逢　甲　大　學

資訊工程學系專題報告

3D 校園導覽系統-

以逢甲大學校園為例

學　　生：范獻巍（四甲）

吳江鎮（四乙）

莊家豪（四丙）

指導教授：謝信芳

中華民國九十二年十二月

目　　錄

圖表目錄

第一章 導論

1.1 背景

在一個廣大的大學校園裡頭充滿了各式各樣的建築物，包含教學大樓，行政大樓…等。剛上大學的新生和來賓一來到學校可能摸不清楚方向，如果這時有一個視覺導覽系統，可以指示來賓和學生想要去的地方，那肯定會帶來不少便利，可以避免走錯路的可能，也節省了許多寶貴的時間。我們就以逢甲大學做例子，來實現這個理想。一個好的導覽系統，會有一些優良的特色，譬如：使用 3D 視覺效果呈現，容易使用的介面，還有清楚的導覽說明。於是我們選擇用開發 3D 遊戲的軟體來實現這個系統。這個導覽系統，是完全 3D 視覺化的，所有建築或人物都是以立體 3D 模型來建構，並且仿照逢甲大學實景作為 3D 世界讓使用者可以有身歷其境的感受，有別於 2D 平面地圖的表達狹隘性，確實是一個虛擬世界的導覽系統。除此之外，我們還會提供一些功能來方便導覽使用者，譬如：會提供路徑選擇的功能，當使用者決定目標時，系統會指示幾種可以走的路徑讓使用者選擇，或者提供環顧校園的功能，帶著學生走遍逢甲校園...等等諸多功能。都是為了明白表示逢甲大學的地理環境，使大家能無礙輕易的暢遊逢甲校園。

1.2 製作動機

3D 遊戲製作是現在所熱門的，我們這個小組在一開始做專題時曾朝這個目標去收集資料、學習和研究相關資料。後來在與謝信芳老師多次洽談後，覺得要在無經驗且時間有限的狀態下去完成一套有系統以及劇情完善的遊戲實在不是一件容易的事情。所以在謝老師的建議下，我們決定嘗試 3D 校園導覽系統這個題目，在這個題目底下，我們一樣得學習如何在程式中去加入 3D 物件，並對這個物件作一些動作的描述並且力求人物動作的自然，這樣也有點寫遊戲的味道在裡頭。

1.3 製作目的

校園導覽系統，這個專題我們期許能做到的功能有以下幾點：

1. 提供多視角導覽

   可以模擬主角自己在虛擬校園中行走環繞四周的功能。

2. 最短路徑搜尋

   在指定目的地後，可以以最短路徑行走到目的地瀏覽。

3. 環境建設真實化

   把人物比例和建築物比例做出適當的調整，讓使用者有擬真的感覺。

4. 導覽系統說明功能

   使用者在每一步驟執行時，出現說明精靈，方便使用者運用。

5. 相關業務手續辦理導覽

   當使用者輸入想辦理的業務關鍵字，系統可以搜尋到相對業務流程並自動導覽，這樣一來可以知道該到什麼地方去辦理業務。

6. 試場座位位置導覽

當考生輸入準考證號碼後，立即化身為虛擬校園中的主角並開始帶領考生到他考試的座位上，這樣一來考生如果來不及在考試前一天先看過考場，也能知道大概的位置而不至於在考試當天急急忙忙得到考場應試。

7. 按鈕點選

使用大量的按鈕圖示進行導覽，減少煩雜的命令列的輸入。

第二章 系統概述

2.1 虛擬校園的歷史背景

目前在網路上看到的虛擬導覽系統，大多 Web Service 為主，大致上分做四類主流。一種是以媒體播放系統結合其他 3D 動畫程式下去製作，另一種則是以 VRML 方式下去製作，再來是工具發展型的 Virtools 和 EON。以第一類來說，逢甲大學以前的虛擬校園導覽系統就是利用此技術完成,在整個校園裡頭分成數各區域，每個區域利用攝影技術拍攝，再提供使用者連結點選線上即時播放而成。VRML(Virtual Reality Modeling Language)對 3D 圖形而言就像 HTML 在文件格式化上所扮演的角色一樣，能夠藉著簡單的文字檔便可描述整個世界的外觀。

Virtools和EON是最近比較新發展出來的3D/VR軟體。Virtools是法國發展出來的軟體，目前台灣這邊有代理商--愛迪斯科技，其最有名的實作案例便是國家歷史博物館的數位3D文物展覽，使用者只要上網點選文物照片便可以進入由Virtools所製成的網頁裡，在這個網頁之中可以做到3D文物的移動、旋轉和翻轉動作，只要連上網便可以瀏覽各式文物的3D影像。EON Studio是由瑋特擬真科技發展出來的3D/VR軟體，就整體功能上比Virtool來的強，不過可惜的是一套軟體要價二十萬，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，只能單就功能性來說是一大強力平台。

而我們想嘗試一下不一樣型態的導覽系統,所以我們選擇用軟體設計方式使用3D Game Studio下去實作這套導覽系統。

2.2 3D 圖形化的理念

為何要採用 3D 介面下去處理，裡由很簡單現今的電腦大多可以支援高載量的 3D 圖形運算，在加上所有視角以 3D 畫面來處理不會有任何死角產生，任何角度都可以一覽無遺。2D 視角的缺點，只能固定一個角度觀看，例如利用斜角 45 度的俯視 2D 圖，用連續貼圖方式作成環繞一個點的俯視觀察點，著名的線上遊戲 Ragnarok-Online 就是採用這種貼圖技巧。

我們可以用這樣來解釋 3D 化的好處，在 3D 模型的環境下我們可將模型渲染出物體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。如果在一件大廈的修繕過程中，能夠先以 3D 模型來模擬所有建築物隔間與機械系統、空間規劃以及修繕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，大概只有專業的設計師才看的懂，一般人來看專業的設計圖只會看到一堆線條和一些無法解釋的數字。

2.3 軟硬體需求

開發環境：Windows 作業系統〔winXp〕、RAM256 MB、3DS MAX R5、3D Game Studio A5、PhotoImpact 7.0。此為開發本系統時所用的個人電腦配備，並非是建議配備等級，使用者不須擁有這些開發軟體也能執行本系統。

系統建議需求如下：PentiumIII 以上的中央處理系統、RAM128MB 以上為佳、支援 3D 加速運算的顯示卡、Windows 作業系統〔win98、win2000、winMe、winXp〕。此為建議配備等級，運用 PIII 以上等級 CPU、RAM 達到 128MB 以上以及 Win98 以上作業系統是考慮到本系統執行時必須執行大量的圖形運算功能，所以有此建議。

第三章 系統分析與設計

3.1 系統開發概述

1.系統開發生命週期

系統開發生命週期(System Development Life Cycle)是公司用來建立一資訊系統的一系列步驟。

〔1〕 總體開發週期

階段	工作目的
系統規劃	規劃階段的目的就是明確辨認問題的性質與範圍。
系統分析	分析階段的目的是準確了解現有系統如何運轉，決定並紀錄系統應當做什麼，提出一些供選用的解決辦法。
系統設計	設計階段的目的是編寫滿足所有紀錄需求的資訊系統設計，並且必須確認全部必要的輸出、輸入、檔案、應用程式以及人工程序。
系統建置	在建置階段中，將撰寫、測試並紀錄應用程式，完成操作文件和程式，最後是取得用戶和管理當局的批准。其目的是提交批准的功能與紀錄完整的資訊系統。
系統運行與支援	在運行與支援階段有時需要進行維護和加強部分，以解決用戶確認的一些問題。進行維護邊更是為了應映某些特殊要求或是糾正錯誤。加強則是提高一些功能的修改。

表 3.1 總體開發週期表

這五階段包括：系統規劃、系統分析、系統設計、系統建置以及系統運行與支援。五階段彼此之間環環相扣缺一不可，故稱為系統開發生命週期。如圖3.1，從規劃開始到分析、設計形成一個小迴路，在這個小迴路裡頭可能會經歷過很多次的反覆，最後才能到達建置階段和運行與支援階段。

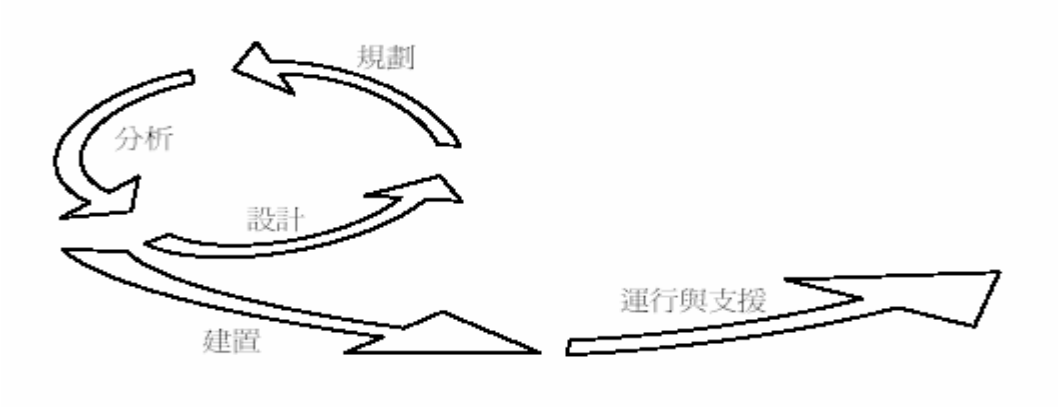


圖 3.1　總體開發週期圖

〔2〕瀑布模型

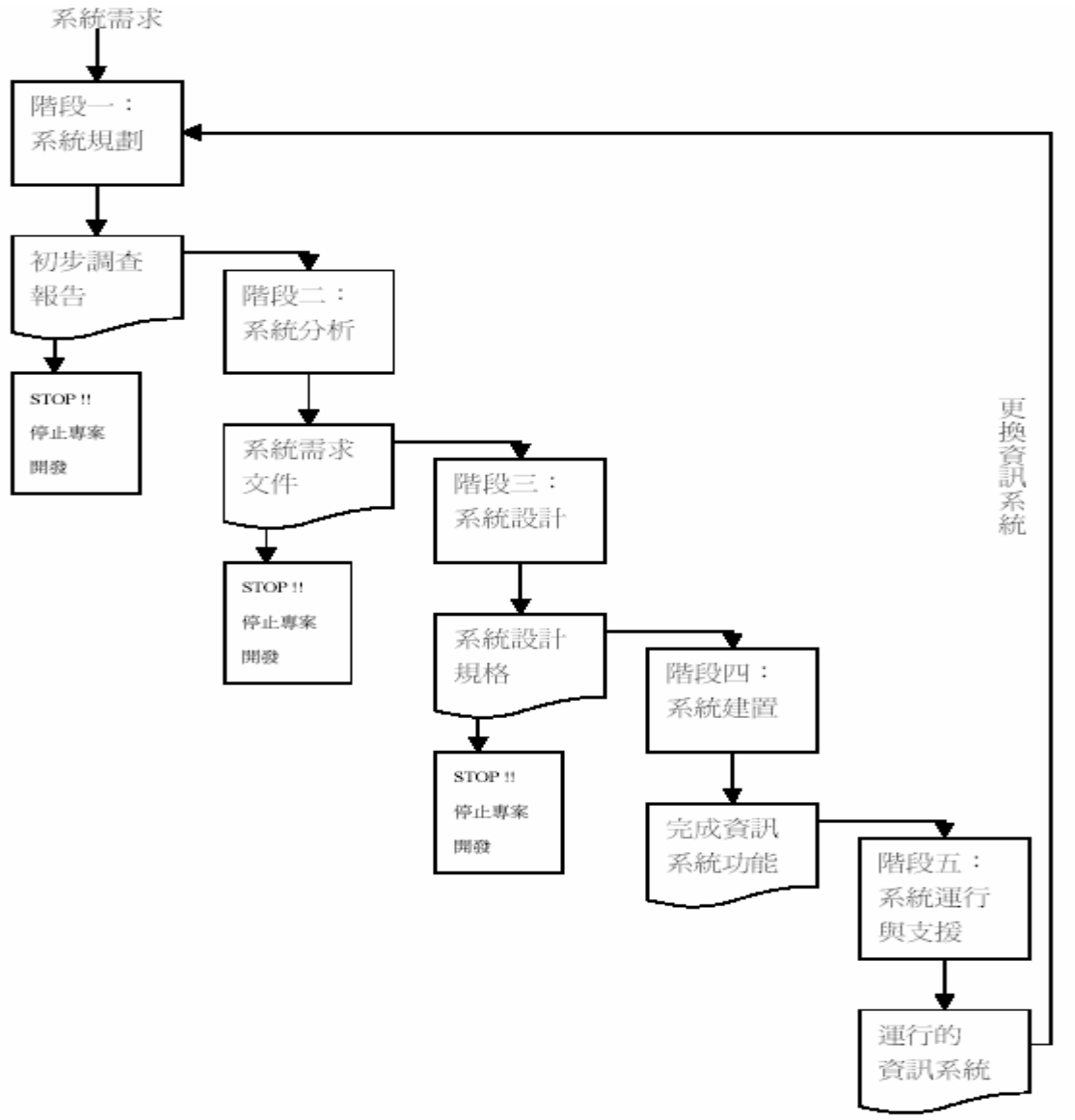


圖 3.2　瀑布模型圖

優點	缺點
統一的步驟，嚴謹標準的開發程式。確保開發的系統品質。	每一階段完成後，才能進行下一階段。系統沒開發完成前，看不到成果。
提供良好專案管理控制。	如有某一階段錯誤，可能導致專案重新來過。
清楚階段劃分，易於分工和管理。	一但某一階段工作無法如期完成，將導致後續階段工作停擺。
系統開發人員可選擇熟悉適合的工具、方法來進行系統開發。並透過各階段輸出文件來互相溝通。	過多的紙上模型和工具，可能造成額外的負擔和溝通上的誤解。

表 3.2　瀑布模型優缺點

2. 資料流圖

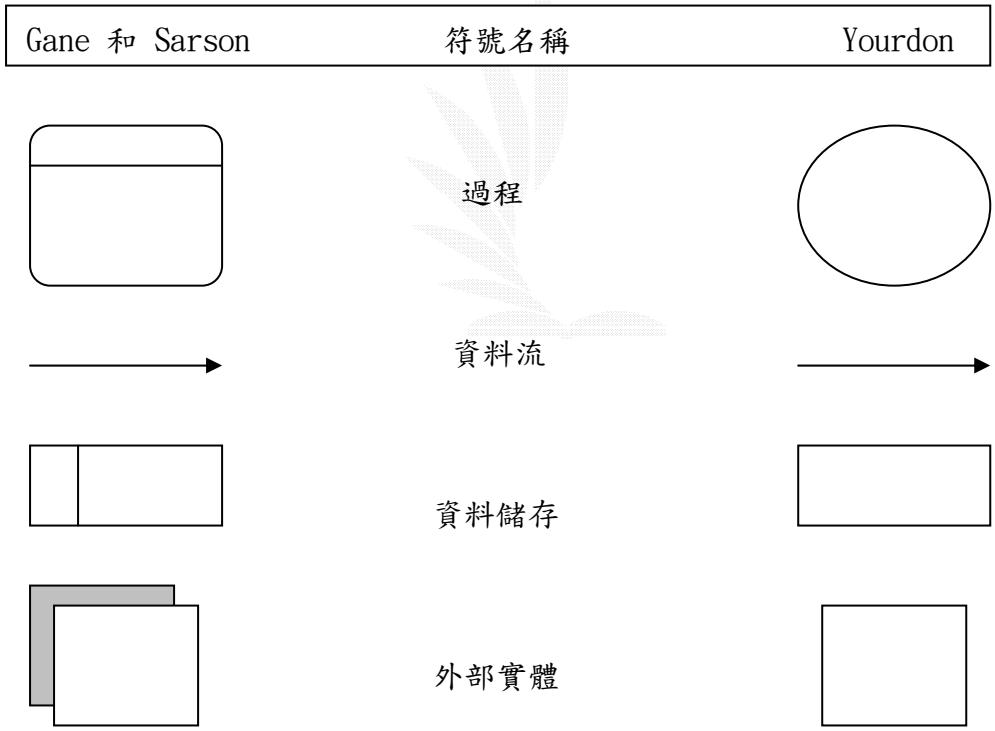


表 3.3　DFD 版本符號差異表

上面這張圖用來展示兩種不一樣的 DFD 符號集合，左邊是 Gane 和 Sarson，右邊是 Yourdon，本系統一律將採用 Gane 和 Sarson 版本的 DFD 符號表示。採用 Gane 和 Sarson 版本的主因是 Yourdon 版本其外部實體與資料儲存兩種圖示容易讓人混淆。

