

摘要

雖然目前主流的個人電腦當初設計時是以「商務機器」為概念來設計的，但個人電腦上的遊戲存在歷史卻幾乎跟這台「商務機器」本身存在的時間一樣長，這顯示了使用者除了希望這台機器可以達到商務上的要求外，也要能提供娛樂。而在網路泡沫化的現在，遊戲產業擺脫早年的不良形象，在線上遊戲上大放異采，無懼經濟不景氣的衝擊，逆流而上，成為少數可以在網路上獲利的模式之一，尤有甚者，更有國家傾全國之力來發展。除少數智育類型的遊戲(如各種棋類、麻將及撲克等)外，目前主流的線上遊戲中絕大多數為角色扮演遊戲類型；而所謂的角色扮演遊戲(Role Playing Game, R.P.G.)，簡言之就是玩家扮演一個或多個的角色，在虛擬世界(不論是線上遊戲或單機遊戲)的大地上冒險、戰鬥、互動及成長的遊戲類型。而目前的角色扮演遊戲中，玩家在遊戲進行中所使用的地圖多由程式設計師以自行撰寫的地圖編輯器(Map Editor)依遊戲企畫所規劃的進行編輯，再以純文字的模式儲存在電腦中，然後在需要的時候以程式呼叫解譯使用，而在如此處理時，將會遇到下列的四個問題：

1. 遊戲地圖檔的內容可能只有地圖編輯器的程式設計師了解。但若地圖的複雜度增加或完成後的時間過久，將會出現連原設計師都不易理解當初所定義的內容。
2. 如果連原作者都可能無法輕易解釋遊戲地圖檔的內容，那其它的程式設計人員更不易瞭解，遑論處理了。
3. 如為已上線的線上遊戲，當需要在地圖檔中加入新的定義時，將需要修改程式碼來達到加入新功能的目的。這將有可能造成需要所有使用者斷線或新程式上線後因漣漪效應(Ripple Effect)而產生程式錯誤。

4. 將會需要為不同產品的地圖重寫或修改地圖引擎及地圖編輯器。

以上的種種問題其實都在於遊戲地圖檔並沒有作好資料與程式碼完全分離及其自我解釋性不夠而造成。而這些問題剛好就是 XML 所可以作到的。因此，在本文中將試以 XML 來構成遊戲地圖檔，並製作一個可以存取此地圖檔的介面，用以解決在角色扮演遊戲中的因地圖所產生的種種問題

關鍵字：XML、角色扮演遊戲、地圖。

Abstract

Except for few intellect games such as chess, mahjong, and poker games, role-play games in on-line games are the mainstream. As for the so-called RPG, briefly speaking, it is the game type that players play the role of one or more characters exploring, fighting, interacting or growing in the virtual world(no matter on-line games or single-machine games). In the present role-play games, the map which players use in the game is usually edited by the map editor which designed by programmers according to the game plan and stored in the computer in text format. Then, it is called to interpret by program when needed. We will meet four problems in the following when we handle it by such way:

- 1.The content of the game map file may be understood by the programmer who designs the map editor only. But if the complexity of maps increases or the time for finishing it is too long, the kind of content which the original programmer is not easy to understand will appear.
- 2.If the original producer may not interpret the content of the game map file easily, then other programmers will not be able to understand or even handle it.
- 3.We need to modify the program to add the new function when we add the new definition to the map file in on-line games. It will terminate the connection from all users or cause program errors because of ripple effect when the new program is on-line.
- 4.It is necessary to re-write or modify a brand-new map and map editors in the future for different products.

All these various problems come into being due to the reason that the game map file doesn't separate completely and interpret enough. XML can resolve all these problems.

Keyword: XML , RPG(Role Playing Game), Map

章 節 目 錄

中文摘要	I
英文摘要	III
章節目錄	IV
圖目錄	VI
表目錄	VII
第一章 導論	1
1.1 前言	1
1.2 研究動機	1
1.3 研究目的	2
1.4 章節安排	3
第二章 背景知識與相關研究	5
2.1 UML	5
2.2 XML	7
2.3 XML Schema	9
2.4 電腦遊戲的分類	10
第三章 遊戲及遊戲地圖設計方法介紹	14
3.1 遊戲功能性模組	14
3.1.1 遊戲程式設計中的其它的支援性程式	16
3.2 Microsoft DirectX	18
3.3 一般角色扮演遊戲中的地圖製作	19
第四章 以 XML 來設計角色扮演遊戲中的地圖檔	23
4.1 角色扮演遊戲地圖檔之 XML 表現方式	23
4.2 案例研究	27
4.3 與原作法之比較	28

4.4 使用 XML 可能會遭遇到的難題	32
第五章 結論及未來工作	33
參考文獻	36
附錄 A 地圖編輯程式執行畫面	38
附錄 B 示範程式執行畫面	43
附錄 C XML 地圖編輯類別	45

圖 目 錄

圖 3-1 遊戲功能模組圖	16
圖 3-2 Microsoft DirectX 架構圖	18
圖 3-3 地圖圖片檔	20
圖 3-4 範例地圖圖片檔	20
圖 4-1 RPG 地圖檔類別圖	24
圖 4-2 角色扮演遊戲地圖檔之 XML Schema 圖	26
圖 4-3 組合後的地圖檔	27
圖 4-4 來源圖片檔	28
圖 4-5 RPG 地圖檔之 XML Schema	28
圖 4-6 地圖原始檔	29
圖 4-7 地圖屬性 BMP 檔	30

表 目 錄

表 2-1 UML 模式圖的分類	6
表 3-1 圖片索引陣列	21
表 3-2 地形編號檔	21
表 3-3 組合後的地圖檔	21

第一章 導論

1.1 前言

在電腦遊戲中，角色扮演(Role Playing Game, R.P.G.)類型的遊戲從不需電腦的 T.R.P.G(Table Top R.P.G.，桌上型角色扮演遊戲或譯為紙筆式角色扮演遊戲)到純文字的 M.U.D.(Multiple User Dimension/ Dungeon/ Dialogue)，再隨著電腦本身的演進而不斷進化，不止出現過許多令人回味再三的大作，更是目前支持大多數線上遊戲獲利的主要遊戲類型。

而角色扮演遊戲主要就是讓使用者在程式設計人員及遊戲企劃所共同構築出來的虛擬世界中所控制的一到多個人物的成長、戰鬥、互動、冒險的故事為遊戲主軸。

而佈置虛擬世界最重要的，就是要有一個可以讓玩家角色可以在裡面活動及探險的大地，這個大地上可能有我們真實世界中常見的平原、森林、高山、河流、大海、沙漠、沼澤或人為建造的房舍、城市、宮殿及堡壘，更要有真實世界看不到的迷宮、地下城或人類原本無法進入的水底世界、火山陷阱及各種傳送點等等，這一切在電腦遊戲中，都是以地圖檔來展現。

1.2 研究動機

在角色扮演遊戲中，地圖的重要性不言可喻，但因為每個程式設

計師皆有自己的設計理念，故每個遊戲程式所用的地圖檔雖功用相近，但卻都完全不能互通，這個情形甚至在同一公司的同一組人員也可能因為程式發展的前後順序不同，而產生在兩個程式間的地圖檔不能互通

目前在一般電腦遊戲設計的時候，多是以 Microsoft 平台為主，也就多為使用 Microsoft DirectX SDK 為主要函式庫，但在地圖檔的使用上，卻多停留在程式設計師自行定義的純文字檔，再以自行設計的函式讀取後，以 Microsoft DirectX 中的 DirectDraw 來作繪出的處理。

