤往下走是學校正門口的方向：



圖 7.1 系統畫面

7.1 自動導覽

1. 當使用者按下自動導覽, 即出現圖 7.2:



圖 7.2 是否確定自動導覽

1. 當使用者按下 YES, 即出現圖 7.3:



圖 7.3 選擇內部或外部導覽

3. 當使用者按下外部導覽, 即出現圖 7.4:

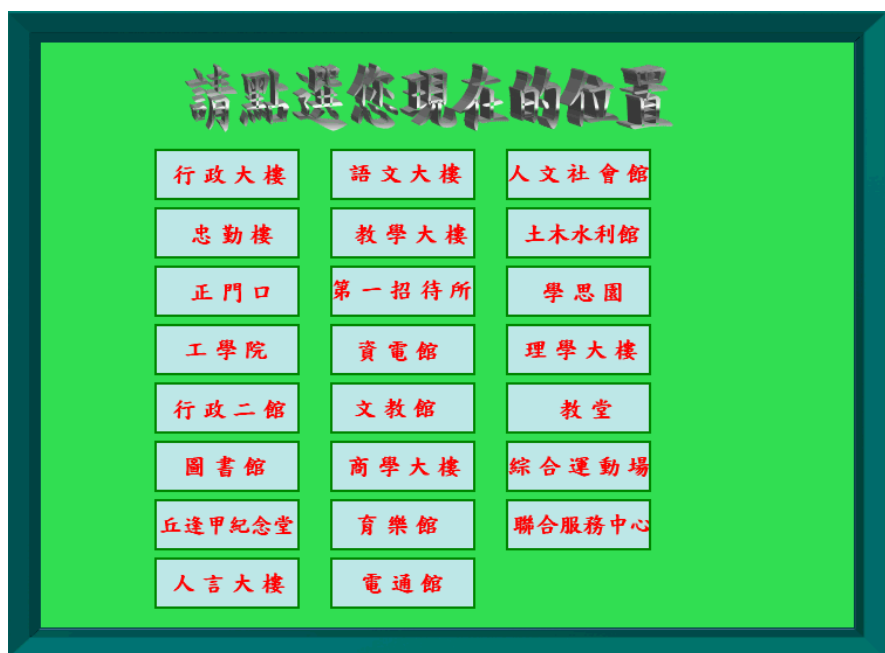


圖 7.4 點選目前位置

4. 當使用者點選建築物後, 即出現圖 7.5:

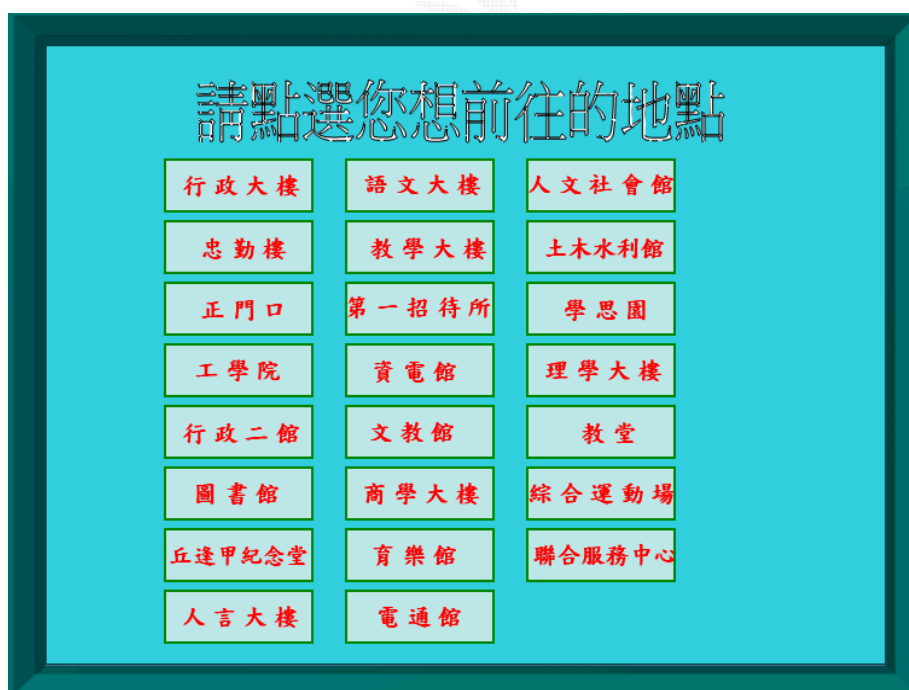


圖 7.5 點選前往位置

5. 當使用者點選建築物後, 即出現圖 7.6 開始導覽畫面:



圖 7.6 開始導覽畫面

圖 7.6 左上角有兩個按鈕, 左邊的是停止自動導覽, 右邊的是暫停以更改原先點選的目的地. 當使用者點選建築物後, 即出現上圖 7.5 點選欲更換之目的地. 到達目的地如圖 7.7:



圖 7.7 到達目的地顯示畫面

7.2 視野切換

當使用者點選視野切換便會由如前圖的第一人稱轉為如圖 7.8 的第三人稱：

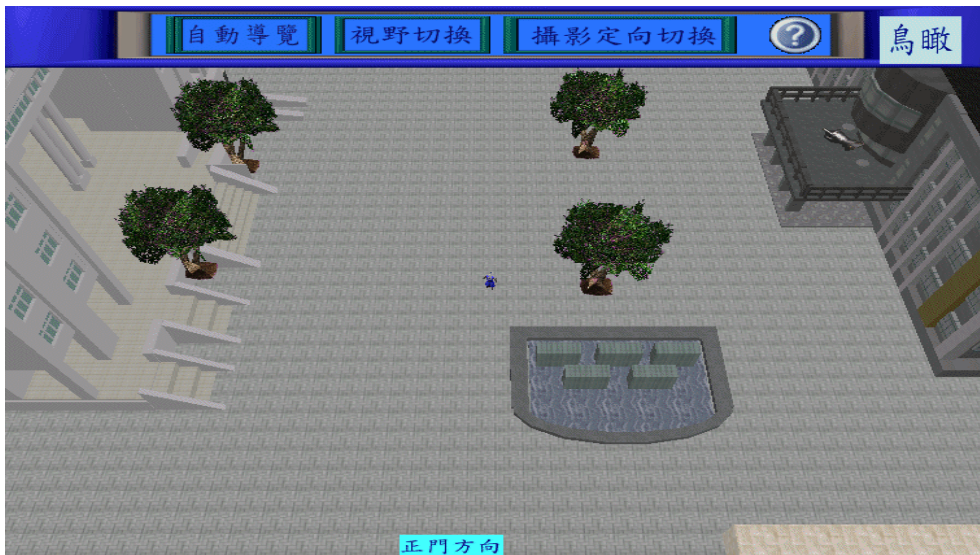


圖 7.8 視野切換畫面

7.3 鳥瞰校園

1. 當使用者點選鳥瞰按鈕時, 便出現圖 7.9:



圖 7.9 鳥瞰圖(1)

2. 此時，螢幕上方所有功能鍵除了鳥瞰外，其餘將不能點選，而當使用者在次點選鳥瞰功能則會改換角度拍攝，如圖 7.10:



圖 7.10 鳥瞰圖(2)

之後要等到使用者按下 ESC 鍵, 才能恢復先前畫面.

第八章 效能評估

8.1 系統評估

跟學長之前發展的系統做個比較下，我們已經對其缺點做了改良，整體看起來不論是在視覺上或者操作上都好了些。但是，由於我們是延續學長的系統，可以說是其‘第二代’，因此我們覺得還是不很滿意。

令我們不很滿意的原因主要是此 3D GAME STUDIO 軟體具有許多功能，而我們所使用到的只有基本的而已。而且，我們這個系統所用到此軟體的功能比起學長的來說並沒有多出多少。比如說我們用了分散編譯和語音功能，此乃其一。其二，本專題系統容量相當可觀，大約 300MB。再加上執行速度所需要的硬體支援也相當講究 (Pentium 4 1.8G)，造成其不方便性。幸好這還沒有結束，因為接下來有三位學弟想再次延續這個系統，希望他們能做的更好。

8.2 功能評估

在本專題中，我們的建築物是做的相當實體化，但是校園環境卻不盡人意。比如噴水池、各種花草樹木以及道路的材質等。這些我們都做得相當草率，主要是這些模型在網路上相當有限，而且我們三個人連一點美術天份都沒有，更不用說自己畫了。

在自動導覽的功能上，由於之前的分析做得不夠徹底，造成維護和更改都相當不便，主要原因是在技術層面上的不成熟。單單研究這套軟體的功能和程式的編寫就花去整整一個月時間，因此在實作上有時便會處在走一步，算一步的不妥狀態下。

最後就是老師所提出的一些要求我們都沒做好，比如有一點是做出一個小地圖顯示整個校園藍圖在螢幕上，方便使用者了解本身處在甚麼位置之下。這個功能我們有去研究實作方法，不過還是失敗了。

第九章 結 論

9.1 組員心得

湯竣杰

本系統開發可以相當順利的進行主要還是得歸功於學長的幫助，因為學長提供了許多資源給我們，這些資源包括開發軟體、範例以及一些模型。除此之外，在一開始學長還教我們該如何起步，這些雖然是非常簡單的事，不過對於當初還不了解狀況的我們，這些可都是寶啊！

一向來喜歡玩遊戲的我，實做了本專題以後，有一種躍躍欲試想寫一個簡單的遊戲軟體的衝動。因為這套軟體提供的功能實在太強大了，省下了開發者不少時間，再加上其開發使用的程式是很接近 C++ 的 C SCRIPT 語言，所以寫一個簡單的 3D 遊戲應該沒有想像中的難吧！