在使用資料儲存與過程時必須注意三種錯誤的使用，自發產生、黑洞、灰洞。自發產生是指沒有輸入，但有輸出的過程。黑洞則是有輸入，但沒有輸出的過程。最後是灰洞，灰洞指的是有輸出和輸入，但輸入的資料不可能會產生該輸出項目。

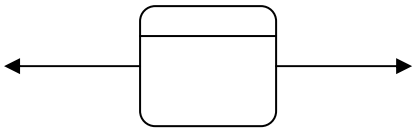


圖 3.3 自發產生

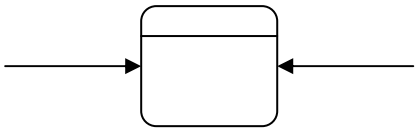


圖 3.4 黑洞

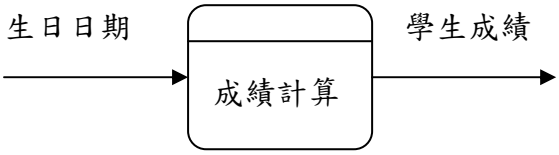


圖 3.5 灰洞

其他方面要注意的是，兩個資料儲存不能沒有過程介入其中而互相連接，資料儲存應該是有進有出的。

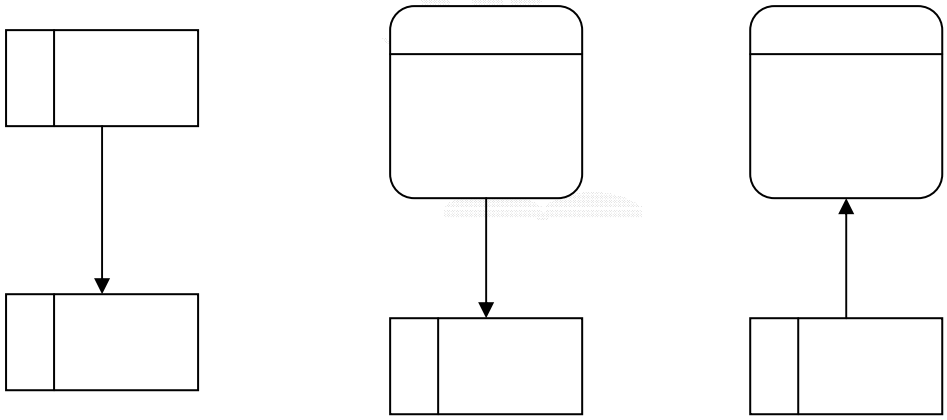


圖 3.6 資料儲存錯誤使用

外埠實體必須由資料流連接於過程，並不能直接連結資料儲存或另一個外部實體。

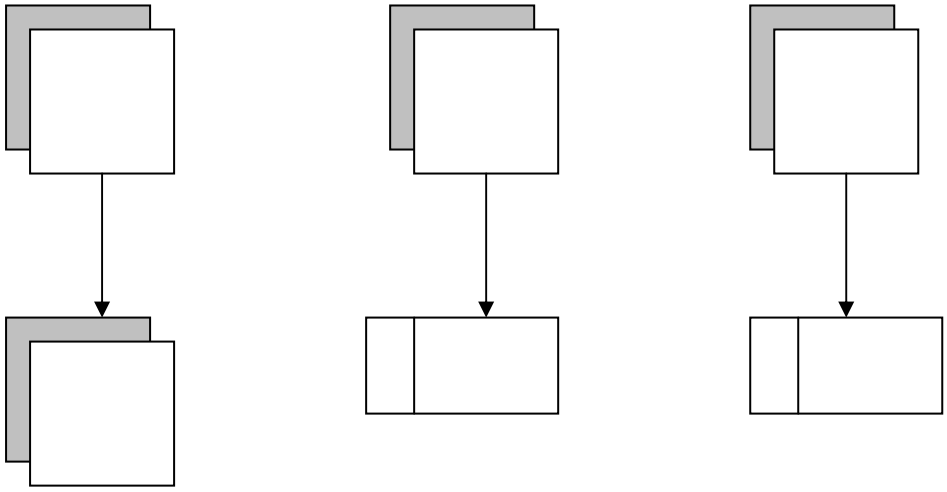


圖 3.7 外部實體錯誤使用

3.2 校園導覽系統系統分析

在這裡我們要做的是第二階段的系統分析、系統需求文件的處理部份，第三階段系統設計、系統規格書已及第四階段系統建置、系統功能將會在第四章系統實作中來陳述。第五階段系統運行與支援，我們將在第六章來做一個評估討論。

1. 系統簡述

此系統以遊戲引擎為主要核心開發出相關的應用軟體。 這套系統主要的服務對象為新生或來校參訪的來賓以及考生。在圖 3.8 系統架構圖中用圖示方法來展示這套系統的預期功能。

2.系統架構

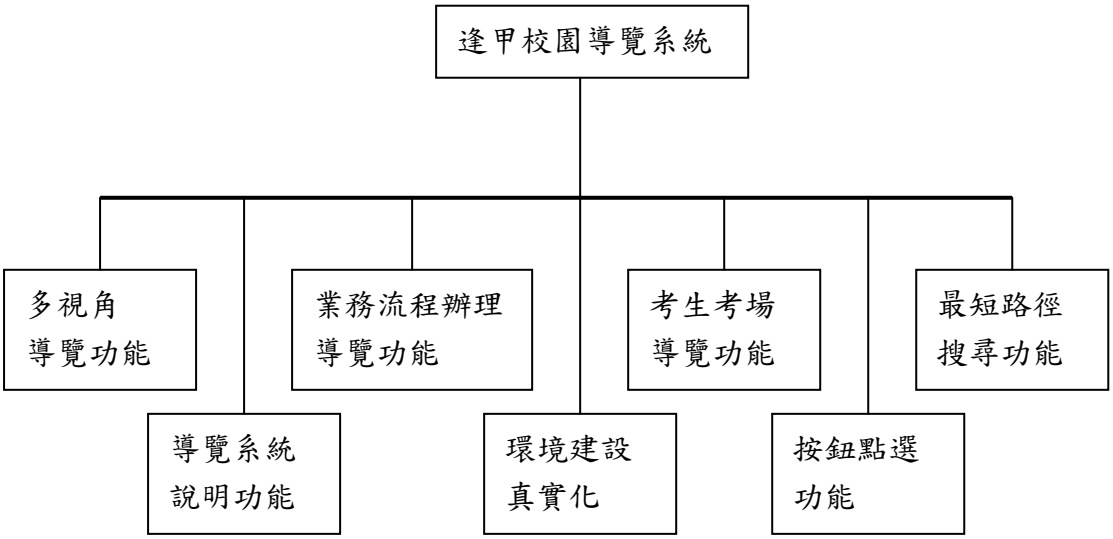


圖 3.8　逢甲校園導覽系統架構圖

在整套系統中加入多種子功能,每一種子功能的目的在於讓整個導覽系統更加友善。各項子功能的介紹在第一章中已有陳述說明。

3.系統資訊

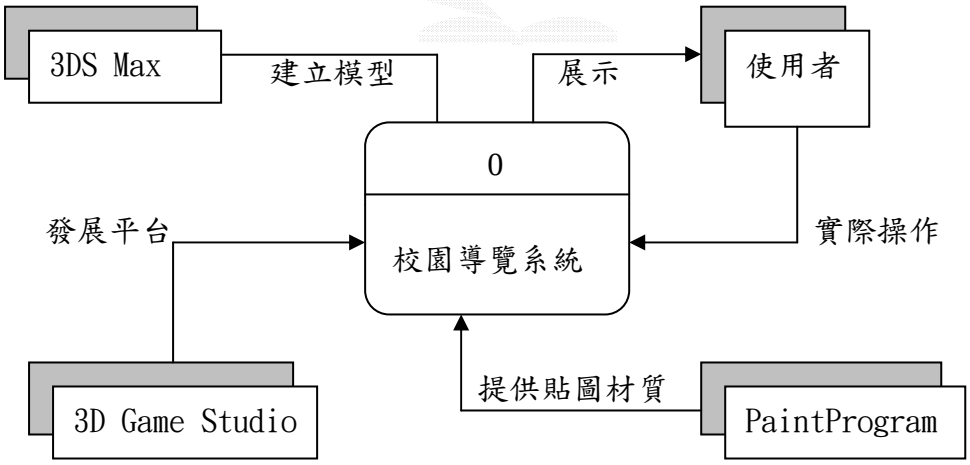


圖 3.9　系統背景圖

利用 3DS Max 做出所有的模型提供系統載入,Paint Program 提供貼圖材質，3D Game Studio 則提供發展平台，讓開發者輕易建構 3D 環境。

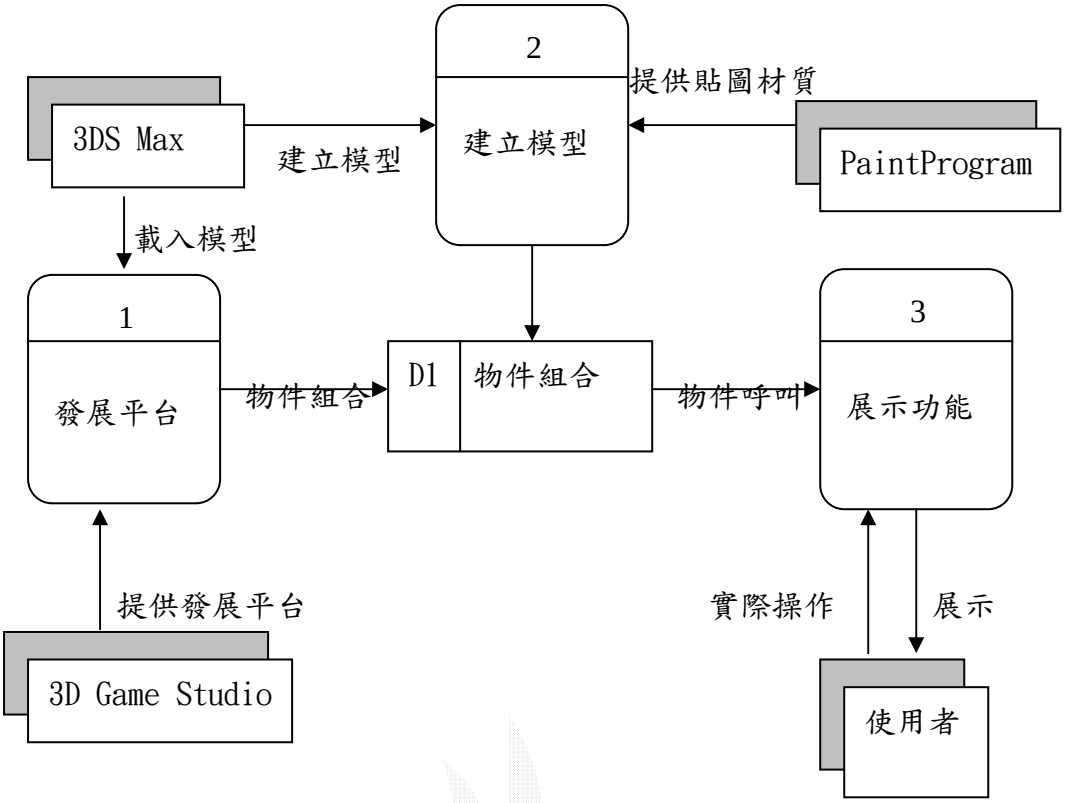

2
提供貼圖材質
3DS Max
建立模型
建立模型
PaintProgram
載入模型
1
3
發展平台
物件組合
D1
物件組合
物件呼叫
展示功能
提供發展平台
實際操作
展示
3D Game Studio
使用者

圖 3.10　系統圖 0

第四章 系統實作

系統實作部份，整個系統分成主角（player）、資訊站（path_entity）、路徑（path)與事件觸發元件（event_entity)等四個主要元件，我們將逐一介紹如何去運用出這些元件去開發整套導覽系統。圖 4.1 是這套系統的開發環境。

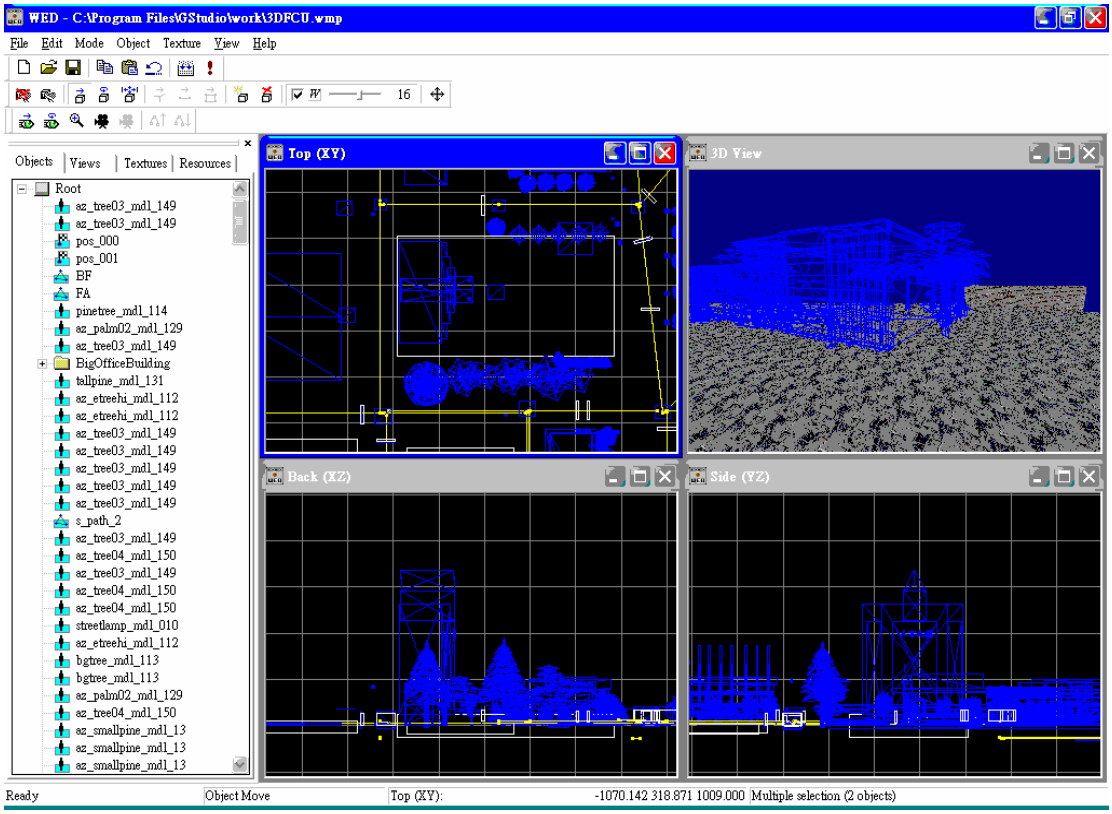


圖 4.1　開發環境

4.1 開發工具

1.　3D Studio Max

3D Studio Max 是國內最早引進，也是最普遍的一套國際級 3D 動畫軟體；其應用的行業範圍非常廣泛，從建築透視動畫、室內景觀設計虛擬、電影電視廣告影片、商業造景動畫及電玩遊戲，都可看到 3D Studio Max 的蹤跡。多樣

豐富的外掛程式，是 3D Studio Max 重要特色之一。

3D Studio Max 是一個非常強而有力的 3D 模型製作軟體。市面上有許多有名的製作軟體，如 Maya、3D Studio Max...等，我們選用 3D Studio Max 的理由是上手容易功能強大，最重要的一點是它能與另一套主要軟體 3D Game Studio 相容。

在 3DS Max 的運用上大致上分成模組建構部分、模組編輯部分和材質與外部設定等三大部分。其中模組建構部分：包含基本物件、延伸形式物件、門窗物件、複合物件、造型與成形物件、塊面物件以及 NURB 物件。模組編輯部分有：物件位向編輯、幾何物件編輯、立體造型物件編輯、進階編輯、特殊物件編輯。材質與外部設定：光源與攝影機、基本貼圖、進階材質、環境設定。

〔1〕建築物製作

本專題先在校園裡使用數位像機對建築物取數個不同角度的實景照片，再依照照片去模擬一棟完整的建築物。在建築物裡我們也做了隔間、大門、桌椅…等，也預留了電梯空間方便爾後設計電梯上下移動之用途。圖 4.2 到圖 4.7 是建築物建構過程圖，圖 4.8 商學大樓骨架模擬圖、圖 4.9 商學大樓門口骨架模擬圖與圖 4.10 商學大樓模擬圖則是完成圖。

圖 4.2 用來描述開始設計一棟建築物時的起始步驟，點選 line 按鈕用來圈出一定範圍作為建築物的底盤。圖 4.3 直接點選剛剛圈出的範圍，再點選 Amount 框格設定建築物高度，並且在 Segment 框格設定表示要將此高度的建築物分成幾層。圖 4.4 到 4.6 是運用到組合物件的功能，組合物件是用來作為製作窗戶以及大門的特殊功能。