這將會造成了四個問題：

1. 遊戲文字檔的內容可能只有地圖編輯器的程式設計師了解。但若地圖的複雜度增加或完成後的時間過久，將會出現連原設計師都不易理解當初所定義的內容。
2. 如果連原作者都可能無法輕易解釋遊戲文字檔的內容，那其它的程式設計人員更不易瞭解，遑論處理了。
3. 如為已上線的線上遊戲，當需要在地圖檔中加入新的定義時，將需要修改程式碼來達到加入新功能的目的。這將有可能造成需要所有使用者斷線或新程式上線後因漣漪效應(Ripple Effect)而產生程式錯誤。
4. 將會需要為不同遊戲產品的地圖檔重寫或修改地圖引擎及地圖編輯器。

也就是說，如此的作法雖然方便，但卻因為地圖檔與遊戲程式中的相關性太高，而造成不便移植而可再用性低落；或因為地圖檔不具自我解釋

性而難被原程式設計師以外的人理解。

因此，在本文中，將以 XML 來作為遊戲地圖檔的文件格式，並輔以一組介面程式，用以作到減低地圖檔與程式的相關性，並增加地圖的可攜性（也就是在不同程式中可以共用同一地圖檔）。

1.3 研究目的

本研究之主要目的有三：

1. 以 XML Schema 來表示遊戲地圖檔。
2. 減低遊戲地圖檔與程式間的相關性。
3. 透過以上的的步驟來提高遊戲主程式的可在用性及可讀性、簡化遊戲的設計過程，縮短遊戲程式開發的時間。

1.4 章節安排

本篇論文之架構如下：

第一章 導論：介紹研究動機及研究目的。

第二章 背景知識與相關技術：介紹相關技術(如 UML、XML 及 XML Schema)與電腦遊戲的分類。

第三章 遊戲及遊戲地圖設計方法：遊戲的功能性模組、遊戲設計中的其它輔助性程式、Microsoft DirectX 及目前一般的遊戲及遊戲中地圖的設計方法介紹。

第四章 以 XML Schema 設計遊戲中的地圖檔：應用 XML Schema 後，遊戲地圖檔的變化，並與使用 XML Schema 前做比較。

第五章 結論：本研究之結論及未來工作。

第二章 背景知識與相關研究

2.1 UML(Unified Modeling Language)

在系統的開發流程中，系統分析設計階段會影響到整個系統的架構。而自從物件導向技術問世以來，各種物件導向系統分析工具也陸續問世，但各家的分析工具有各家自己特有的表示方式，並沒有一個統一的表示方法，而產生許多困擾。

這個問題在 1997 年由 Booch, Rumbaugh 和 Jacobson 三位專家合作下，提出了統一模式語言(UML)而獲得解決，它整合了這三位先進之前各自所提之方法(OMT, OOSE 和 Booch)及其它好的方法。統一模式語言使用一致的圖示符號來表達系統分析設計階段時所產生的文件，使其成為一套普遍性的語言，用來描述整個系統架構和物件間的關係等[4]。統一模式語言利用標準的符號及語法，讓軟體工程師之間能有效的溝通，進而使得軟體開發正式成為一門工程學[5]。而統一模式語言也在 1997 年被 OMG/ISO 認定為業界標準語言，自此成為各家廠商所遵循的一個標準語言。

統一模式語言以視覺化的方式表示，將物件導向中的物件、繼承、類別等觀念結合起來，形成各種圖示符號。而這些圖示符號是參與系統開發相關人員的一個共同語言，幫助使用者、系統分析師、程式設計師及系統整合人員可透過一個標準的共同語言互相溝通了解。不需等到成品出來時才能驗證是否合乎需求，節省許多時間。

統一模式語言共分為案例圖、類別圖、物件圖、循序圖、狀態圖、活

動圖、元件圖、合作圖和部署圖九種圖示。而利用這幾種圖示，統一模式語言可廣泛的應用在各種複雜的軟體系統上[6]。

模 式 圖	使 用 時 機
Activity Diagram(動作圖)	代表一系列動作(Actions)的行為
Class Diagram(類別圖)	表示類別(Class)之間靜態關係的結構
Collaboration Diagram(合作圖)	代表某限制條件中，物件(Object)之間的連結與溝通
Component Diagram(元件圖)	代表一應用程式中實際的物件(Object)
Deployment Diagram(配置圖)	代表部屬在某一機器或之間的元件(Component)
Object Diagram(物件圖)	代表的是物件(Object)之間的關係，對應的是一簡化且無訊息(Message)的 Collaboration Diagram
Sequence Diagram(順序圖)	代表一群物件之間動態的溝通行為
State Chart Diagram(狀態圖)	代表某種狀態之下的類別(Class)
Use case Diagram(案例圖)	以使用者的觀點來代表某種情境

表 2-1 UML 模式圖的分類 (以字母排列)

許多論文均有提到利用統一模式語言來分析系統的實用性，如描述系統行為時，可利用使用案例(Use Case)來描述使用者之行為與系統間行為之差異，進而確定使用者的需求系統是否能符合[7]。而在建立領域模型時，可利用類別圖(Class Diagram)來描述各個類別之間的靜態關係，如屬性、操作方法等。而利用活動圖(Activity Diagram)可描述系統之工作流程[8]。而互動圖(Interaction Diagram)則是用來描述物件之間如何互動，其中包含循序圖(Sequence Diagram)和合作圖(Collaboration Diagram)。狀態圖則是用來描述物件所可能遭遇的各種狀態及各種狀態之間的觸發時機[9][10]。

自從物件導向觀念出現之後，傳統以功能為導向的系統分析模式所造成的系統日後改變困難的問題已獲得解決[11]。而統一模式語言可利用包

裏圖(Package Diagram)將類別以群組的方式來表現，進而表現出群組與群組之間的相依關係，使得系統能以更高階的形式來表示，較助於瞭解[12]。

2.2 XML (Extensible Markup Language)

XML 是近年來備受矚目的標記語言(Markup Language)，由 W3C (World Wide Web Consortium)所制定。

XML 是 SGML(Standard Generalized Markup Language)的子集合，將 SGML 簡化後，使其適合在網路環境下擴充使用，並補足 HTML 無法表達資料意義的缺點。XML 承繼 SGML 強大的擴充能力，讓使用者可以自行定義標籤、文件結構，以純文字的方式表示多種文件資料。

例如：

```
<?xml version="1.0" encoding="big5"?>
<通訊錄>
  <姓名>張三</姓名>
  <電話>04-22929405</電話>
  <地址>子虛市烏有路 2 段 168 號</地址>
</通訊錄>
```

XML 的特性可歸納成三點，文件自我描述、適合資訊互相流通及文件保值，茲分別說明如下：

- 文件自我描述：

一份 XML 文件可以分為兩類的文件，一類是 XML 的文件定義，而

這部份是非必要的，因此在某些應用中，我們會發現 XML 文件中並沒有提及它的文件定義。XML 文件定義是用來描述 XML 文件架構及標籤定義，因此我們可以使用它來定義我們自己的 XML 文件，如前面的 XML 文件就是一例。

- 適合資訊互相流通：

在網路環境是由眾多終端使用者與伺服器所交織成複雜的網際網路，彼此間作業環境及異質性都相當高，例如網路環境、作業系統、作業平台、應用軟體可能都有很大的差異。但是現在只要資訊系統內有 XML 腳解析器(Parser)，最常見的就是瀏覽器(Browser，如 Microsoft Internet Explorer)，想要輕鬆直接地在分散式的網路環境下交換資料，透過 XML 來達成是非常容易的事情。