最後希望下一屆的學弟們可以把本系統做得更好，甚至可以正式在學校派上用場，也算是替學校做點貢獻。

陳思鴻

這個專題讓我們有機會自己嘗試實作出一個類似 3D 遊戲的校

園導覽系統，真是個難得的機會。這個專題的魅力太過大了，給了我們很大的動力想要把它做到盡善盡美，遺憾的是時間有限，光要把那個軟體學得熟練些也要花一般功夫。之前還未對這個軟體很深入了解時，遇到的問題如：整合所有的模型面對記憶體不足的問題，用過多的物件去組成一個建築物等。但是，對該軟體認識的時間越長這些問題都已不是問題了；反而現在更有信心地想完成更堅難的題目。

從這次的專題上，讓我體會到團隊的努力是沒有白費的。每個組員都很盡力地完成各自的部分，可能是彼此間無形的壓力迫使大家都不敢鬆懈。能完成這個專題，有一半要歸功於學長的耐心指導還有老師給我們的機會。

翁國淵

科技發展一日千里，人們不再追求簡單的介面而要求一種較為接近真實的介面，在現今設備和技術上要進行這點已不是大問題。在摩爾定律下每十八個月硬體都會以倍數增長，所以因為這樣令高速電腦普及化，使得個人也能借助高效能的電腦把一個個方塊從平面拼為立體，3D 校園導覽系統也是基於這點開始開發的。

由於科技的進步，導覽系統也要因應跟進，平面的導覽已不太吸引使用者的興趣。

在做這個專題前，我從未接觸過製作 3D 圖型的軟體，相較之前學過的工業繪圖軟體，複雜度相差很多。一開始雖能隨意地把外型架構畫出來，因為對軟體還不熟悉，對細部的調整或建造都覺得蠻困難的。不斷地嘗試畫圖，也漸漸對該軟體有了一定的認識。由於時間緊迫，唯有一邊做一學習。在建構第一座建築物上花了很多心思，為了能使建出來的模型和原來的建築物更加相似，材質的製作也得謹慎處理。因為做出來的材質不會依所建好的模型自行分配，而是併排排列。在製作材質的時候，我是拍下建築物的外觀，再透過軟體加工做成材質，讓所建出來的模型外觀更加逼真。其中一個大問題是因電腦速度和記憶體的關係，使我們在整合時編譯出現問題的情況，因而決定簡化各個建築物以便能順利通過編譯。後來發現了解決的方法，故已修改後的建築已不再修改因此有些建築物看起來會有不真實的感覺。在製作的過程中，我們學習到新的軟體，因為有想修完多媒體學程，其實算蠻有用。使用一個軟體必須先徹底了解其全部的功能，不然是無法完善地發揮出該軟體的威力。熟知你所使用的軟體是提高效率的最佳方法，借此體驗到一個遊戲設計師的辛勞，也知道高效能

的電腦對一個 3D 遊戲的製作是必備的。

因為是初學者，有些奇形怪狀的東西都不太會建構，在很多小土丘而且複雜度高的學思園唯有敬而遠之而不作嘗試。有明確的目標才能努力地奮鬥，謝謝之前學長的努力指導使得我們能順利地完成這個專題題目。



9.2 未來展望

我們所做的導覽系統並不算是很完美，它還有許多缺點。比如：我們的內部導覽功能還很缺乏，而且效果不佳，還有就是建築物外觀上其實可以做的更像一點。至於程式碼方面來說，有些很好的功能我們無法實作出來，像是 MINIMAP 功能，它就像玩 3D 即時戰略遊戲一樣，有個小小的地圖在某個角落，使用者可以根據這個小地圖瞭解整個大地圖的情況。

再來就是這套軟體無法將系統網頁化，因為網頁化就可以放到學校網站上去讓同學去瀏覽之，可以提高使用性。除此之外，缺乏與資料庫相連結是其一大缺點。否則的話，可以將許多資訊存在資料庫裡，這樣對程式碼的實作會帶來很大維護性，使程式行為和資料分開以後就不用在程式裡面一味的產生一大堆的宣告，而且，資料的更新也不用去修改程式碼了。

參考資料

書目資料

1. 黃松浪譯，Roger S. Pressman著，軟體工程第三版，
儒林，Mar. 1993，pp. 495 ~ 516。
2. 潘冠宇譯，Valacich。George。Hoffer著，系統分析與設計概論，
碁峰，Aug. 2002，pp. 8-2 ~ 8-52。
3. 劉登榮譯，Jason Kolb 著，DirectX 發展工具，
松格，May. 1997，pp. 11 ~ 24。

網頁資料

1. 3D Game Studio： <http://www.conitec.net/a4info.htm>
2. 瑋特擬真科技：
<http://www.vimtek.com.tw/eonstudio4/about%20eonstudio4.htm>
3. 愛迪斯科技： <http://www.axis3d.com.tw/index.jsp>
4. 逢甲校園導覽系統連結：
<http://www.fcu.edu.tw/life/campus/map1.htm>

逢甲大學
資訊工程學系
專題報告

逢甲大學3D校園導覽
系統製作

湯 陳 翁
竣 思 國
杰 鴻 淵

92