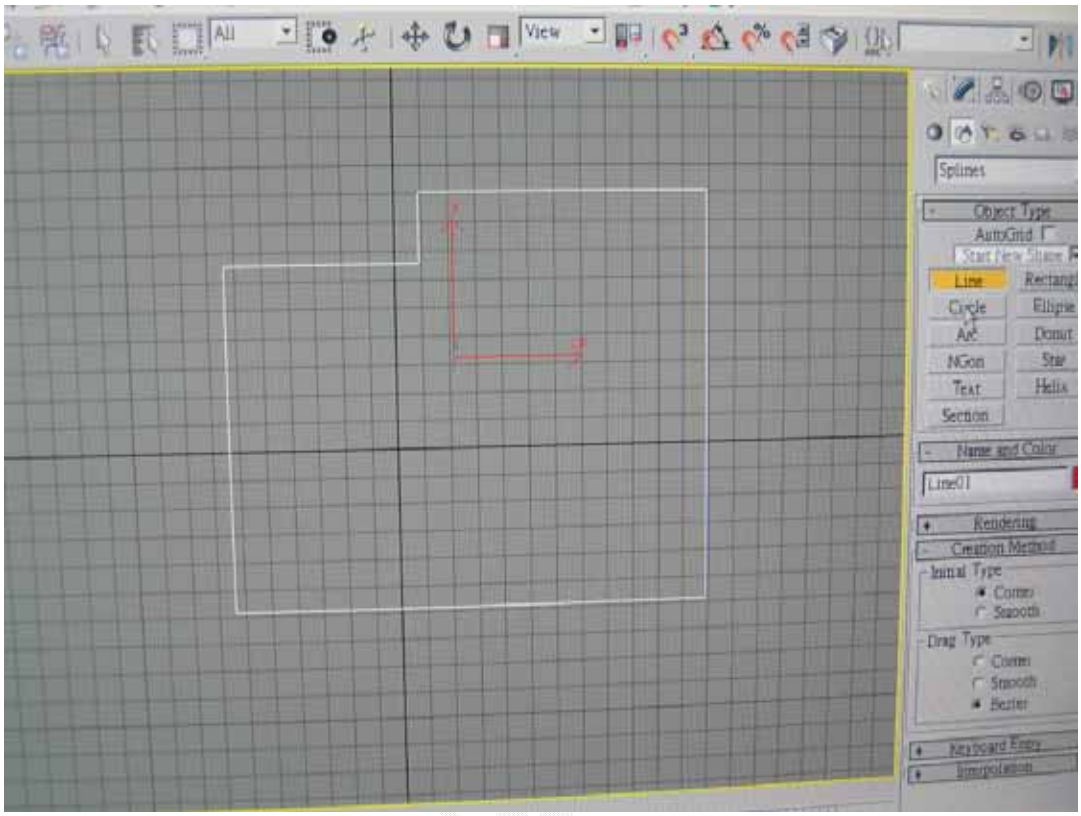


圖 4.2 建築物建構圖一

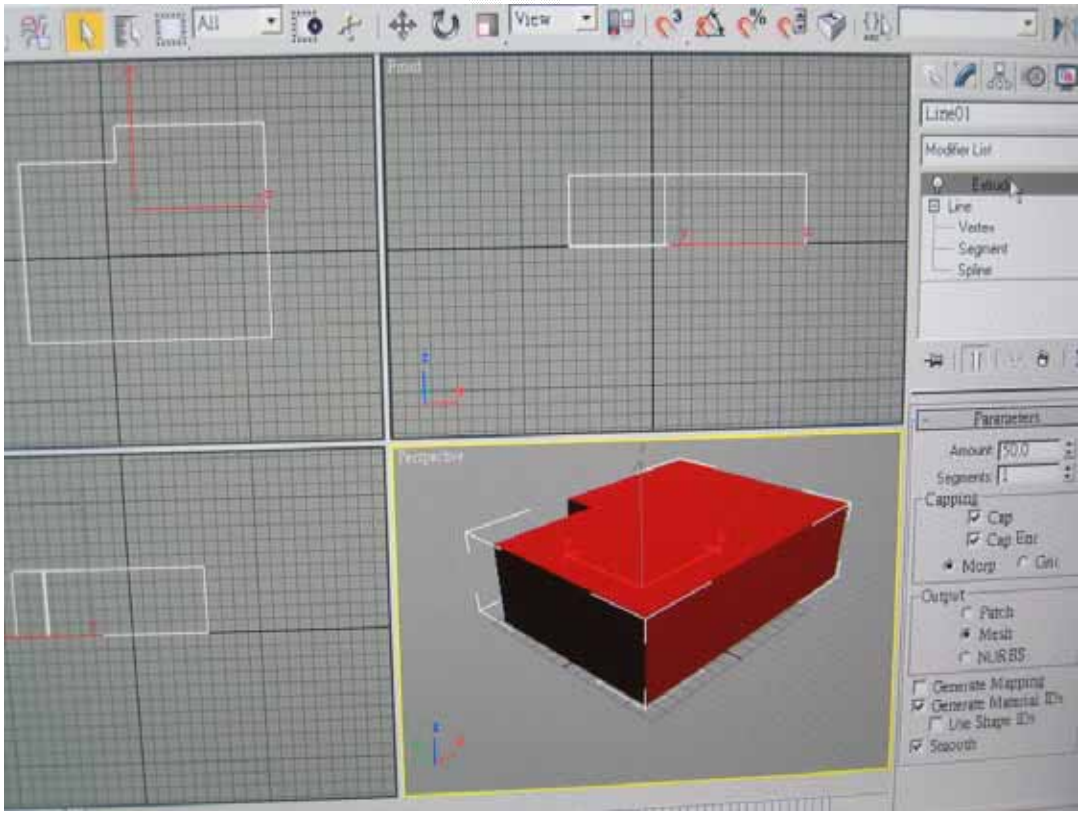


圖 4.3 建築物建構圖二

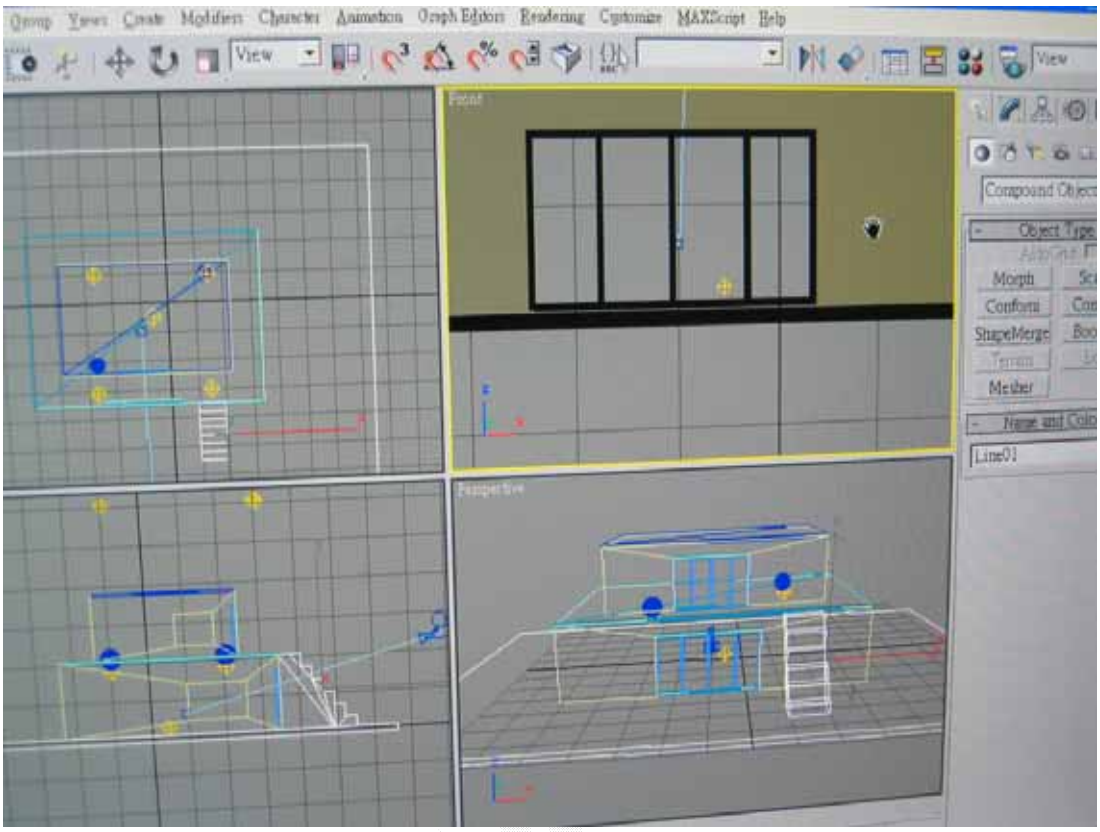


圖 4.4 建築物建構圖三

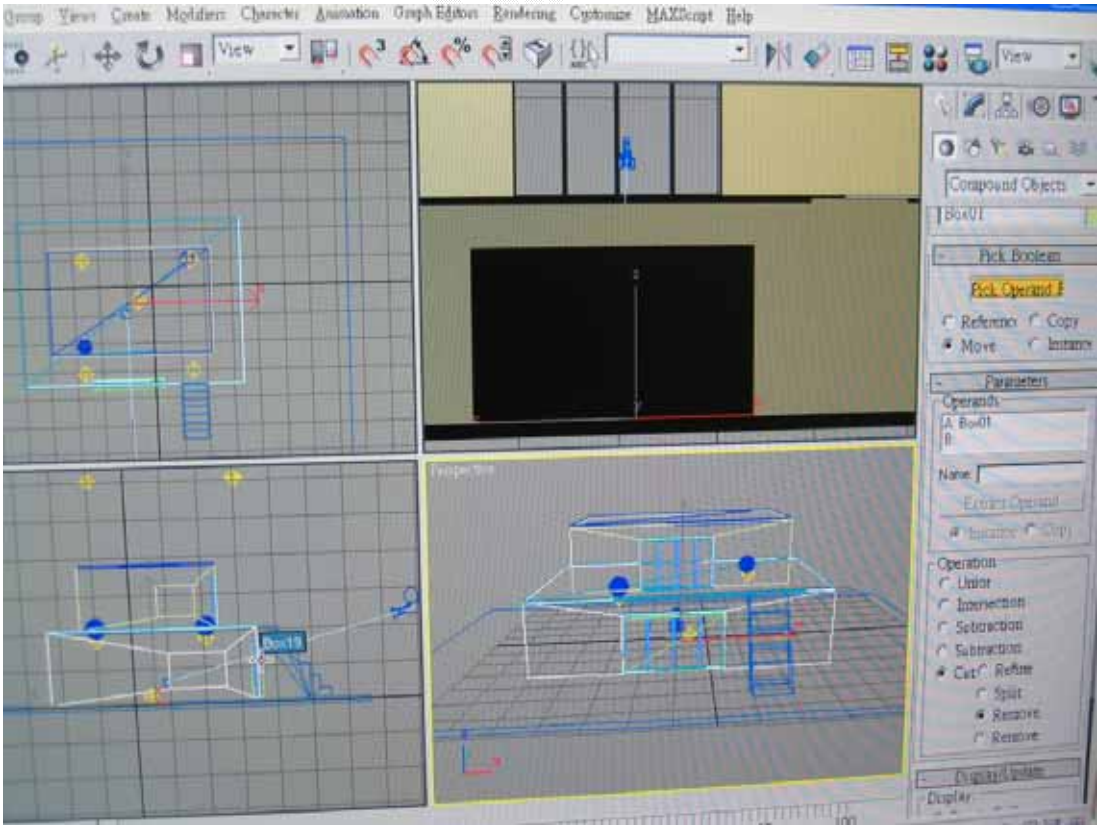


圖 4.5 建築物建構圖四

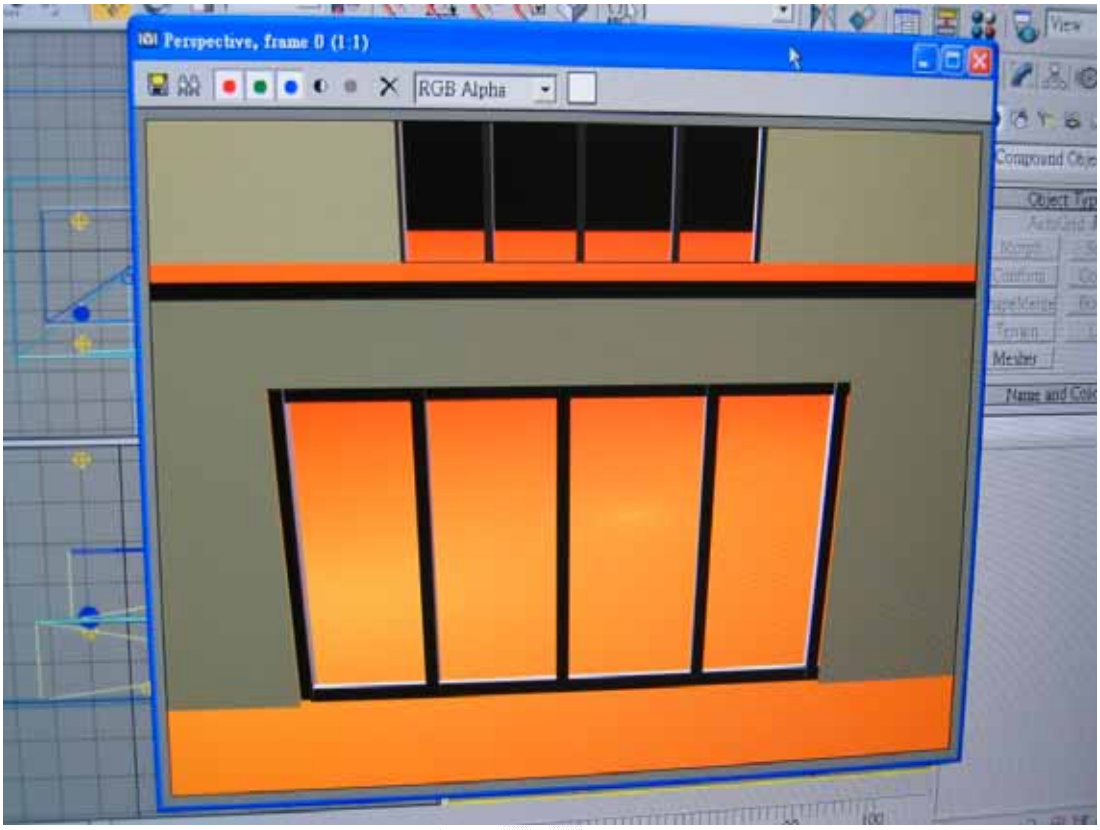


圖 4.6 建築物建構圖五

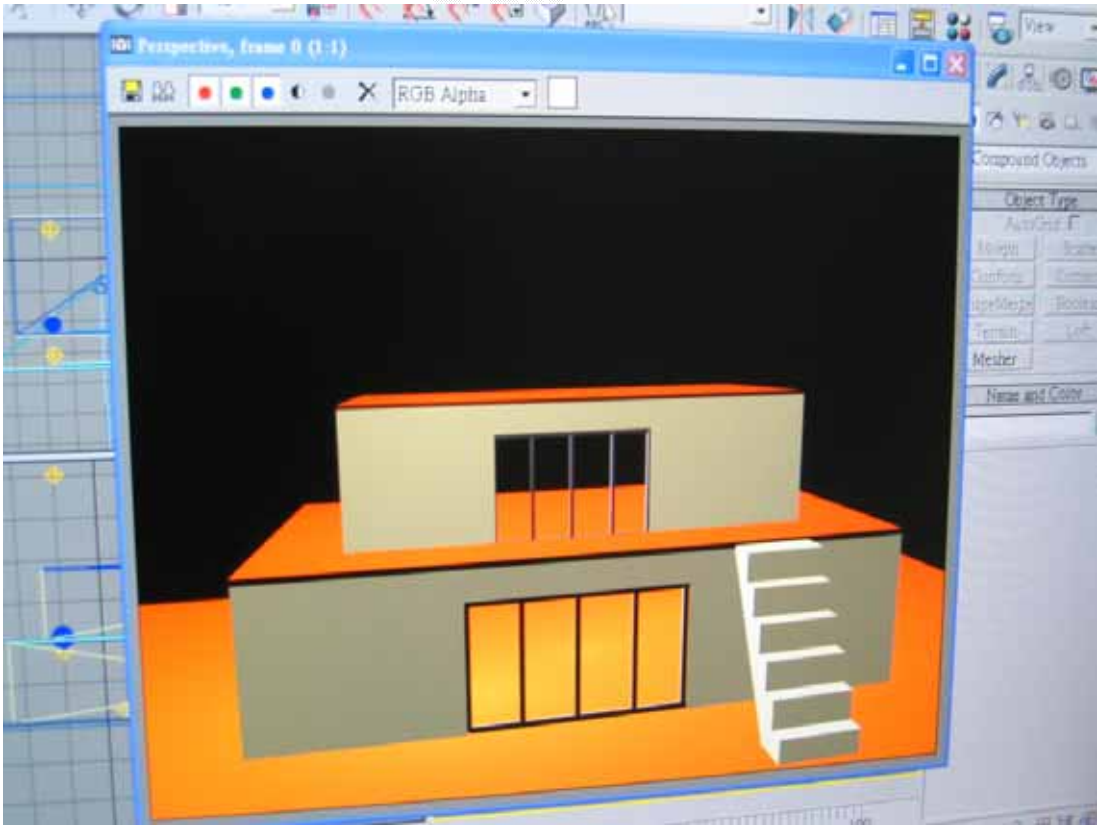


圖 4.7 建築物建構圖六

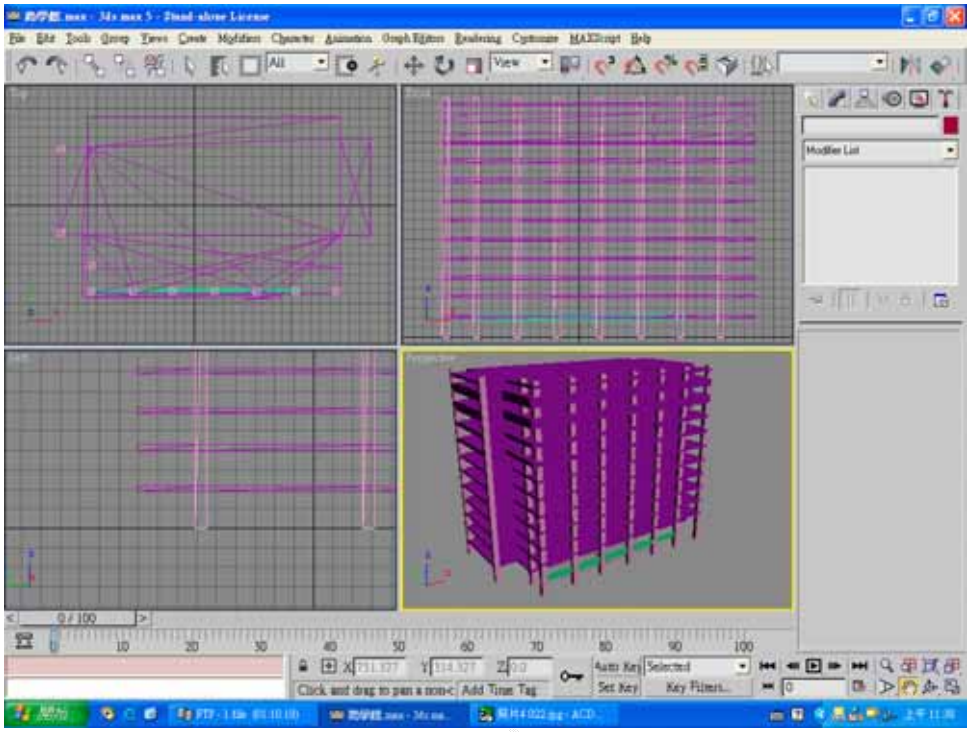


圖 4.8 商學大樓整體骨架模擬圖

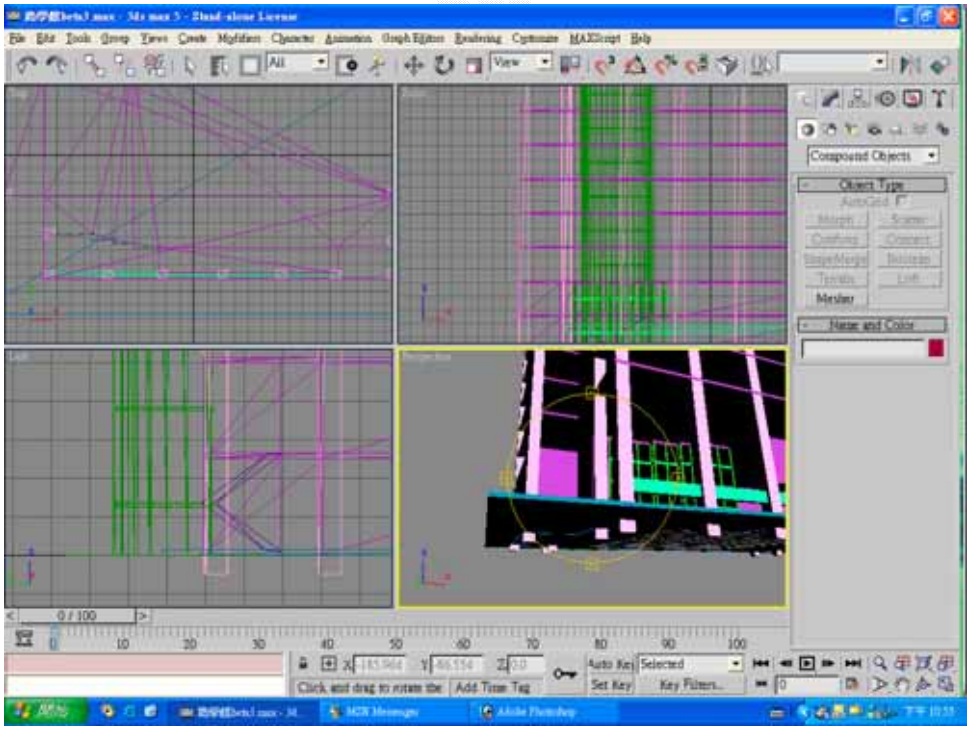
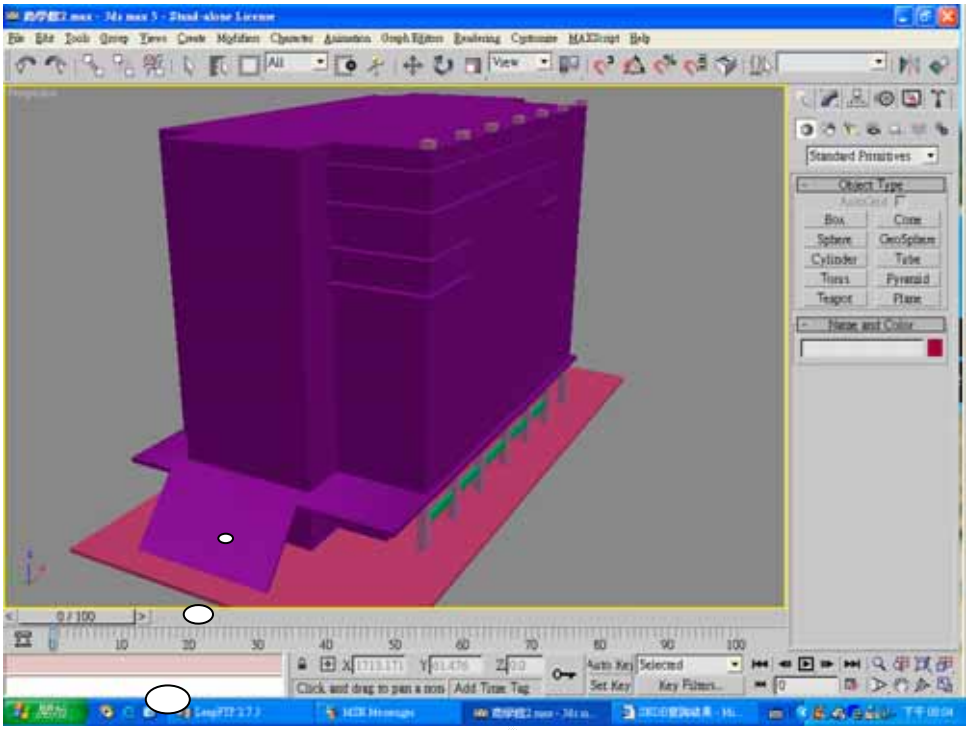


圖 4.9 商學大樓門口骨架模擬圖



測試行走
爬坡用

圖 4.10 商學大樓模擬圖

2. 3D Game Studio

〔1〕架構概述

3D Game Studio 是一套 3D 遊戲製作引擎，內含有簡易的 3D Model Maker 與 C-Script Maker。3D Model Maker 其功能就是簡化的 3D Max Studio，C-Script Maker 提供開發引擎的程序運作編寫。系統引擎如圖 4.11 所示。

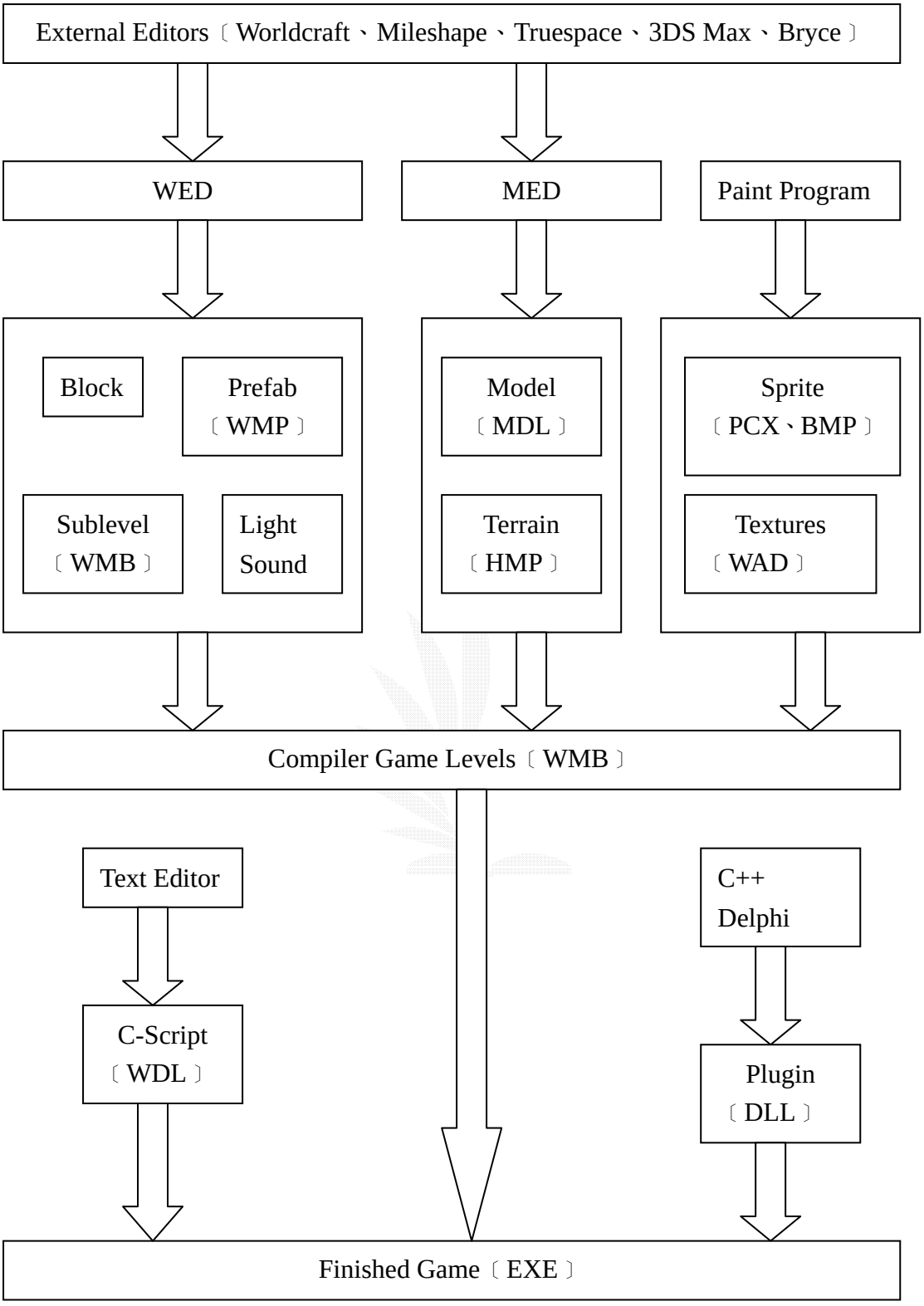


圖 4.11 3D Game Studio 引擎架構圖