- 文件保值：

XML 文件能夠利用一種 XML 格式話的語言操作 XSLT(XSL Transformations) 來把文件轉換成另一種格式的語言。這無疑是對目前正在推動的電子公文提出了一個完美的解決方案，因為目前一般的紙本公文有保存不易，較浪費紙張的缺點。而若一般文件以 XML 的方式保存，就不用擔心未來的應用軟體不支援目前的檔案格式，只要經由 XSLT 轉換成新格式即可。

為了讓 XML 解析器可以正確解析，XML 文件必需是格式正確(Well-formatted)的，要符合格式正確，必需要滿足以下條件：

- 所有元素都要正確的關閉。

- 標籤之間不得交叉。
- 所有的屬性值都必需用引號。

所有的 XML 文件除了需格式正確外，其標籤定義及文件結構也應符合法則，否則程式會不清楚標籤內容的資料形態，而且 XML 文件本身也可能會有語意上的錯誤發生，而這套法則就稱為 XML 文件定義。另一個觀點是從電子商務環境下文件交換的自動化來看，各產業需訂出全球通行的文件定義，讓有意進入這個交易環境的成員能夠瞭解彼此的資料意義，才能夠順利與交易伙伴交換資料。能夠以 DTD 或 XML Schema 驗證的 XML 文件，就可以稱為 Valid XML 文件。

2.3 XML Schema

XML schema 是用來定義 XML 文件的語彙及文件結構的描述語言，本身就是一份 XML 文件，早期 W3C 是延用 SGML 的 DTD (Document Type Definition) 來定義 XML 文件，也廣為瀏覽器及應用程式採用，軟體支援度很高，但是由於 DTD 語法特殊、自成一格，與 XML 語法不同，因此可讀性較低，而且無法支援命名空間 (Namespace)，僅支援簡單的資料型態 (只有字串)，使得 XML 文件的使用及表達受到限制。舉個簡單例子，我們不能用 DTD 來定義某元素的資料型態為數字，只能定義成字串型態；我們也不能用 DTD 來定義某元素個數最少要有一個，最多只能有四個，只能定義其個數為 0、1 或是無窮多個。基於以上種種原因，W3C 又制訂了以 XML 本身為語法的 XML Schema 來替代 DTD，並擴充 XML 文件的能力。

XML Schema 借重物件導向程式資料模擬的技術，使開發人員可以利用語言中暨有的元素屬性的繼承關係來指定新的元素，提高重複使用度，也降低了開發的成本，這使 XML Schema 可因不同的專業領域的作業需求，來做機動性的調整和變更整體的資料結構。

2.4 電腦遊戲的分類

電腦在日新月異地鑄造著自己的輝煌的同時，作為其產業的重要分支—電腦遊戲產業也隨之獲得了令人眩目的發展。電腦遊戲的發展過程在很大程度上體現著電腦軟硬體的更新速度。

從上古洪荒時代到未來世界的星際旅行，從俠骨柔腸的江湖兒女到血肉橫飛的殺戮戰場，從經營上的精打細算到把握全局的宏觀調控，從劍拔弩張的冷兵器對決到高科技的重裝機甲，從神秘莫測的魔幻世界到光怪陸離的外星文明等等 電腦遊戲在不斷地隨著人類大腦的開闊思維而開拓著其自身的疆域。

所以，電腦遊戲的發展，也可以說是人類本身形象思維的發展，是人類本身生活及發展的影射和延續。它所向世人展示的也就是人類自己的夢想。近幾年，電腦遊戲不斷推陳出新，故在本文中針對較常見的遊戲大致作了一個粗淺的分類如下[15]：

- 動作類 (ACT)：
 - 特色：常需要使用者按下各種按鈕多次來控制遊戲中角色的動作，格鬥或運動類型的遊戲多屬此類。
 - 著名代表作：快打漩風系列、NBA 2003、FIFA 2002 等。

- 冒險類 (AVG) :
 - 特色：遊戲中有一道道相互勾連的謎題及情節等著讓使用者解開，具有較強的邏輯推理性質。
 - 著名代表作：古墓奇兵、惡靈古堡等。

- 策略類 (SLG) :
 - 特色：使用者在遊戲中可以控制小到一個部隊，大到一個國家的行政、外交、軍事行動等等以消滅或征服對手。
 - 著名代表作：沙丘魔堡系列、三國誌系列、世紀帝國系列等。

- 模擬類 :
 - 特色：以操縱現實中或虛擬存在的載具(如飛機、坦克、車輛、船艦等)為目的的遊戲。
 - 著名代表作：長弓阿帕契、極速快感、一路狂飆等。

- 角色扮演類 (RPG) :
 - 特色：以人物在虛擬世界中的成長、戰鬥、互動、冒險為遊戲主軸，目前的線上遊戲也大多是此類遊戲。
 - 著名代表作：創世紀系列、魔法門系列及線上遊戲的天堂、傳奇石器時代等。

- 射擊類 (STG) :

- 特色：以各種武器射擊敵人以進行遊戲，早期為以 2D 畫面為主流，現多為 3D，近年來大行其道的第一人稱射擊(FPS)也可歸為這一類。
- 著名代表作：小蜜蜂、雷電、毀滅戰士、重返德軍總部等。
- 養成類：
 - 特色：多是日系遊戲，以安排角色之食衣住行為為遊戲內容，使用者多為女性，通常為多結局之遊戲。
 - 著名代表作：美少女夢工廠系列等。
- 經營建設類：
 - 特色：在遊戲中經營一家公司或建設一個城市，以商業或建設手段達到遊戲中的目標。
 - 著名代表作：金錢帝國、模擬城市系列等。
- 智育類 (PZG)：
 - 特色：規則簡單但變化多，需要使用者的推理及思考的遊戲類型，也有許多是在現實生活中可以見到，但移植到電腦上成為遊戲。
 - 著名代表作：踩地雷、俄羅斯方塊、各種棋類遊戲等。

以上的分類法可以說是已經將包含目前大部份的遊戲的類別，但由於

人類創意的無限及電腦技術的不斷進步，許多無法歸類於以上其中之一的遊戲也不斷出現中，如結合動作與角色扮演的動作角色扮演(ARPG, Action Role Playing Game)或在網咖引領風潮且歷久不衰的第一人稱射擊遊戲(FPS, First Person Shooting game) 及幾個原為純戰略類型(如魔獸爭霸 War Craft 系列)的遊戲也紛紛在遊戲中加入了角色扮演的概念，這些都是以往未曾出現過且難以歸類的遊戲類型。

第三章 遊戲及遊戲地圖設計方法介紹

3.1 遊戲的功能性模組

一般而言，在遊戲程式設計時，依據每個設計師的經驗的多寡，皆會對遊戲程式整體作一些功能上簡單的區分，以避免程式混雜不清。

但這個工作，常因為每位設計師能力及經驗的不同，而有不同的分法。在本文裡，將依遊戲程式中不同任務，依一般常見的分功能，分為如下幾個部份：

- 主程式：整個遊戲的主要處理中心。往來訊息的交會、協調及邏輯處理單位。
- 使用者介面：使用者所接觸到的介面(User Interface)，負責整個遊戲與使用者互動的部份，包含聲音及影像的輸出及鍵盤、滑鼠甚至搖桿等輸入部份。
- 劇本引擎：在遊戲進行中，用以讀入預設劇本用的程式，確保遊戲如設計者所預想的一般進行，在冒險類或角色扮演類型遊戲中較為常用，在本文範例中並未使用。
- 地圖引擎：在遊戲進行中，用以讀入地圖用的程式，並不包含顯示的部份，也就是可以依設計者要求的以 2 維(2-Dimension)或 3 維(3-Dimension)方式顯示在使用者介面上，在本文範例中並未使用。