圖 4.11　3D Game Studio 引擎架構圖用來說明 3D Game Studio 的引擎架構。External Editors〔Worldcraft、Mileshape、Truespace、3DS Max、Bryce〕是指用來編輯 WED 和 MED 的額外編輯器，這些編輯器包含 Worldcraft、Mileshape、Truespace、3DS Max、Bryce 等有名的軟體。

如圖 4.11a ，WED 代表的是整個世界環境建設，一個世界的建設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〕。

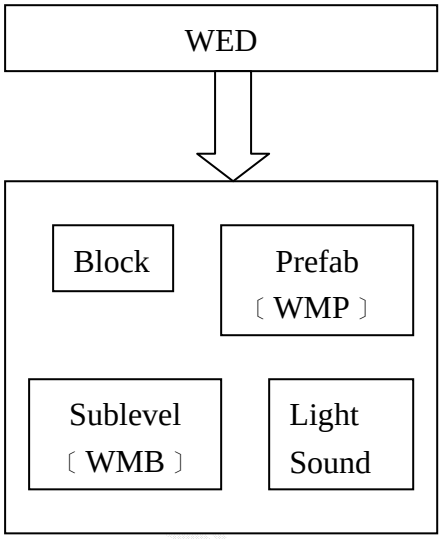


圖 4.11a　WED 子系統

如圖 4.11b ，MED 代表的是模型建造，這些東西像是人物模型、環境建築物模型、材質貼圖也包含在其中。

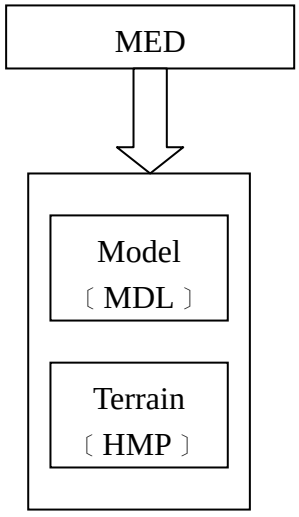


圖 4.11b　MED 子系統

如圖 4.11c，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。

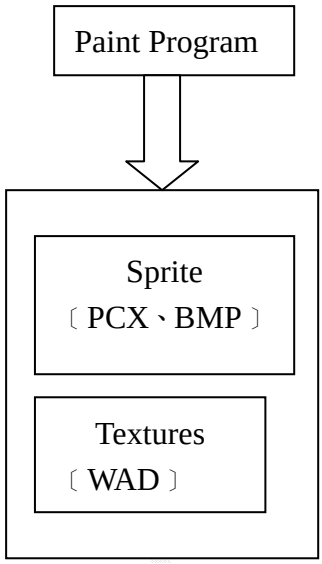


圖 4.11c 繪圖程式

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。

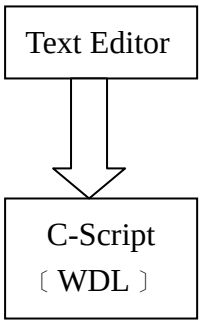


圖 4.11d 動作語法

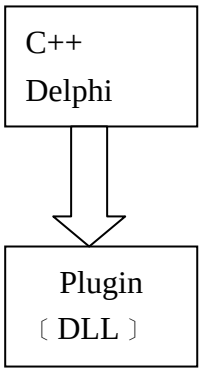


圖 4.11e 動態函式庫

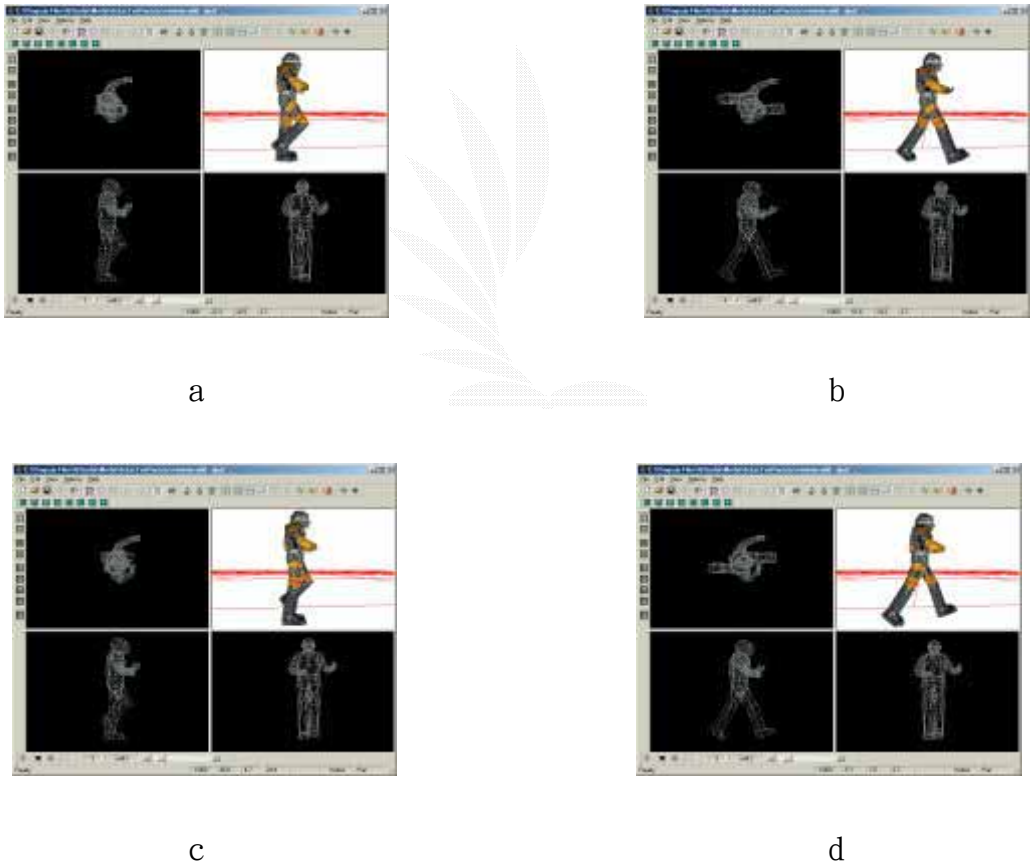
在完成 WMB 的製作後，為了要做出在世界裡面的所有動作，C-Script 和其他程式編輯器在這時後被運用上。如圖 4.11d，C-Script 可以用來製作所有物件的動作語法。這邊的動作可包含有人物的行走、陽光的照射角度、樹葉隨風飄逸、雲朵的飄動、門的開關、電梯的垂直移動…等等。

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。

4.2 功能建置

1. 人物移動

在 3D Game Studio 的功能中把人物移動的動作用框架播放方式連續呈現。也就是，我們先利用 3D 軟體編輯製造出多種動感的 3D 模型人物，利用連續播放框架技術做出人物移動的感覺。下面我們展示幾種人物的姿勢動作，從圖 4.12 來看，一個完整的行走過程是從 a->b->c->d。為了更深一層去了解，最後我們放上與這些功能有關的相關演算法。

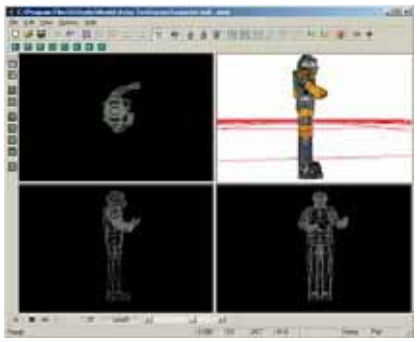


a　　b

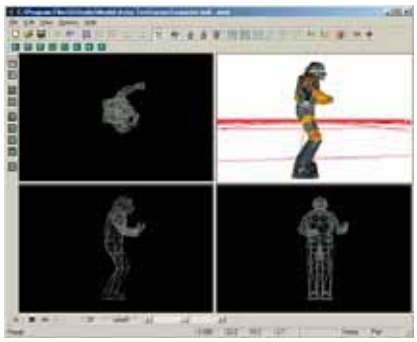
c　　d

圖 4.12 人物行走 3D 姿態

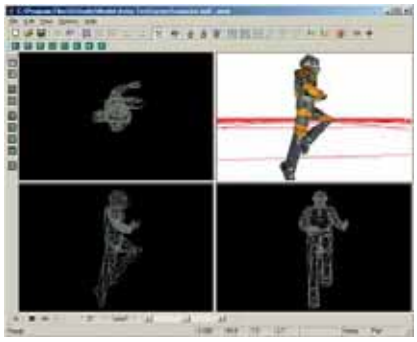
如圖 4.13 來看，一個完整的行走過程是從 a->b->c->d。



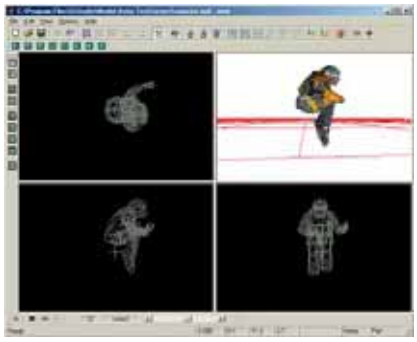
a



b



c



d

圖 4.13 人物跳躍 3D 姿態

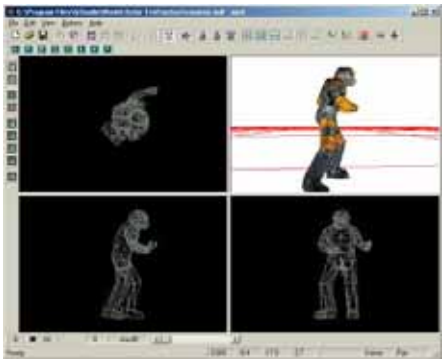


圖 4.14 人物站立 3D 姿態

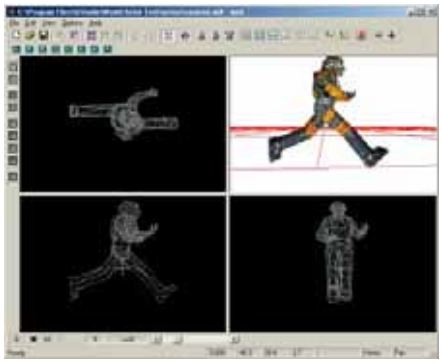


圖 4.15 人物跑步 3D 姿態

```
function    actor_anim_transition(str_anim_target)  //狀態轉換
{
    MY._ANIMDIST = 0;                              //轉換編號預設為 0
    temp = MY.frame;                               //把目前動作放到暫存起來
    ent_frame(str_anim_target,0);
    MY.next_frame = MY.frame;                      //把動作變換指到下一個動作

    anim_trans_cycle = MY._MOVEMODE;     //狀態變數指到動作轉換循環變數
    MY._MOVEMODE = _MODE_TRANSITION;     //轉換變數指到狀態變數
    while(MY._ANIMDIST < 4)
    {   //這部分說明連續動作的變換
        MY.frame = int(MY.frame) + (MY._ANIMDIST / 4.0);
        MY._ANIMDIST += TIME;
        wait(1);
    }
    MY._ANIMDIST = 0;                              //將變數回到預設值
    MY._MOVEMODE = anim_trans_cycle;     //動作轉換循環變數指到狀態變數
}
```

演算法 4.1 人物移動 狀態轉換

2. 環境建設

虛擬校園環境的安排是參照逢甲大學的實際環境去做規劃。以圖 4.16 逢甲大學校園分佈圖為準，針對每一棟大樓和校園內部環境做路徑規劃的設計。圖 4.17 和圖 4.18 是本專題中用來設計校園路徑的設計圖。

圖 4.16 逢甲大學 校園分佈圖

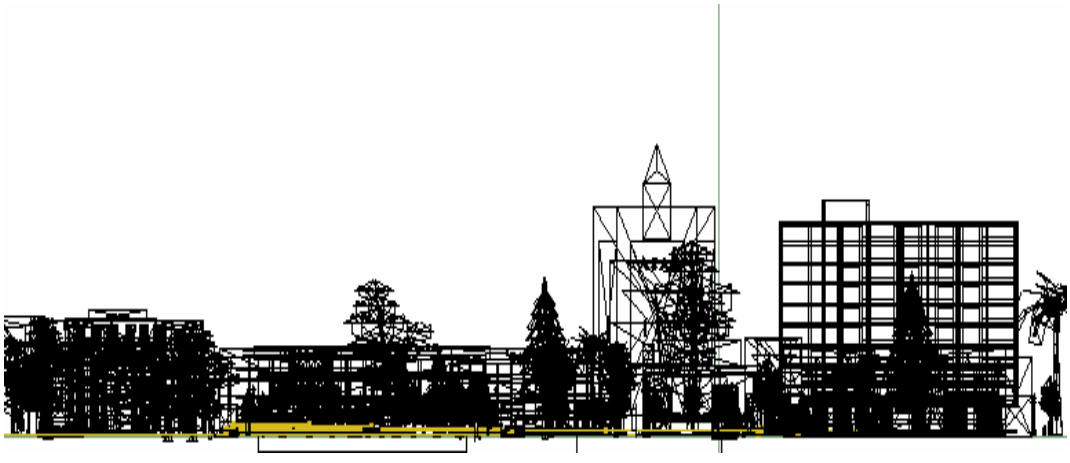


圖 4.17 校園路徑設計圖 側面

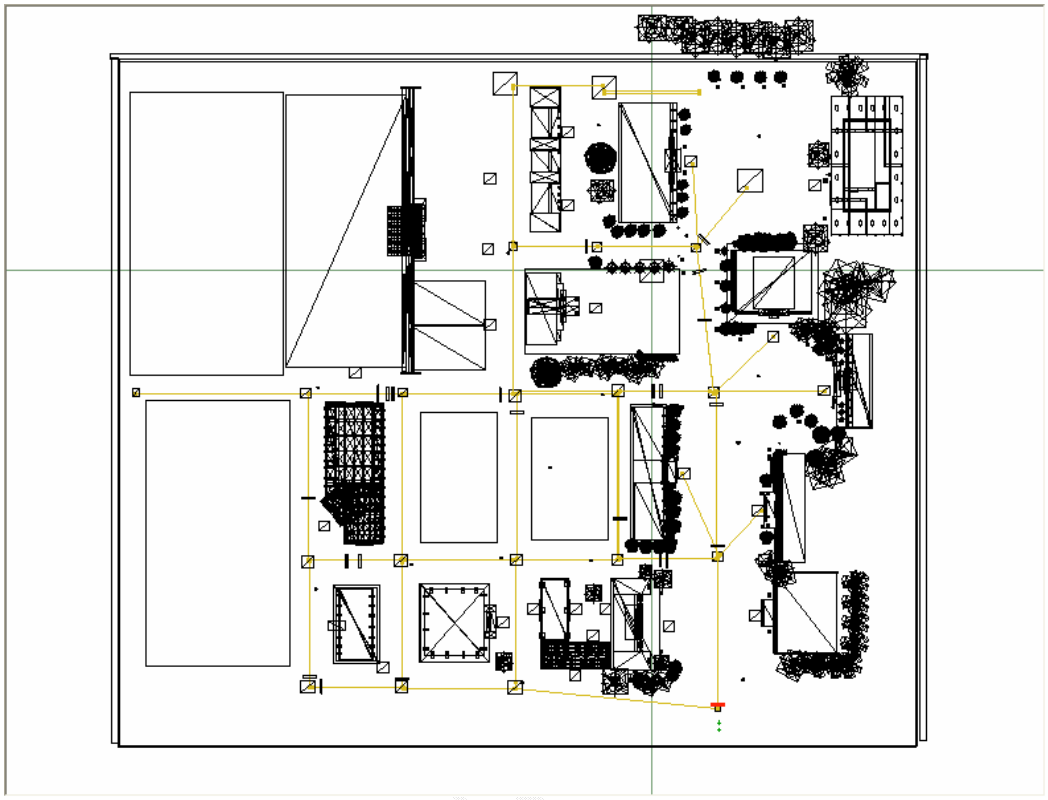


圖 4.18 校園路徑設計圖 俯視

3.路徑控制

〔1〕自由模式和自動導覽

在我們的系統之中，一開始可選擇自由移動或是自動導覽， 當主角在自由模式下行動時，使用者可以用鍵盤或是滑鼠來操控在 3D 校園裡任意走動。如果選擇自動導覽模式，則選好目的地後，系統將會自動帶到目的地，並且人物可以在任意時間內，轉換兩種模式(按畫面上的紅色大圓球)，如圖 4.19 右下方的大圓球。

圖 4.19 校園導覽系統 執行畫面

自動導覽途中，使用者可以停下來切換回自由移動模式；相反的，使用者也可以從自由也可轉回自動導覽模式。在路徑規劃完善的情況下，我們能保證主角能走到目的地。最後我們放上與這些功能有關的相關演算法。

```
function  auto_to_free()   //自動轉自由模式
{ me=player;
  update_v=0;
  close_now_to();          //按下控制鍵後結束目前的行走
  player_walk();           //跳離到人物行走函式  }
function a_free()
{  update_v=0;
   fcu_move_mode=1;  }
function free_to_auto()   //自由轉自動模式
{ update_v=0;
  fcu_auto_time=1;
  choose_place();          //按下控制鍵後跳離到目的地選擇函式 }
```

演算法 4.2 路徑處理 導覽模式切換

〔2〕路徑偵測和規劃

路徑偵測和規劃是花最多時間的部分這個子系統是整個導覽系統的最大特點。為了完成這個系統必須要先學會 3D Game Studio 平台的 C-script 語言，還有可能用到的內建功能然後依照這個平台所提供的內建功能來規劃路徑。起初完成的功能，是主角可以沿著一條有很多節點的路徑行走，依照這種想法下去規劃，我們發現只能有一個起點，而且路徑會變的很多，並且不能隨地開始導覽而且有幾個目的地就幾條路徑這樣下來，如圖 4.20 路徑規劃，路徑的數目可能會上達百條而且雜亂無章規劃者也不容易去做好路徑規劃，所以並不理想。

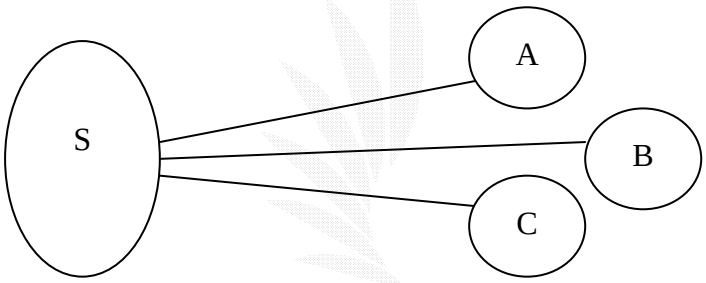


圖 4.20 路徑規劃

後來改良為在 3D 校園中設立多個站點就像是公車站一樣。

第一次改良---觸碰式站點設計。依照這種方法 如果要導覽的話，主角只要走到站點觸碰一下， 就可以開始自動導覽。如此一來不管是規劃路徑的人好規劃，使用者也比較方便，但是其缺點還是不能隨時隨地就切換自動導覽。

第二次改良---隨意切換導覽模式。為了實現這種理想情況，也就是自動及自由導覽模式能隨時隨地切換，規劃路徑的人也能可以容易規劃。我們引進了資訊站(path_entity) 圖 4.21 這個偵測元件，這個元件在執行模式時是個透明可

穿透式物件，更貼切的說法是此元件像是一個路由站---紀錄著通往各地的路徑。當主角走到此時，此元件便會通知主角下一個路徑的位置，這樣提升了很多規劃路徑的彈性。

第三次改良---在本專題中，逢甲校園內共設定有 20 個資訊站，作為判斷路徑用，若每個資訊站若按照原來的方法設計，則必須紀錄非常大量的資訊，耗時浩工不符合效益，且在以後的程式碼維護上變成是一種大問題。這些問題來自於，如果要在增加一個資訊站，我們則必須去針對每個資訊站做大幅度的修改。於是我們最後在資訊站的實作上改良成，資訊站只紀錄到某站的路徑一直走到最接近目的地的資訊站時才告知直接通往某大樓的路徑，這樣一來在規劃路徑上就會方便許多並且在選擇目的地時，只要讓系統知道最接近的目的站和大樓名稱兩個條件就能無誤行走到目的地。