- 圖形引擎：負責在顯示使用者介面之上之圖形的處理，通常還可以分為秀圖及秀字兩種程式，同時也包含圖檔讀入的功能。
- 聲音引擎：負責在使用者介面上各種聲音的處理，通常還可分為音樂及音效的兩種程式，同時也包含音樂、音效讀入的功能。
- 檔案處理：位於硬體之上，負責各引擎與資料庫、音樂、音效、圖形、劇本及地圖的低階輸出入程式。
- 通訊程式：負責遊戲中主機端對用戶端間或用戶端對用戶端間的網路通訊及封包處理、加解密之工作。
- 相關資料庫：遊戲程式初始化及進行中資料的儲放位置，通常可分為劇本、進度、圖形、物品(如武器、防具、道具等)、音樂及音效等資料庫分開儲放。

以上的各部份程式的相關位置如圖 1 所示：

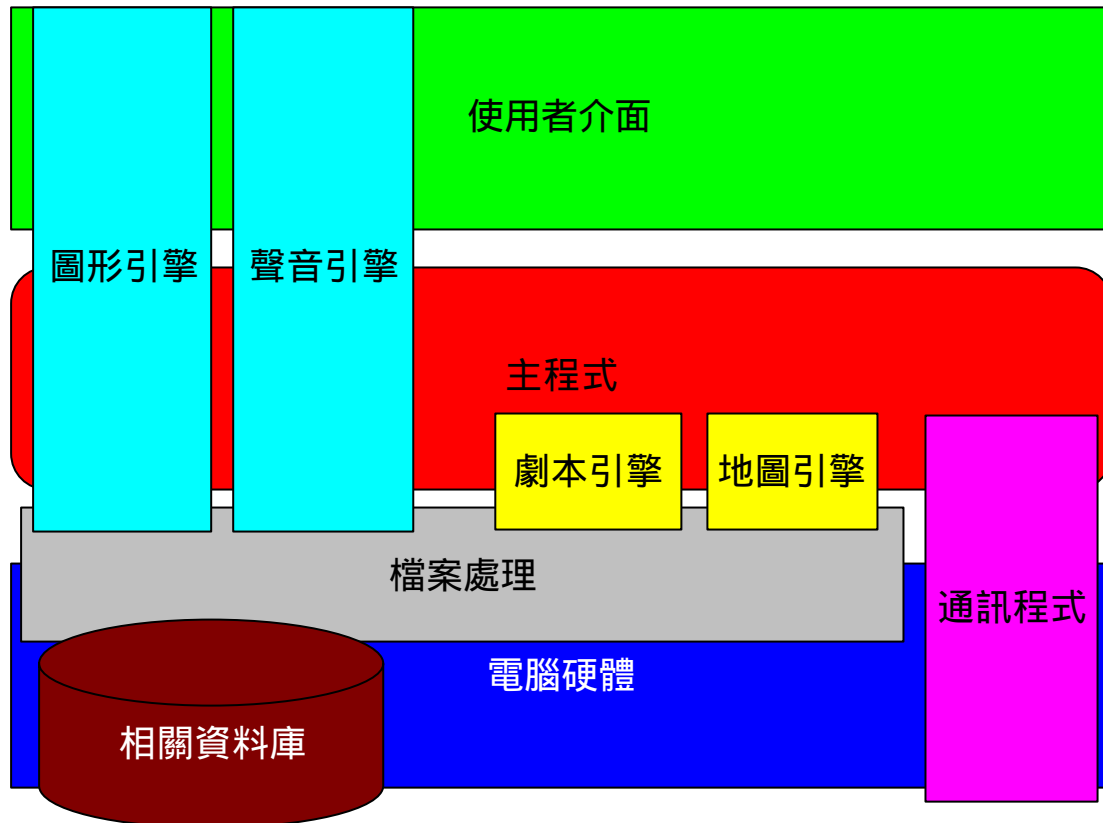


圖 3-1 遊戲功能模組圖

除了因設計師的能力不同會對上圖的分割法有所不同以外，另外就是因為各類遊戲的需求並不一致，上圖的各部份所使用的比重也不一定相同。

3.1.1 遊戲程式設計中其它的支援性程式

在遊戲的設計過程中，除了最後呈現在使用者面前的遊戲本身外，事實上還有一些程式因為與遊戲內容沒有直接關係而未公開，但這些程式在設計過程中也是非常重要的，茲分別說明如下：

1. 劇本編輯器：用以編寫整個遊戲故事流程進行的程式，目的在與簡化遊戲中故事劇情的編排，使得易於編寫出符合遊戲企劃需求

故事的流程；亦或在發現原本排定劇情有問題的時候易於改寫的程式。多用於冒險或角色扮演類遊戲。

2. 地圖編輯器：用於遊戲中角色人物所行走的大地或各種地形(如人為建物的內部地形)之編寫，通常會與劇本編輯器一起使用，以便讓使用者在某特定地點觸發特定事件。通常用於角色扮演遊戲較多。
3. 音樂/音效處理程式：遊戲進行中所使用的音樂及音效編輯程式，現多採用現成的商業軟體。
4. 圖片/3D 影像處理程式：編輯遊戲中所使用的圖片(如地形或角色等)所使用的程式，依圖片的種類不同而可分為 2D/3D 影像處理程式，現多採用現成的商業軟體，但所產出的成品多經轉換程式處理過才上市。
5. 轉換程式：早期多為處理程式二進位碼內容，避免遭反向工程破壞的程式；近期多作為將遊戲中的圖片或影像轉換為一般使用者無法直接觀看的格式之用。也有音樂或音效類型的轉換程式。
6. 防盜程式模組：防止盜版之用，通常各公司皆有不同的模組，在遊戲出貨前加入到程式中，也有設計師是直接寫在程式中的。

以上的程式功用皆因輔助遊戲程式的設計而出現，而在各種類型的遊戲中因為各遊戲偏重的成份不一，或因各種原因而並未獨立成個別程式的情形也所在多有。當然也有程式設計師應用市面上一般的商用軟體來處理，自然就不需要其中的部份程式了。

3.2 Microsoft DirectX

Microsoft DirectX 是一組低階的應用程式界面(API)，供製作遊戲和其它高效能多媒體使用。可以透過不依賴裝置的方法，提供對硬體的低階存取權。支援二維(2-D)及三維(3-D)的圖形、聲音效果及音樂、輸入設備、及線上遊戲這類的網路遊戲[14]。

Microsoft DirectX 包含了DirectX Graphics、DirectX Audio、DirectInput、DirectPlay、DirectShow 及 DirectSetup 六個元件。

DirectX 的架構是組合式驅動程式(Team driver);是將硬體抽象層(the Hardware Abstraction Layer)跟硬體模擬層(the Hardware Emulation Layer)組合起來，以達到 DirectX 所提出的要求。