另外之後在程式碼的維護上也很快速，因為整個路經偵測系統與目的地大樓是分開的，要新增一條新路徑和目的地都變的很容易。

最後我們放上與這些功能有關的相關演算法。

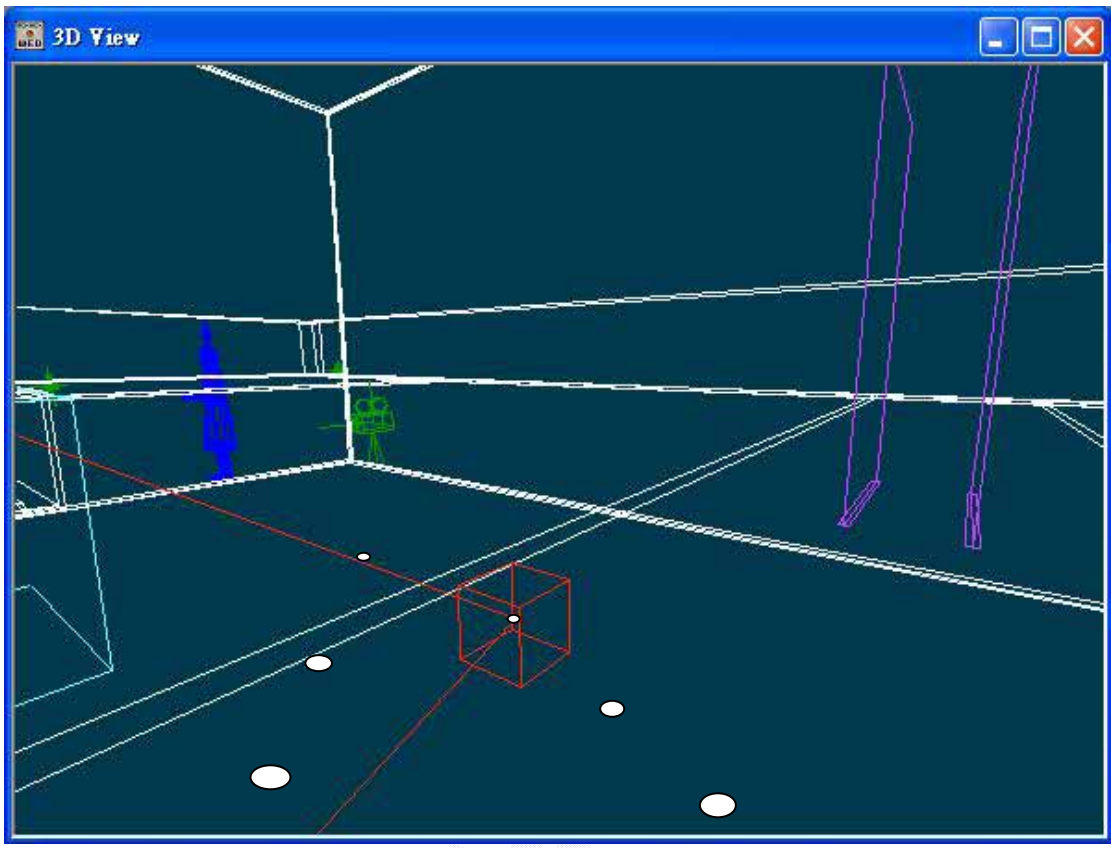


圖 4.21 資訊站


```
function scan_event       //路徑資訊區域通知
{
    str_for_entname(info_station_name,me);
    me=player;

if (EVENT_TYPE == event_scan)
{
if (str_cmpi(info_station_name,"F")==1)    // F 站
 {
   F_station_main();   // F 站主程式
 }
  function F_station_main()
{
  find_next_at_F_station();   // F 站次程式
 }
```

演算法 4.3 路徑處理 資訊站處理

```
function find_next_at_F_station()
{
me=player;
if  (str_cmpi(player_info_place_building,"iecs")==1)
//若這是最進站 則通知直達路徑
{
   ent_path("to_iecs");
   ent_waypoint(my._TARGET_X,1);
   return;
  }
   if (str_cmpi(place_entity,"A")==1)
                    //通知往 A 站的路
  {
   ent_path("FA");
   ent_waypoint(my._TARGET_X,1);

  }

  if (str_cmpi(place_entity,"B")==1)
                   //通知往 B 站的路
  {
   ent_path("FB");
   ent_waypoint(my._TARGET_X,1);

  }
  // 自動導覽主程式
  // 說明:角色隨一路徑行走 並且能偵測資訊站
  action patrol_path
{
  actor_init();

   my._FORCE=5;

    if (result == 0) { my._MOVEMODE = 0; }  // 找不到路徑 則停止

    while (my._MOVEMODE > 0)
```

演算法 4.3 路徑處理 資訊站處理(續)

```
{
                //經過特殊事件觸發元件
                result = content(my.x);
                if (result == CONTENT_PASSABLE)
                  {
                   temp.pan = 120;
                   temp.tilt = 120;
                   temp.z = 200;
                   result=scan_entity (my.x, temp);
                                }
        // 主角方向
        temp.x = MY._TARGET_X - MY.X;
        temp.y = MY._TARGET_Y - MY.Y;
        temp.z = 0;
        result = vec_to_angle(my_angle,temp);
        force = MY._FORCE; //速度
        // 接近路徑尾端 偵測路徑
        if (result < 50) {
        result=ent_nextpoint(my._TARGET_X);
        if (result==1)
         {
             if (with_serial_path==1)   //串列路徑
            {
                if  ( PATH_CURRENT < = PATH_TOTAL )
                {
                     find_next_path();
                     PATH_CURRENT += 1;
                 }
                 else{end_with_serial_path(); }
             }
                else {  temp.pan = 120;
                        temp.tilt = 120;
                        temp.z = 200;
                        result=scan_entity (player.x, temp); }
//掃描所在區域,掃到的話會自動通知新路徑
// 已經到達目的地
             if (result==0) {break;}
```

演算法 4.3 路徑處理 資訊站處理(續)

```
                        }   }    }
        actor_turnto(my_angle.PAN);
        actor_move();
             //固定攝影機和隨身攝影機的切換
             if (update_v==1)
               {
                 fixed_views();  //固定攝影機
                 }
             if (update_v==0)
                {
                 move_view(); //隨身攝影機 4 VIEWS
                 }
        wait(1);
        if (fcu_move_mode==1)  //切換成自由模式~
        {  break;}
    }
    auto_to_free();  //自由模式
}
```

演算法 4.3 路徑處理 資訊站處理（續）

4.切換視角

利用照相機功能的定位，在程式中我們可以用來作為切換視角的功能。共有四種視角可供切換，如圖 4.22-圖 4.25 所示，我們在人物上建立了四個像機功能只要利用 F7 控制按鈕就能進行切換。最後我們放上與這些功能有關的相關演算法。

圖 4.22 第一人稱視角圖

圖 4.23 第三人稱視角圖

圖 4.24 固定視角圖

圖 4.25 側面視角圖

```
function cycle_person_view()        //循環切換視角
{   if(person_3rd > 2)   //數值大於 2 時，按下控制鍵後數值的變換
    {   person_3rd = 0;
        return; }
    if(person_3rd > 1)   //數值大於 1 時，按下控制鍵後數值的變換
    {   person_3rd = 3;
        return; }
    if(person_3rd > 0)   //數值大於 0 時，按下控制鍵後數值的變換
    {   person_3rd = 2;
        return; }
    person_3rd = 0.5;  //在其他情況時，數值的預設值 }
```

演算法 4.4 視角切換 循環切換視角處理

```
function move_view_1st()     // 處理第一人稱視角
{
    if(_camera == 0)
    {
        CAMERA.DIAMETER = 0;
        CAMERA.GENIUS = player;
        CAMERA.X = player.X;                //像機位置處理
        CAMERA.Y = player.Y;
        CAMERA.Z = player.Z + player.MIN_Z;
        CAMERA.PAN = player.PAN;
        CAMERA.TILT = player.TILT + head_angle.TILT;
```

```
        CAMERA.ROLL = player.ROLL;
        if(my_height < 5)       //人物水平高度在 5 以下
        {
            headwave = sin(player_dist*walk_rate);   //擺頭動作
    if((player._MOVEMODE == 0)|| (player._MOVEMODE == _MODE_WALKING))
            {  //狀態數值為 0 的行走
if(((headwave > 0) && (walkwave <= 0))|| ((headwave <= 0) && (walkwave
> 0)))
                {
                play_sound(thud,30);    //走路聲音          }
                 演算法 4.5 視角切換 第一人稱視角處理
                walkwave = headwave;
        headwave = walk_ampl*(abs(headwave)-0.5);   //擺頭動作       }
        if(player.__BOB == ON) { CAMERA.Z += headwave;  }
        person_3rd = 0;     }}
              演算法 4.5 視角切換 第一人稱視角處理（續）
function move_view_3rd()              //處理第三人稱視角
{
    if ((_camera == 0) && (player != NULL))
    {
        CAMERA.DIAMETER = 0;        //像機處理
        CAMERA.genius = player;
```

```
    CAMERA.pan += 0.2 * ang(player.pan-CAMERA.pan);
    if ( (player._MOVEMODE == _MODE_PLANE)
        ||(player._MOVEMODE == _MODE_CHOPPER))
    {
        CAMERA.tilt += 0.2 * ang(player.tilt-CAMERA.tilt);      }
    else
    {
        CAMERA.TILT = head_angle.TILT;
        if((person_3rd < 1) && (camera_dist.Z == 0))
        {
```

演算法 4.6 視角切換 第三人稱視角處理

```
camera_dist.Z = -(player.MAX_Z-player.MIN_Z)*eye_height_up;}    }
vec_set(temp,temp_cdist);
CAMERA.X += 0.3*(player.X - temp.X - CAMERA.X);  //像機處理
CAMERA.Y += 0.3*(player.Y - temp.Y - CAMERA.Y);
CAMERA.Z += 0.3*(player.Z - temp.Z - CAMERA.Z);
temp = ent_content(NULL,CAMERA.X);
    if((temp == CONTENT_SOLID) && (camera_solidpass == 0))
    {   //畫面變換
        temp_cdist.X *= 0.7;
        temp_cdist.Y *= 0.7;
        temp_cdist.Z *= 0.7;        }
```

```
        else
        {  //畫面變換
temp_cdist.X += 0.2*(player.MAX_X + camera_dist.X - temp_cdist.X);
temp_cdist.Y += 0.2*(player.MAX_Y + camera_dist.Y - temp_cdist.Y);
temp_cdist.Z += 0.2*(player.MAX_Z + camera_dist.Z - temp_cdist.Z);  }
        if (temp == CONTENT_PASSABLE)
        {
            if (FOG_COLOR != _FOG_UNDERWATER)   //打霧功能（未使用）
            {
                current_fog_index = FOG_COLOR;
```

演算法 4.6 視角切換 第三人稱視角處理（續）

```
                FOG_COLOR = _FOG_UNDERWATER;    }       }
        else
        {
            if (FOG_COLOR == _FOG_UNDERWATER)
            {
                FOG_COLOR = current_fog_index;          }       }
                person_3rd = 1;         }}
```

演算法 4.6 視角切換 第三人稱視角處理（續）

```
function move_view_orbit()     //隨身視角
{
    CAMERA.DIAMETER = 0;          //像機處理
```

```
CAMERA.GENIUS = PLAYER;
CAMERA.X = PLAYER.X + (orbit_camera_dist * SIN(orbit_camera_pan));
CAMERA.Y = PLAYER.Y + (orbit_camera_dist * COS(orbit_camera_pan));
CAMERA.Z = PLAYER.Z + orbit_camera_zOff;
TOUCH   NULL,CAMERA.POS;
IF (IN_PASSABLE)
{
    FOG_COLOR = _FOG_UNDERWATER;    }
ELSE
{
```

演算法 4.7 視角切換 隨身視角處理

```
FOG_COLOR = current_fog_index;   }
temp.X = player.X - camera.X;      //畫面變換
temp.Y = player.Y - camera.Y;
temp.Z = player.Z - camera.Z;
TO_ANGLE temp,temp;
camera.PAN = temp.PAN;
camera.TILT = temp.TILT;  }
```

演算法 4.7 視角切換 隨身視角處理（續）

```
function move_view_chase()    //移動追逐視角
{   if ((_camera == 0) && (player != NULL))
    {   CAMERA.DIAMETER = 0;
```

```
        CAMERA.genius = player;
        vec_diff(temp,nullvector,chase_camera_dist);
        vec_to_angle(chase_camera_ang,temp);
        chase_camera_ang.roll = 0;
        vec_set(camera.x,chase_camera_dist);
        vec_rotate(camera.x,player.pan);
        vec_add(camera.x,player.x);
        vec_set(camera.pan,player.pan);
        ang_rotate(camera.pan,chase_camera_ang);    }}
```

演算法 4.8 視角切換 追逐視角處理

```
function fixed_views()          //固定視角處理
{if (fixed_cam == null) { return; }
if (update_v==0) { return; }
vec_set (camera.x, fixed_cam.x);
vec_set (temp, player.x);
vec_sub(temp, camera.x);
vec_to_angle(camera.pan, temp);}
```

演算法 4.9 視角切換 固定視角處理

5.其他功能

在本專題中還運用到了按鈕介面以及事件觸發元件，如圖 4.26 為事件觸發元件。圖 4.29 是 GUI 按鈕介面，設計控制按鈕方便操控系統。事件觸發元件(event_entity)提供固定式攝影機或是動態事件觸發，跟環境有關，如圖 4.27 和圖 4.28。事件觸發元件先設定好處發時要做的事情，然後在主角行進中觸發時就一併啟動該元件的事件，譬如：如圖。最後我們放上與這些功能有關的相關演算法。

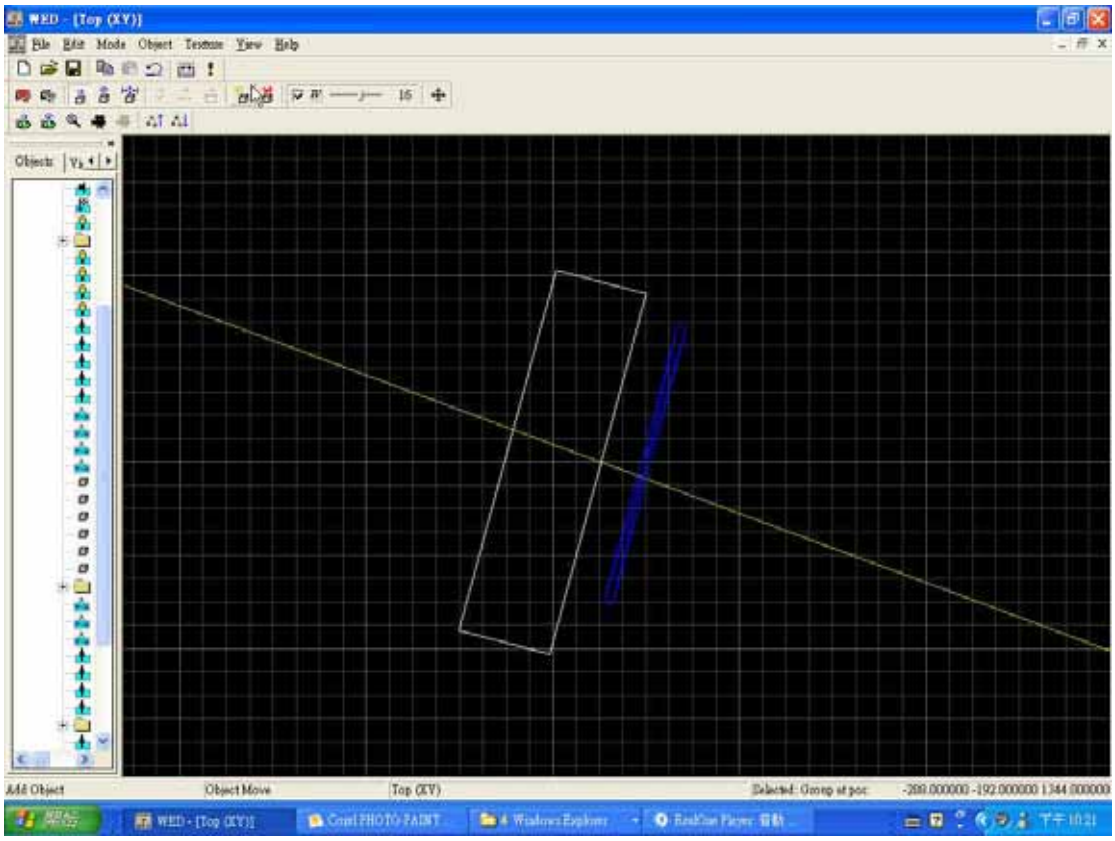


圖 4.26 事件觸發元件

圖 4.27 固定式攝影機一

圖 4.28 固定式攝影機二

```
action touch_event_1    //event_entity 物件驅動設定
{
my.ENABLE_scan = ON;
my.EVENT = what_event_1;
}

function what_event_1()   //執行的 EVENT
 {
  update_v=1;
  fixed_cam=ptr_for_name ("cam_1");
 fixed_views();
 }


action patrol_path
{
 result = content(my.x);
 if (result == CONTENT_PASSABLE)  //通過透明區塊
                          {
                               temp.pan = 120;
                               temp.tilt = 120;
                               temp.z = 200;
                               result=scan_entity (my.x, temp);
                           //偵測 ENTITY TYPE 的物件
                          }
}
```

演算法 4.10 其他功能 觸發元件

圖 4.29 GUI 按鈕介面

```
Panel test_panel
{
  pos_X = 20; pos_y = 20;    //面板位置設定
  mouse_map = test_b;
  bmap = main_intro;          //面板圖示
  flags = REFRESH,d3d,transparent;  //決定是否一開始就出現
  on_click = chengemap ;     //擊發時的動作  }
function chengemap()
{  test_panel.bmap= test_b ;     //圖示變換  }
```

演算法 4.11 其他功能 按鈕元件設計

```
function begin_title()                    //開始的畫面
{
  set system_title_panel.visible,on;    //面板設定
  set begin_panel.visible,on;
  set main_intro_panel.visible,on;
  system_title_panel.alpha = 0;
  begin_panel.alpha = 0;
  main_intro_panel.alpha = 0;
  while (main_intro_panel.alpha < 100)
{
system_title_panel.alpha+= 2*time;       //設定延遲時間
begin_panel.alpha+= 5*time;
main_intro_panel.alpha += 2*time;
wait(1);
}
}
```

演算法 4.12 其他功能 畫面秀出延遲函式

```
function text()
{
set test_panel.visible,on;      //面板設定
set ask_auto_panel.visible,on;
test_panel.transparent = ON;
test_panel.alpha = 0;
while (test_panel.alpha < 100)
{
test_panel.alpha += 2*time;     //延遲時間
wait(1);
}
wait(1000);
test_panel.alpha = 0;
set test_panel.visible,off;
wait(1);
}
on_s begin_title;            //點擊 S 控制鈕後的動作
on_f favor_choose;           //點擊 F 控制鈕後的動作
ACTION wall_touch {
SET MY.ENABLE_IMPACT, ON;
SET MY.EVENT text;   }
```

演算法 4.13 其他功能 按鈕觸發功能

4.3 流程設計

1.程式處理程序

〔1〕系統紀錄此目的地。

〔2〕尋找最近路徑並前往之。

〔3〕最近路徑走完遇到資訊站。

〔4〕資訊站依目的地告訴主角該走的路徑或者提供幾條路徑選擇。

〔5〕主角沿著被告知的路徑行走。

〔6〕走完又會遇到資訊站如此不斷重複。

〔7〕到達目的地。

2.從自由移動切換到自動導覽

〔1〕選擇目的地。

〔2〕尋找最近路徑。

〔3〕出發。

3.路徑類型

〔1〕單一路徑型 :一條路徑(走完就會偵測) 。

〔2〕串列路徑型 :多條序列路徑(需都走完才會偵測,也就是會忽略所有半路。

〔3〕混合路徑型 :單一路徑與多條序列路徑結合使用。

4.四種特殊狀態

〔1〕恢復:前往上一個目的地,中斷後的恢復。

〔2〕中斷:自動模式轉成自由移動。

〔3〕到達:結束自動模式,呼叫結束處理函式。

〔4〕繼續:到達後,選擇下一個要自動導覽的目的地。

5.路徑設計的原則

〔1〕路徑起頭必須在資訊站裡。

〔2〕路徑末端必須在資訊站或是目的站裡。

〔3〕資訊站大小約一人站立的範圍即可。

〔4〕目的點多的區域設的站數多，目的點少的區域設的站數少(建議)。

下面我們以流程圖來表示整個路徑程序的轉換情形。

自由模式流程

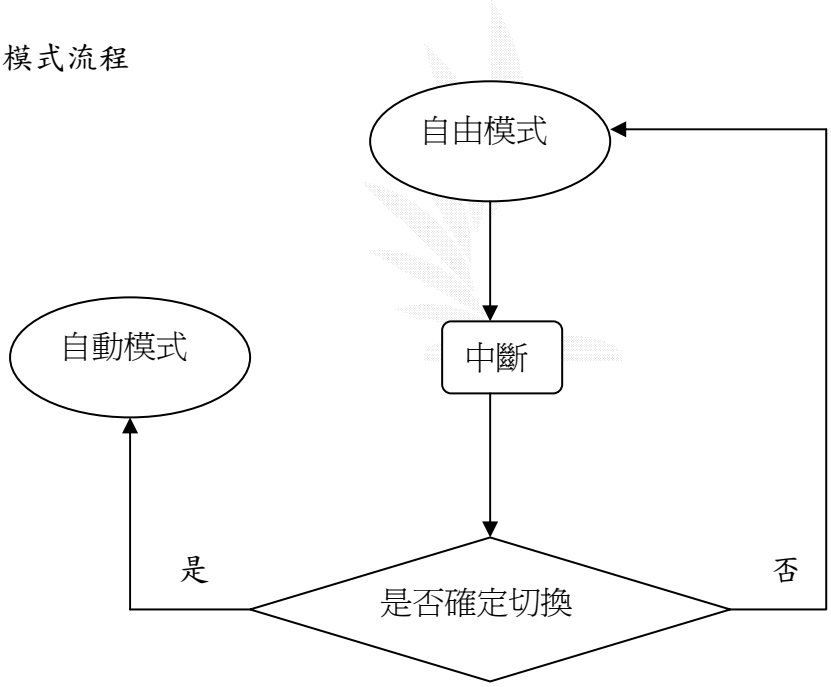


圖 4.30 自由模式流程

當我們一開始從系統進入時，會先進入自由模式。在自由模式下，人物可以自由的隨意走動,而當想要進行自動模式導覽時必須先按下預設控制鍵來進行切換。當我們再次確定要切換到自動模式後，程序就替換到自動模式下。

自動模式流程

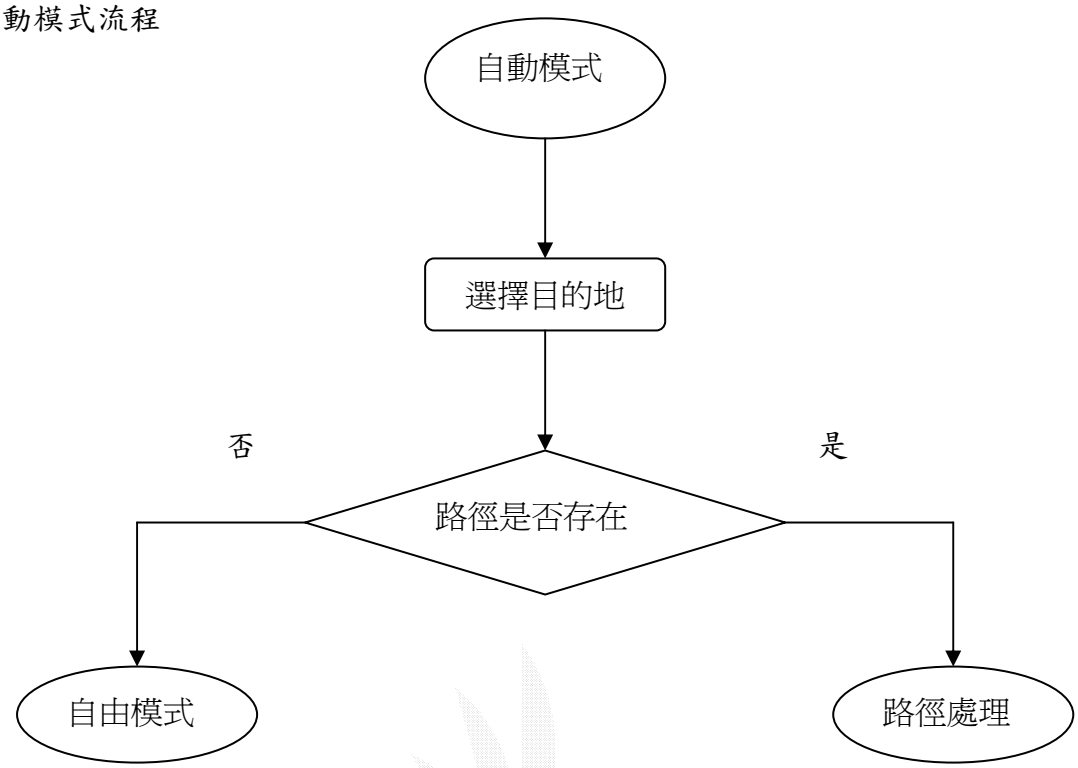


圖 4.31 自動模式流程

當我們從自由模式切換到自動模式時，會出現一系列的地點選單，而當我們按下某一個地點的按鈕元件時系統會偵測是否這個地點已經經過設計人員規劃好。如果尚未完全規劃完全會跳回到自由模式程序下，當存再路徑規劃時會開始最重要的路徑處理程序。

路徑處理流程

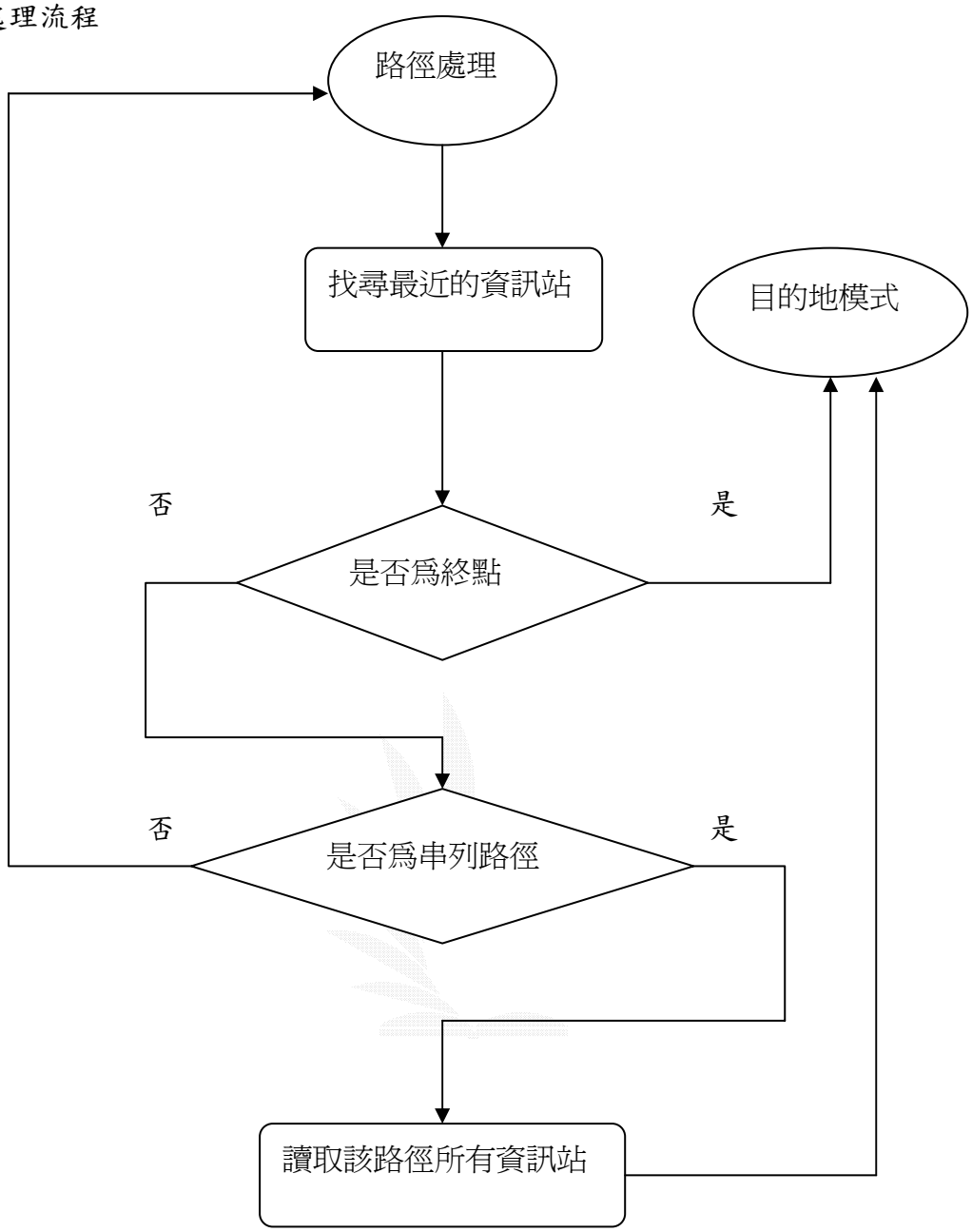


圖 4.32 路徑處理流程

進入路徑處理程序，便會直接先找尋相對應最近的一個資訊站點，並由系統判斷是否為最後終點，如果是終點的話就會進入目的地程序觀看簡介；不是終點的話，會進入判別是否為串列路徑，不為串列路徑的話會直接回到路徑處理再一次的搜尋最近的資訊站並判定是否為終點；若為串列的話，會一次讀取從此資訊站到最後目的地的所有資訊站點，一次走完並直接進行目的地程序。

串列路徑：一次走完一連串的資訊站，用於上樓梯前往教室時使用。

非串列路徑：一次走完一小段路徑，相當於兩個資訊站間的距離，用於一般平地到各樓門口使用。

目的地流程

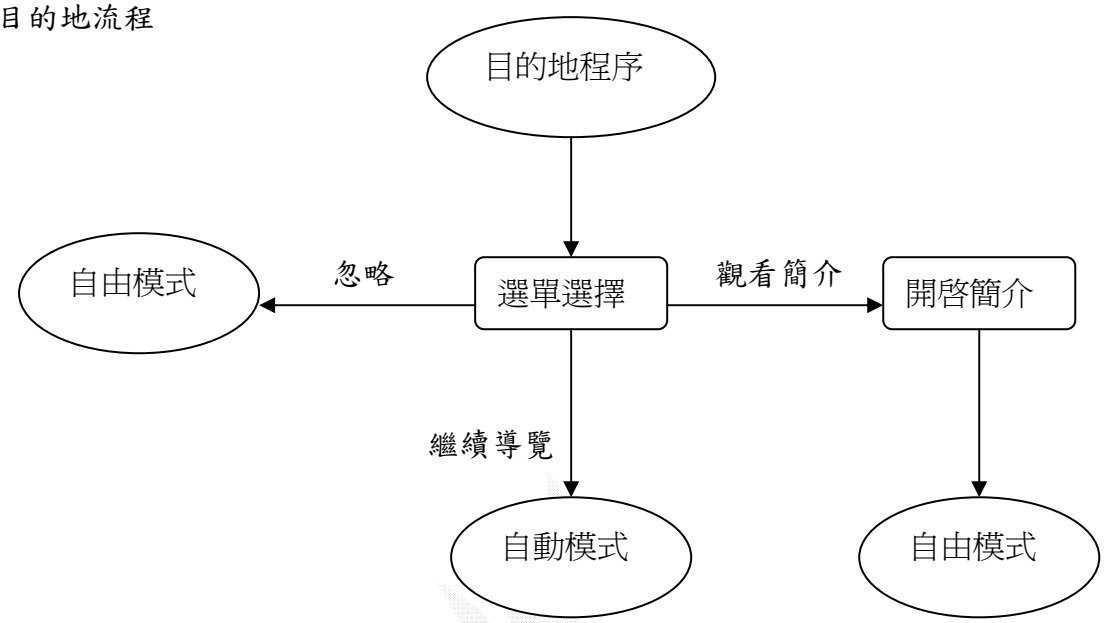
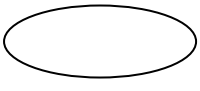
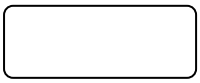


圖 4.33 目的地流程

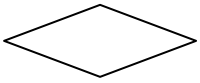
這是當我們程序跑到目的地程序的時候，所會看到的流程。人物觸碰到目的地的「資訊站」後會先開啟選單選擇畫面，在選單選擇中會出現三個選項—開啟簡介、忽略、繼續導覽。開啟簡介是把目的地的簡介檔開啟，讓使用者觀看簡介說明；等使用者瀏覽結束後，會自動跳到自由模式程序。忽略則是會直接跳回到自由模式。點選繼續導覽會回到自動模式下，在自動模式下可以繼續選擇下一個導覽目標。



用橢圓代表一個程序的替換



用方框代表一個動作



用菱形代表一個決策模式

第五章 系統介面說明

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。

人物的移動可以使用鍵盤上的上下左右鍵來作移動。

圖 5.1 是系統剛開始登入時的畫面，左手邊建築物是忠勤大樓，右手邊則是行政一館，這張圖是正門校門口取圖。在登入系統之後，當我們按下鍵盤控制鈕 S 便會出現圖 5.2 的系統開始畫面。當單擊導覽開始鈕後，會出現圖 5.3 的系統畫面。