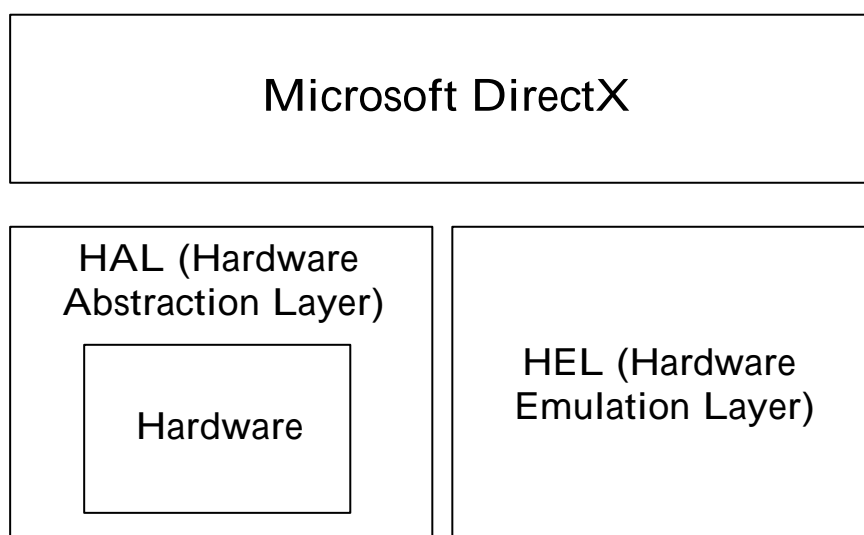


圖 3-2 Microsoft DirectX 架構圖

目前 Microsoft DirectX 最新的版本為 9.0 版。

3.3 一般角色扮演遊戲中的地圖製作

地圖的製作，可以說是 RPG 中最浩大的工程，不管是包括遊戲中世界地圖中的山、海、森林、城堡中的街道、房屋、洞穴中的曲折地道、高塔中的詭異迷宮，均是屬於 RPG 中的地圖部份[28]。

而在遊戲中，所有的大地圖，也都是由一片一片的地形地物的小圖所組成的，每一張小圖，所代表的意義都不相同，再視要組成那種地形，將相關的圖片放在一起，就可以組成遊戲中的地圖了。

一般在遊戲中，為了方便遊戲讀取，都是將所有的地形圖片檔放在同一個檔案中，如下所示：



圖 3-3 地圖圖片檔

將地圖圖片檔全部讀取後，以固定的像素大小將圖片切割成一片片放入陣列中，並以陣列索引取用，組合成所需要的地圖。

例如地圖圖片檔內容如下：

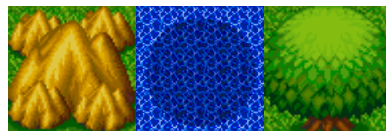


圖 3-4 範例地圖圖片檔

讀入後，將其依同等大小切割後，依順序放入陣列中：


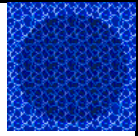

陣列索引	1	2	3
圖 片			

表 3-1 圖片索引陣列

當要組合成大地圖時，以二維陣列將上表中的陣列索引放入即可

假設大地圖的地形資料所組成的二維陣列內容為：

2	2	1	2	2
2	1	3	3	2
1	3	3	3	1

表 3-2 地形編號檔

組合以上的資料，將會變成如下所示：



表 3-3 組合後的地圖檔

以上的工作僅是以拼圖的方式來完成一張大的圖檔，要放入遊戲執行

前還要針對地圖上的每一區塊設定如是否可行走、是否要觸發特殊事件等等的屬性進行設定。以防遊戲中的角色可以通過原本在如海洋或城牆等在現實生活中不能通過的地區。

屬性的設定方面，一般而言是以事件編輯器來處理，但視情況及設計師偏好的不同，有的遊戲設計師會全部擺入地圖檔中，有的只擺入部份(如只設定可否行走)，當然也有完全將屬性及地圖檔分開來處理的。

而大部份的地圖編輯器功能都一定會有讀入文字設定檔(也就是如表二的地形編號檔)，以方便作初始設定。而地圖編輯器的輸出檔案多為規則性的二進位文字檔，此檔案的內容僅有遊戲程式可以讀取，而每個遊戲程式之間的地圖檔也可能僅因一點小小的差異而不能互通。

以上的範例中是以 2 維的上視圖形作示範，同樣的處理可以應用在橫向捲動 3 維俯瞰或是現在大多數網路角色扮演遊戲所採用的 3 維斜 45 度角圖都可以應用。

第四章 以 XML 設計角色扮演遊戲中的地圖檔

在本文中，由於是以 RPG 地圖檔的高互通性及低程式相關性作為論述要點，因此並未涉及檔案內容的安全性或保密性以及效率等議題，並設定屬性區塊大小與每一圖片大小相等，以便進行處理。而在事件處理上，則以在地圖檔中編入事件代碼(Event ID)的方式提供遊戲程式連結事件資料庫(Event Database)處理後續事宜。

4.1 角色扮演遊戲地圖檔之 XML 表現方式

由第三章的例子可以得知，角色扮演遊戲中的地圖檔應該有的要素如下：

1. 大地圖的相關資料：應包含 此地圖的編號(因為可能是整張地圖中的一塊)、所使用的圖片檔、大地圖的長及寬，還有每一圖片的長及寬。
2. 圖片資料：應包含 所使用的圖片編號(在圖片檔中)以及在此圖片中的移動難度(以一數值代表，如"0"代表無法通行，而"100"代表可以暢行無阻)，另外還需要有連結事件用的事件編號，以便連結事件進行觸發。
3. 建築物資料：應包含 建物本身的圖片位置、在圖上所展示的實際大小、在地圖檔上的位置以及連結事件用的事件編號。

將以上的概念以類別圖來表示：

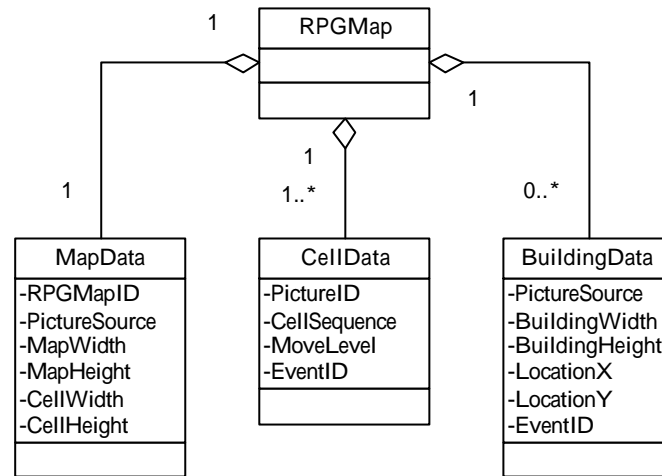


圖 4-1 RPG 地圖檔類別圖

以上的類別圖說明如下：

RPGMap 為 MapData、CellData 及 Building 的聚合。

MapData 為整張地圖的相關資料為必要元素，每張地圖檔僅有一份，包含下列資料：

1. RPGMapID：代表此地圖的編號，可以使用此功能來拼湊出一張更大的地圖，也可以代表地圖在橫向捲動遊戲中出現的順序。
2. PictureSource：為地圖中元素所組成的圖檔位置，以字串的方式指出該元素圖檔在那台電腦或那個目錄中，以何檔名儲存。
3. MapWidth 及 MapHeight 為整張地圖的尺寸。
4. CellWidth 及 CellHeight 為構成本地圖的元素尺寸。

CellData 為地圖檔中，每一個元素的資料內容，其數目應為 MapData 中的 $(\text{MapWidth} / \text{CellWidth})$ 乘上 $(\text{MapHeight} / \text{CellHeight})$ 的積，如非這個數字，代表此份文件不正確。其內容說明如下

1. PictureID 為在 MapData 中 PictureSource 所指出的圖片檔中，圖片元素在圖片檔中的順序。
2. CellSequence 為本元素在地圖中的順序，為一橫向排列，當其號碼大於 $(\text{MapWidth} / \text{CellWidth})$ 時，將自動向下換行，並由下行的第一個位置開始排列。
3. MoveLevel 表示遊戲中的角色在此元素上的移動難度，預設值為 100，小於此數代表移動難度較高，大於此數表可以快速移動，0 為無法通過的地形。
4. Event ID 為玩家角色移動至此區塊時所觸發的事件代碼，用於連結至遊戲事件資料庫，預設值為 0 代表不觸發。

而為了表現在地圖上會出現的人為或非人為建築物，使用 BuildingData 來記錄地圖上的建築物資料，由於一張地圖上可能有也可能沒有建築物，因此地圖檔中未必會使用到此 Element。將其內部資料說明如下：

1. PictureSource 為存放建築物外的圖片所在位置，以字串的方式指出該元素圖檔在那台電腦或那個目錄中，以何檔名儲存。
2. BuildingWidth 及 BuildingHeight 為建築物圖片在地圖上的顯示時的尺寸。

3. LocationX 及 LocationY 為建築物圖片在地圖上的位置。
4. Event ID 為玩家角色移動至此區塊時所觸發的事件代碼，用於連結至遊戲事件資料庫，預設值為 0 代表不觸發。

轉換成 XML Schema 圖如下：

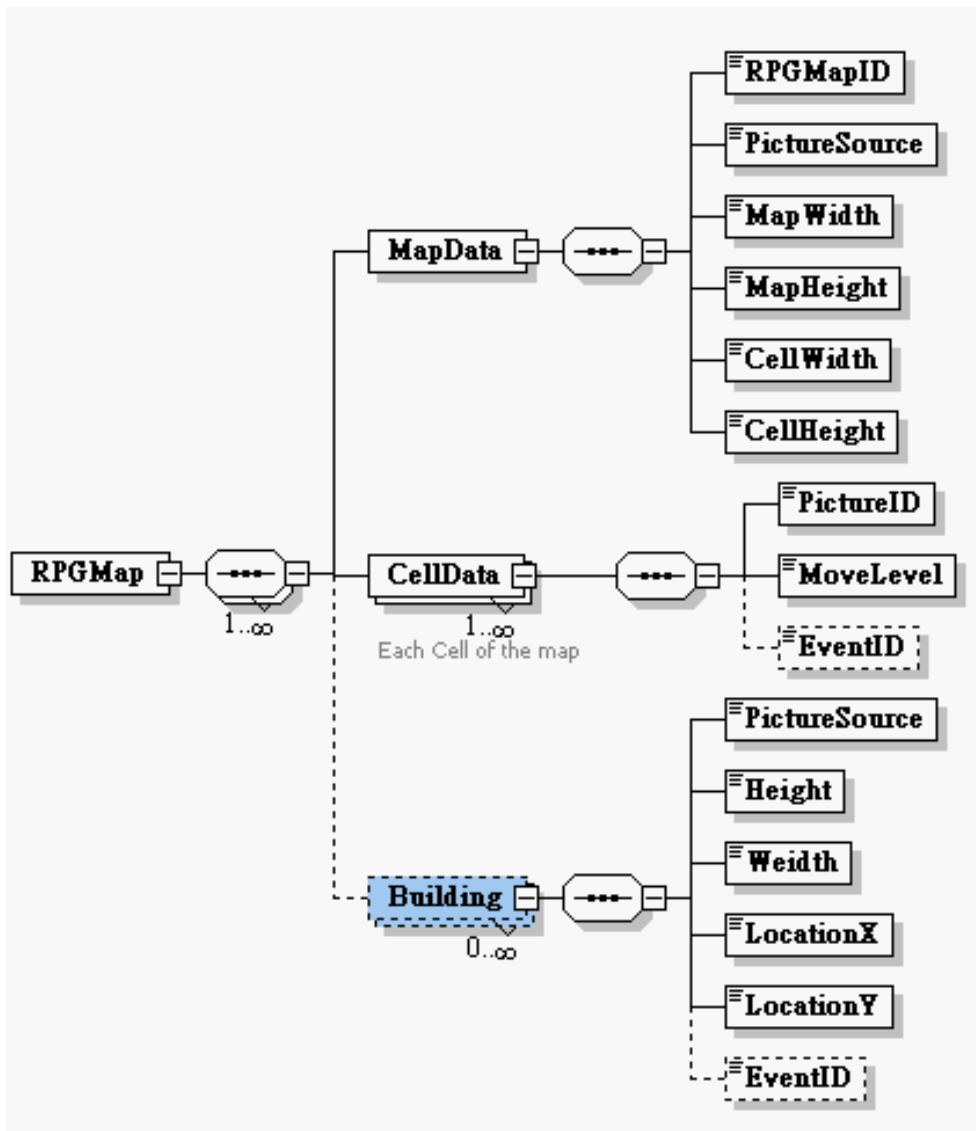


圖 4-2 角色扮演遊戲地圖檔之 XML Schema 圖

4.2 案例研究

假設現在需要一地圖檔如下：

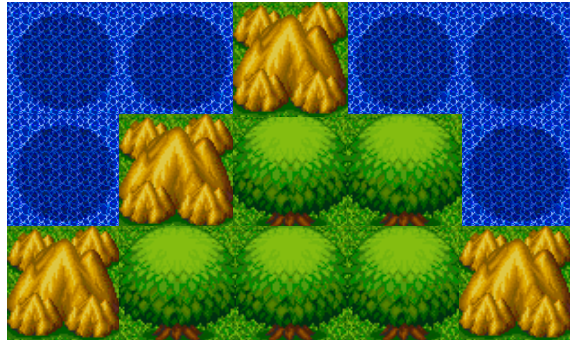


圖 4-3 組合後的地圖檔

並設定地形通過難度為：

1. 海洋 0(禁止通行)。
2. 山丘 50(難以通行)。
3. 森林 80(輕易通行)。

在此地圖檔中不觸發任何事件。

原始圖片檔的路徑位於本機電腦中的 ” /images ” ，其檔名為 ” map1.gif ” 。

內容如下：

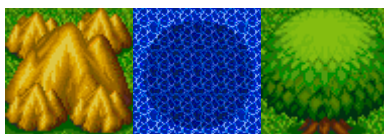


圖 4-4 來源圖片檔

以上的每一張的圖片為 32X32 像素，而以上的地圖大小為 3X5 格，故可以得知整張地圖的大小為 96X160 像素。

本張地圖並未放入任何建築物，如以本文所規劃的 XML Schema 來表示：

```
<?xml version="1.0" encoding="Big5"?>
<RPGMap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\My Docs\report3\RPGMAP.xsd">
  <MapData>
    <RPGMapID>1</RPGMapID>
    <PictureSource>images/map1.gif</PictureSource>
    <MapWidth>160</MapWidth>
    <MapHeight>96</MapHeight>
    <CellWidth>32</CellWidth>
    <CellHeight>32</CellHeight>
  </MapData>
  <CellData>
    <PictureID>2</PictureID>
    <MoveLevel>0</MoveLevel>
    <CellSequence>1</CellSequence>
  </CellData>
  <CellData>
    <PictureID>2</PictureID>
    <MoveLevel>0</MoveLevel>
    <CellSequence>2</CellSequence>
  </CellData>
    .....(中略)
  <CellData>
    <PictureID>1</PictureID>
    <MoveLevel>50</MoveLevel>
    <CellSequence>16</CellSequence>
  </CellData>
</RPGMap>
```