圖 5.1 系統登入畫面一

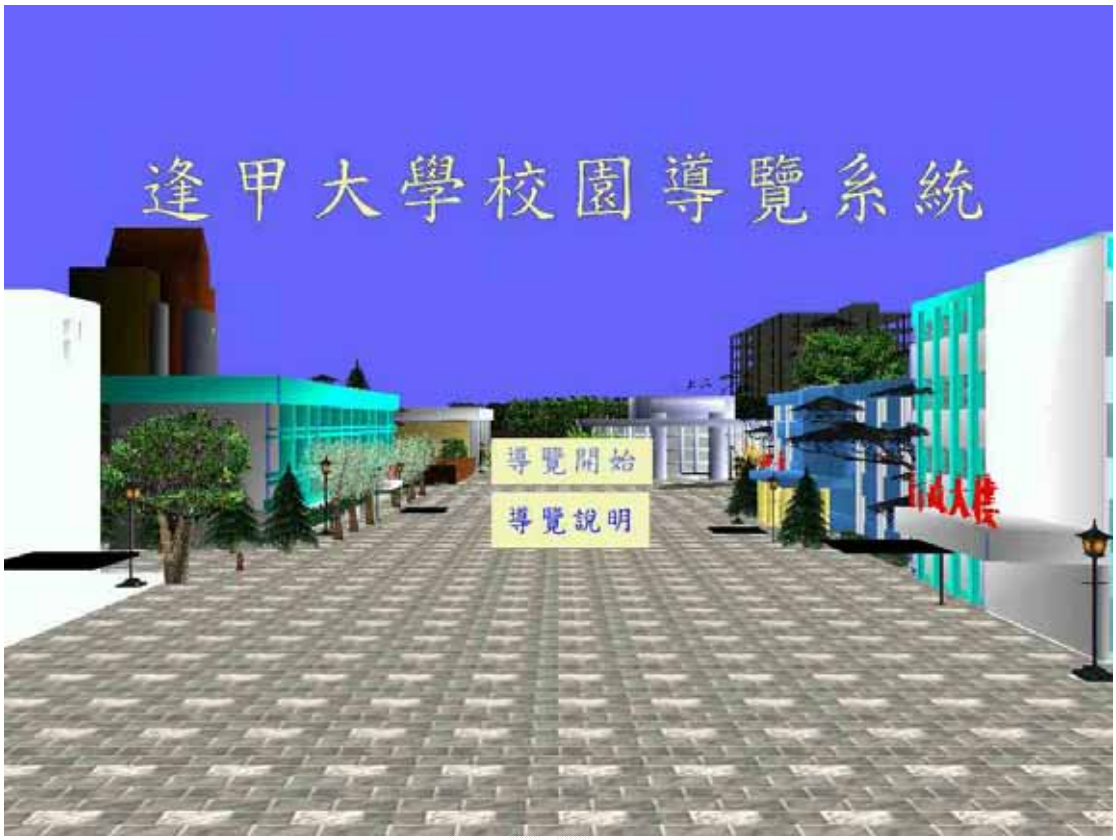


圖 5.2 系統登入畫面二

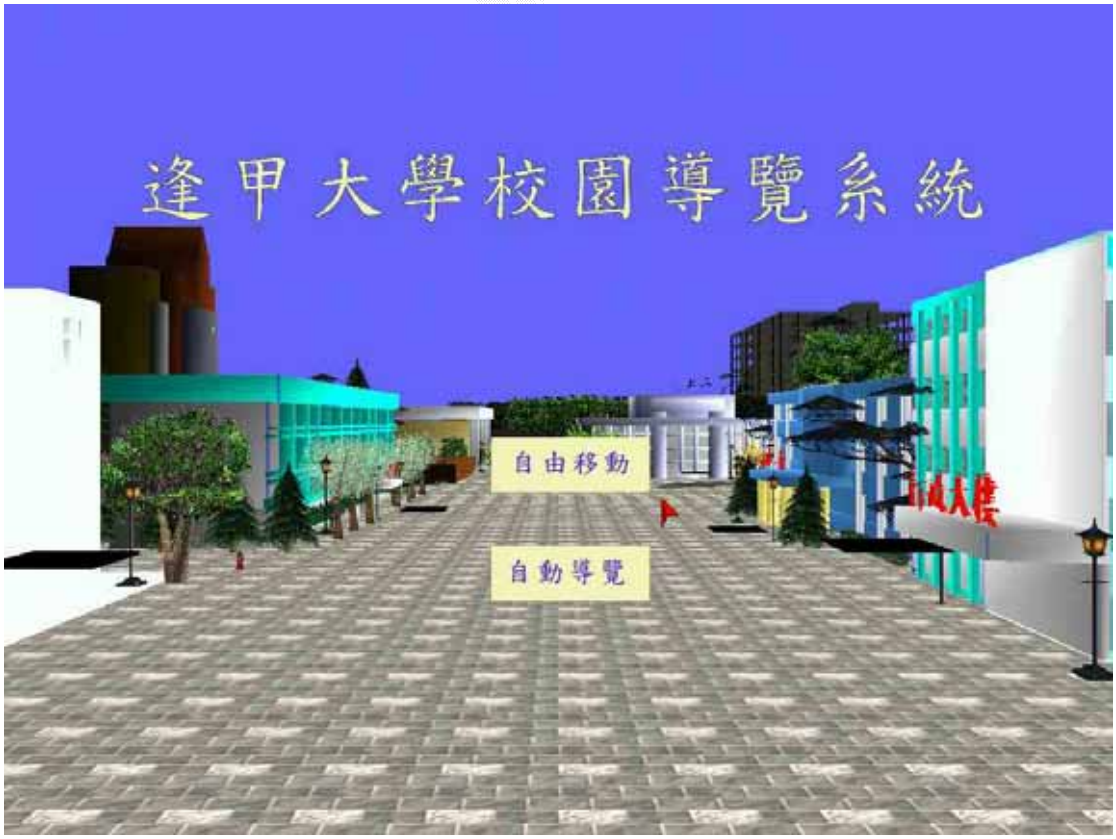


圖 5.3 擊點導覽開始鈕後的畫面

單擊自動導覽鈕後，會先進入人物選擇畫面圖 5.4，左邊是預設的男性角色，右邊則是預設的女性角色。圖 5.5 是用滑鼠點擊角色後出現的畫面。

圖 5.4 人物選擇畫面一

圖 5.5 人物選擇畫面二

確定角色後，擊點目的地選單鈕會陸續出現圖 5.6 到圖 5.8 的目的地選擇的操作畫面。最後，在圖 5.7 中擊點確定鈕，開始自動導覽。在導覽過程中可利用 F7 控制鈕切換畫面，如圖 5.9 到圖 5.12。

圖 5.6 目的地選單一

圖 5.7 目的地選單二

圖 5.8 目的地選單三

圖 5.9 導覽開始畫面 視角一

圖 5.10 導覽開始畫面 視角二

圖 5.11 導覽開始畫面 視角三

圖 5.12 導覽開始畫面 視角四

圖 5.13 到圖 5.20 是在本專題系統中所有大樓的實景圖。

圖 5.13　系統實景圖　人言大樓

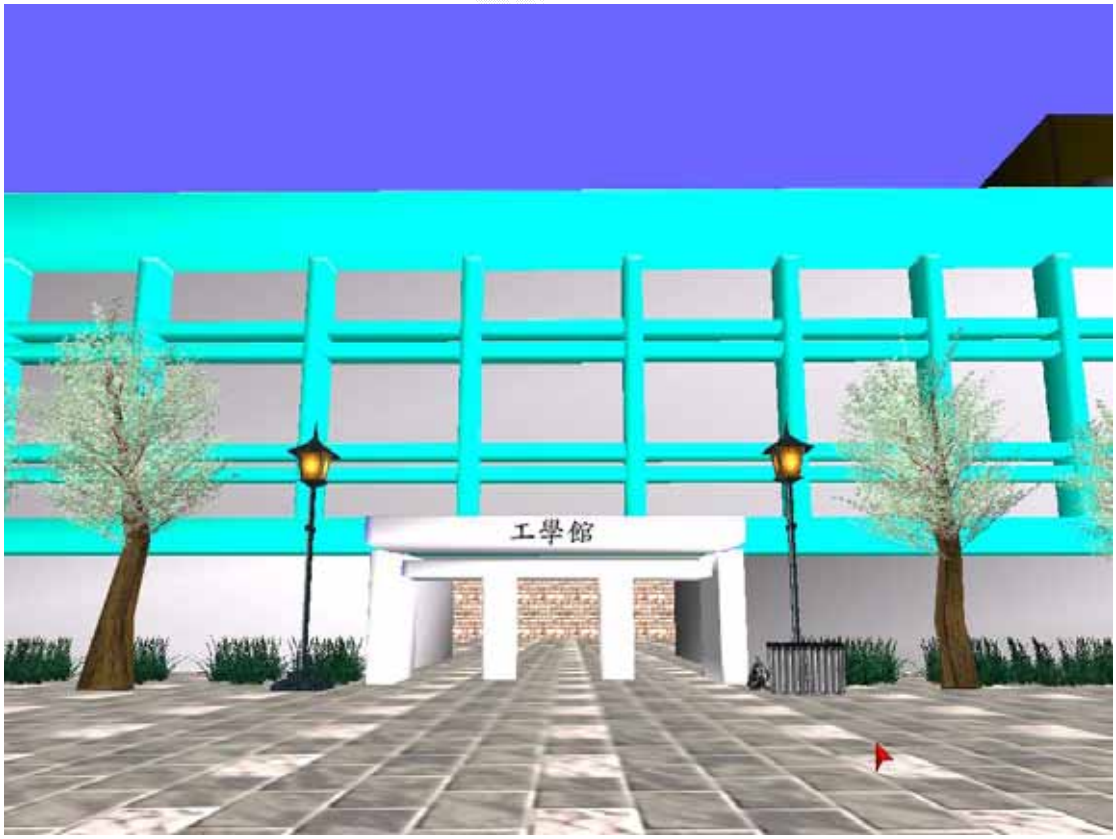


圖 5.14　系統實景圖　工學館

圖 5.15　系統實景圖　行政一館

圖 5.16　系統實景圖　行政二館

圖 5.17　系統實景圖　忠勤樓

圖 5.18　系統實景圖　邱逢甲紀念館

圖 5.19　系統實景圖　資訊電機大樓

圖 5.20　系統實景圖　逢甲圖書館

第六章 系統評估

6.1 綜合評估

整個校園導覽系統並沒有如我們所預期的開發出所有功能,目前達到的功能有多視角導覽功能、最短路徑搜尋、按鈕點選功能以及環境建設真實化。我們這次的專題大都把時間花費在建築物的設計上以及路徑行走的規劃,光是研究這兩項就能耗掉我們專題上的大多數時間。

另一方面，很可惜的是我們並沒有 100%發揮出 3D Game Studio 這套軟體的強大功能。顧名思義，既然他是遊戲發展軟體，那我們其實可以把整個逢甲大學發展成一各大型遊樂場，藉由結合數個小遊戲遊戲來導覽逢甲校園，也是一個不錯的方案。

本專題中，由於所呈現的成果是一套將近 20MB 檔案大小的程式檔，這對使用者來說要下載這麼大量的程式才能進行導覽實在有其不方便性,這點也是一項很大的缺憾。

6.2 功能評估

回頭看第四章的建築物設計方面，我們力求把模型的盡量的真實化。這一點我們是覺得有其必要性，一個校園導覽如果沒法盡量的貼實，那使用者在導覽上便沒有那一份真實感，這樣就失去了導覽的目的了。在本專題中，除了商學大樓有精確的設計教室隔間、樓梯以及一些細微的佈置外，其他如行政一館、行政二二館、忠勤樓、工學館、逢甲圖書館、丘逢甲紀念館與人言大樓和資訊電機大樓等大多數的大樓都是只能看到外觀，無法進行內部行走。

人物建置上，我們使用了原有範例裡頭所用的模型，這樣可以讓我們節省很多專題開發上的時效。

路徑方面，在最短路徑搜尋的功能上，實際做出來的功能並不是最短路徑搜尋，而是經過一步步下去規劃的路徑設定。設計過程中已經盡可能考慮到了所有可能性，包括如果使用者中途想變換目的地或是到了目的地後想再導覽其他地方，其重點就是在於變換路徑的過程中找到最近的資訊點去做路徑變換。唯一的遺憾是所有的路徑規畫都必須經過人工設定。

有幾段程式是參考範例程式改進完成的，視角變換和人物姿態變換程式，這兩段程式非常的冗長研究起來相當的花費時間。在一段時間的探討後決定在視角變換和人物姿態轉換程式上繼續使用 3D Game Studio 裡頭的範例程式。而在路徑規劃以及按鈕元件上則是經由手工設計出來所符合我們所需的功能。

第七章 結論

7.1 心得

當我們從三上開始做這個專題的時候，一開始的想法是要寫個 3D 遊戲，內容曾經想過賽車競賽類型、角色扮演類型，後來跟老師多次的討論之後決定以校園導覽系統作為我們的專題題目。決定了題目，我們開始想如何是實現它。首先，我們想到的是運用 DirectX 發展工具再搭配 3D 模型工具。3D 模型工具對我們來說不是問題，因為我們裡頭有一位專精美工的小組成員，比較令人擔心的是 DirectX 的學習。不過後來在某種因緣際會下，我們找到了一套遊戲引擎，由於它的強大功能能使得開發過程更加簡易，於是我們決定改採使用遊戲引擎開發。找到了合用的軟體那就要學著去使用它。我們從軟體的指引手冊一步步的學習如何操作，等到我們可以清楚的把它畫分成三大部分—MODEL、C-Script、環境本身，我們也就完全清楚了它的架構。到了美工階段，我們已經要開始整個環境的設計。整體環境的規劃是依照逢甲大學目前的狀況來建置的，所以我們參考了逢大的校園圖下去實作。

整個專題下來，我們學到包括 C-Script 與 3D Game Studio 的結合使用以及 3DS Max 的模型建造過程，最後是在導覽系統中我們可以呈現哪些觀點給使用者。我們很感謝謝信芳老師在整個專題上給我們的指導，尤其是在程序處理流程上給了相當多而且寶貴的看法，這些建議使得我們在專題上能夠順利的展現出成果。

7.2 未來展望

在未來，我們希望看到的是能夠把其他功能一併完成。在環境規劃上繼續完成所有建設，在路徑規劃上力求能達到完善。在功能建置上，尤其是考場導覽功能，我們覺得那是個不錯的想法，如果可以實現將是各不錯的成品。另一方面，由於我們所做的路徑規劃、建築物環境規劃、簡介說明是使用軟體設計沒辦法與資料庫搭配使用，只能在程式碼上做苦工，這點是最可惜的一點。

如果可以網頁化也就是將一切資料都放在網站上，當使用者點擊到瀏覽按鍵時，就開始下載大量圖形元件。這些圖形元件被一次下載之後，爾後在點擊時不須另外下載，就能以網頁形式的方式導覽校園。此外並可以結合網頁資料庫，把相關資料建成資料庫，如此一來可以不須變更到主程式便可以即時更新最新的資訊。

參考資料

書目資料

[1] 季延平譯，Gary B. Shelly、 Thomas J. Cashman、 Harry J. Rosenblatt 著，系統分析與設計，東華，Aug. 2002，pp.10 ~ 164。

[2] 黃敬仁，系統分析，碁峰，Feb. 2001，pp.50 ~ 150、pp.298 ~ 325。

[3] Johann C. Lotter著，Gamestudio C-Script Tutorial & Manual，pp.38 ~ 39、pp.103 ~ 107、pp.133 ~ 171。

[4] 位元文化著，Visual C++入門進階 從C++ 物件導向到視窗程式設計，文魁，May. 2000，pp.22-33 ~ 22-64。

[5] 劉登榮譯，Jason Kolb 著，DirectX 發展工具，松格，May.1997，pp.11 ~ 24。

[6] 徐正欣、黃科森著，深入淺出 Direct3D 應用程式設計，碁峯，Dec.1998，pp.37 ~ 72。

[7] 許子凡著，3ds max 5 完全手冊，文魁，Dec. 2002，pp.17 ~ 52、pp.57 ~ 59。

[8] Kwangwoo，嗯!我也會 3DS MAX-輕鬆成為 3D 動畫專家，博碩文化，Jan. 2002，pp.7 ~ 19、pp.31 ~ 42。

[9] 火星時代 王琦電腦動畫工作室著，3ds max 5 白金手冊上，北京科海電子出版社，Dec. 2002，pp.37 ~ 41。

網頁資料

3D Game Studio： http://www.conitec.net/a4info.htm

瑋特擬真科技： http://www.vimtek.com.tw/eonstudio4/about%20eonstudio4.htm

愛迪斯科技： http://www.axis3d.com.tw/index.jsp

逢甲校園導覽系統連結： http://www.fcu.edu.tw/life/campus/map1.htm

逢甲大學資訊工程大學系專題報告

校園導覽系統—以逢甲大學為例

莊家豪
吳江鎮
范獻巍

92