圖 4-5 RPG 地圖檔之 XML Schema

4.3 與原作法之比較

而在地圖編輯器讀入以上的檔案，並作好相關的修改後，其將產生兩個檔案，一為二進位地圖檔，包含了地圖上所使用的圖片資訊、相關位置及地圖的整體設定，並以二進位的方式儲存，以減少檔案體積及加快讀取速度，故無法為人類所辨識讀取；另一為如圖 9 的地圖屬性 BMP 檔，通常是以地圖尺寸等比例縮小的 BMP 圖檔格式來儲存，並以純色塊來代表如可否通行或觸發何種事件，以圖 4-6 為例：

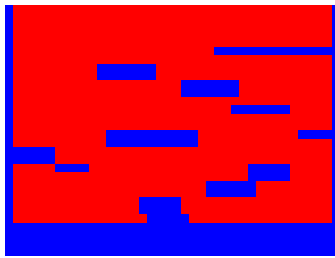


圖 4-7 地圖屬性 BMP 檔

其中紅色的部份代表可以通行的區，藍色代表禁止通過。如果要表現角色在不同地形上不同的移動能力，就必需使用更多的顏色，並且需要修改程式以作相對應的處理。

以上的作法在確實有相當優良的效率及表現，但其缺點卻更顯而易見，茲條列如下：

1. 初始文件檔編寫不易，如何將企畫人員腦中或紙上的地圖轉成地圖編輯器可以讀取的格式是一個煩雜的工程，而各遊戲中不同的格式更加重了其中的困難度。
2. 轉出成二進位檔可說是追求效率上的必然，但無法以人力閱讀及修改可說是一大致命傷。

3. BMP 屬性檔中各種顏色所代表的意義及的處理方法需寫在程式中，如有新的屬性出現時，則需針對加入的顏色設計新的處理方法並重新編譯。
4. BMP 屬性檔的移植性低，如果在 A 遊戲中定義為不能進入的顏色但在 B 遊戲卻是可以正常移動時，其所造成的結果是不堪想像的。
5. 分為兩個檔案在單機遊戲或許可以接受，但如在網路遊戲時，分為多次讀取將會增加伺服器的工作量，並可能造成遊戲進行中的延遲。

而在使用 XML 來表示遊戲地圖檔後，將可獲得下列的改善：

1. 地圖檔的可讀性提高：不再有以往不易讀取及維護的問題。
2. 與程式的相關性降低，更易於在不同遊戲間移植。
3. 修改容易：不論在遊戲程式設計的那一階段，皆可輕易的加入或修改地圖。
4. 將移動難度放入地圖檔中，不需再分離處理，也不需再另外讀取一地圖屬性檔，避免原來作法中的地圖屬性檔需將各種顏色的相對處理方法寫入程式碼中而產生的相關問題。更可以減少原方法需要修改程式碼才能處理新的移動難度。
5. 以事件編號的方式來記錄其所需要觸發的事件，並配合事件資料庫的作法來降低與事件的相關性。
6. 所需傳送及讀取的檔案數目減少，可加快處理速度。

4.4 使用 XML 可能遭遇到的難題

目前相關遊戲程式設計的 XML 定義，僅有 XGF(XML Game Format)一種，但其所提出的範圍僅在棋類遊戲中的圍棋，且尚未被接受。故建立一個可以被接受的定義是目前的當務之急。

至於在安全性及保密性上，則可以配合其它技術加以解決，如以特定程式轉成無法解讀的格式，或傳輸的時候以封包加密來處理。

但在網路遊戲流行的現在，或許完全不加以設限才是應該考慮的方向，如發行多年而歷久不衰的第一人稱射擊遊戲”絕對武力(CS, Counter-Strike)”就是另一個遊戲”戰慄時空(Half Life)”開放給一般玩家可以自行編輯地圖及程式後所產生出來的遊戲。

第五章 結論與未來工作

以 XML Schema 來處理 RPG 地圖檔是相當可行的，而利用 XML 文件的特性確實可以達到地圖檔與遊戲程式間的相關性降低、可攜性及再用性提高等優點，同時也因為使用 XML 而變的容易閱讀或交換。

本文並實作一個以上述之 XML Schema 為基底的地圖編輯器可提供相關的應用。

在本文所使用的圖片大小與屬性區塊大小是一致的，但目前的遊戲發展上，屬性設定區塊卻是會小於圖片區塊，以便角色在地圖上運動時，可以作更細緻的碰撞偵測，這也是本研究未來的努力方向。

參考文獻

- [1] Gamma E., Helm R., Johnson R., and Vlissides J., "Design Patterns: Elements of Reusable Object-Oriented Software", Professional Computing Series, ISBN 0 - 201-63361-2, Reading Mass.: Addison-Wesley, Oct. 1994.
- [2] " Extensible Markup Language (XML) ", W3C, <http://www.w3.org/XML/>.
- [3] Gamma E., Helm R., Johnson R., and Vlissides J., Design Patterns: Elements of Reusable Object-Oriented Software, Professional Computing Series, ISBN 0 -201-63361-2, Reading Mass.: Addison-Wesley, Oct. 1994.
- [4] " UML 1.1 Notation Guide ", Rational Software Corporation 1998, <http://www.rational.com/>.
- [5] Rational Software, et al. " UML Semantics, Version 1.1, Description of Specification Meta-Association to the Association End Meta-Type ", Rational Software Corporation, Santa Clara, CA, Sept. 1997, pp. 19.
- [6] Paech B., " On the Role of Activity Diagrams in UML, " Procs. <<UML>> ' 98: Beyond the Notation, Mulhouse, France, 1998, pp. 245-250.

- [7] Booch G., Jacobson I. and Rumbaugh J., “ Unified Modeling Language User ’ s Guide ” , Addison-Wesley, Reading, MA, 1997.
- [8] Fowler M. and Scott K., “ UML Distilled: Applying the Standard Object Modeling Language ” , Addison Wesley, Reading, MA, 1997.
- [9] OMG, UML Semantics, Version 1.1, “ OMG documents ad/08-04-97, Sept. 1997.
- [10] OMG, UML Notation, Version 1.1, OMG document ad/08-05-97, Sept. 15, 1997.
- [11] Steve McClure, “ Object Database vs. Object-Relational Databases ” ,
<http://www.cai.com/products/jasmine/analyst/idc/14821E.htm>
- [12] “ Simple Object Access Protocol (SOAP) 1.1 ” , W3C,
<http://www.w3.org/TR/SOAP/>.
- [13] Dung (“ Zung ”) Nguyen and Stephen B. Wong, “ Design pattern from a game ” , SIGCSE 02.
- [14] “ DirectX “ , Microsoft MSDN,
<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000410>
- [15] “ 遊戲分類 ” , 遊戲之魔法城堡
http://www.jl.jl.cn/jlweb/game/basic/fen_lei.html
- [16] Dave Morris, Andrew Rollings 著,徐政棠 譯, “ 電腦遊戲 結構與設計:

理論篇”，電腦玩家

[17] Cathy Stein Greenblat, “Designing Games and Simulations”, SAGA publications.

[18] 結城浩 著,李于青 譯,“Design Patterns 於 Java 語言上的應用”, 博碩文化股份有限公司

[19] Lars Bishop, Dave Eberly, Turner Whitted, Mark Finch, Michael Shantz, “Designing a PC Game Engine”, in Computer Graphics and Applications (1998), IEEE.

[20] 普悠瑪數位科技, “Java 2 網路遊戲設計”, 第三波資訊,

[21] LaMothe, Retcliff, Seminatore & Tyler 著, 徐維成 譯, “揭開毀滅戰士之謎-基礎篇”, 松崗電腦圖書

[22] Michael Morrison 著, 陳常侃 譯, “Java 網路遊戲設計 21 天完全課程”, 儒林圖書

[23] Alan Shalloway, James R. Trott 著, 鐘永哲 譯, “設計模式入門”, 培生教育出版集團

[24] Gamma, Jahnson, Helm, Vlisside 著, 葉秉哲 譯, “物件導向設計模式”, 培生教育出版集團

[25] 張裕益 著, “UML 理論與實作”, 博碩文化

[26] Bradley Bargaen, Peter Donnelly 著, 唐正一 譯, “深入剖析系列—Direct

X ”，松崗電腦圖書

[27] Jim Adams 著，張世敏譯，“2D/3D RPG 角色扮演遊戲程式設計—使用 DirectX ”，博碩文化

[28] 吳逸融、張程焜 著，“RPG 遊戲製作程式設計實務—使用 C/C++ ” 靖宇資訊
科技圖書

[29] XML Schema Part 0:Primer,W3C, <http://www.w3.org/TR/xmlschema-0/>

附錄 A：地圖編輯程式執行畫面



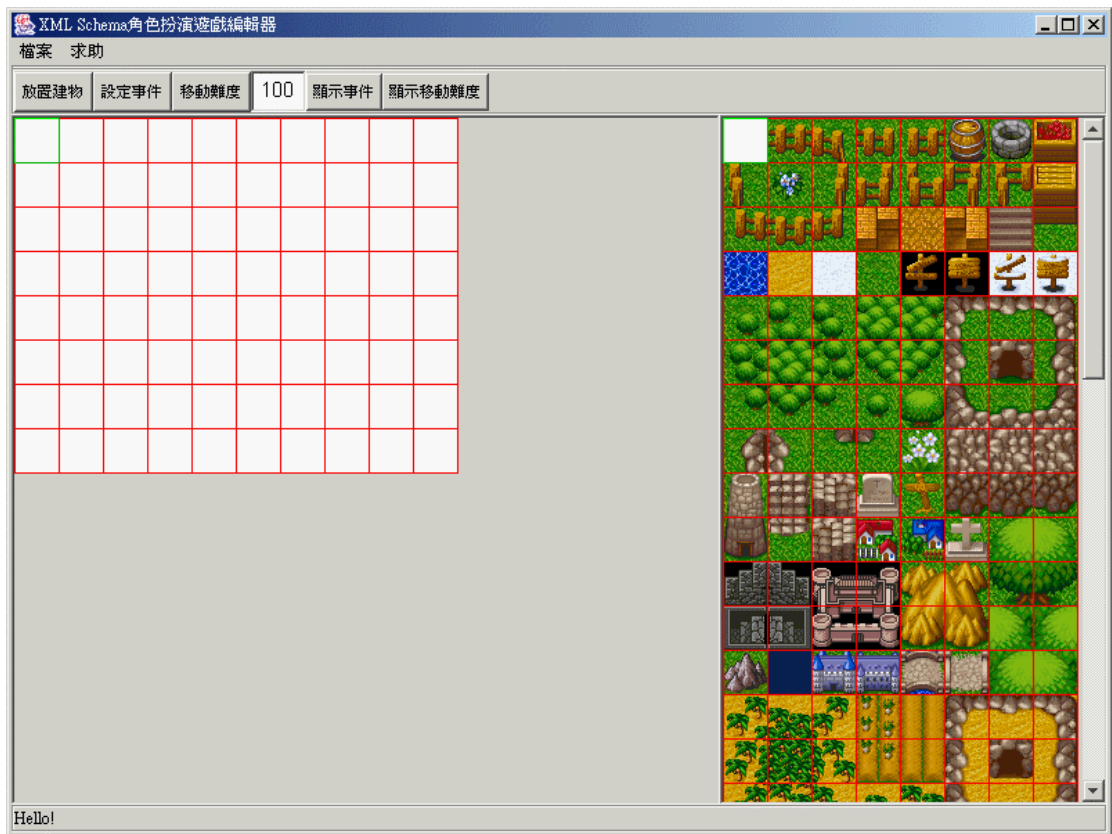
地圖編輯程式執行畫面 (C 語言版本)



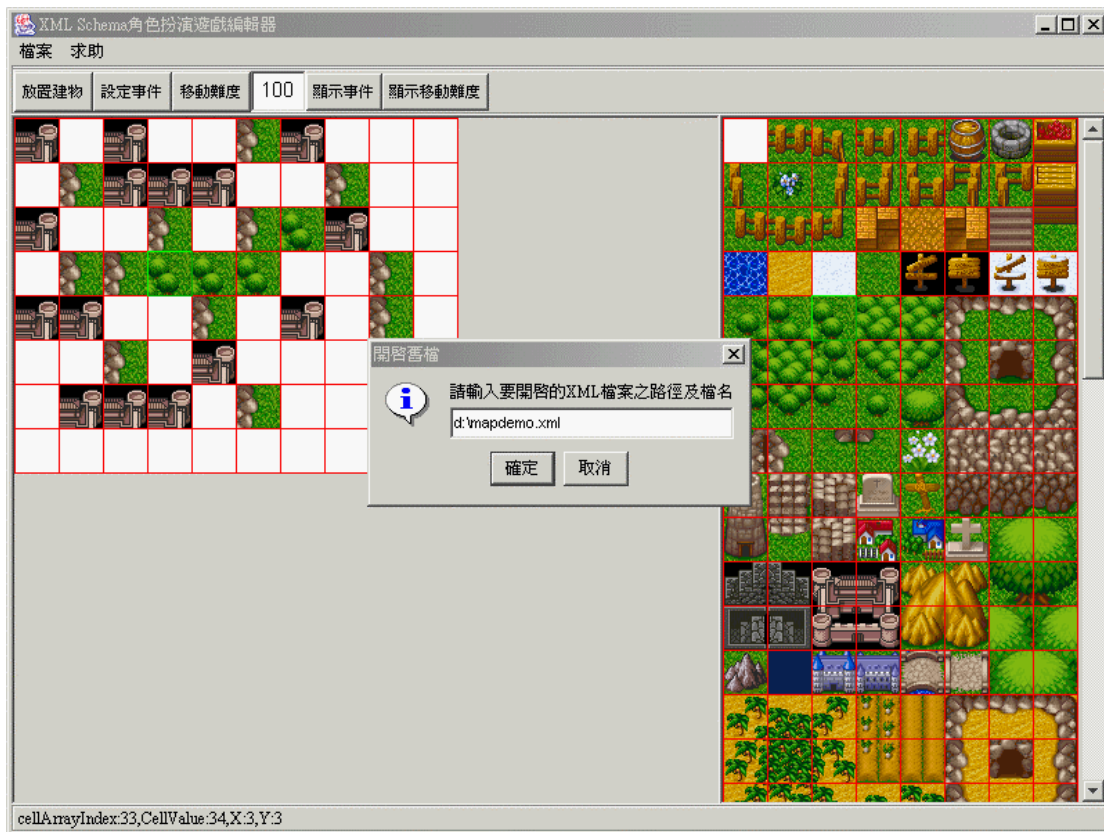
修改移動難度 (C 語言版本)



加入事件編號 (C 語言版本)



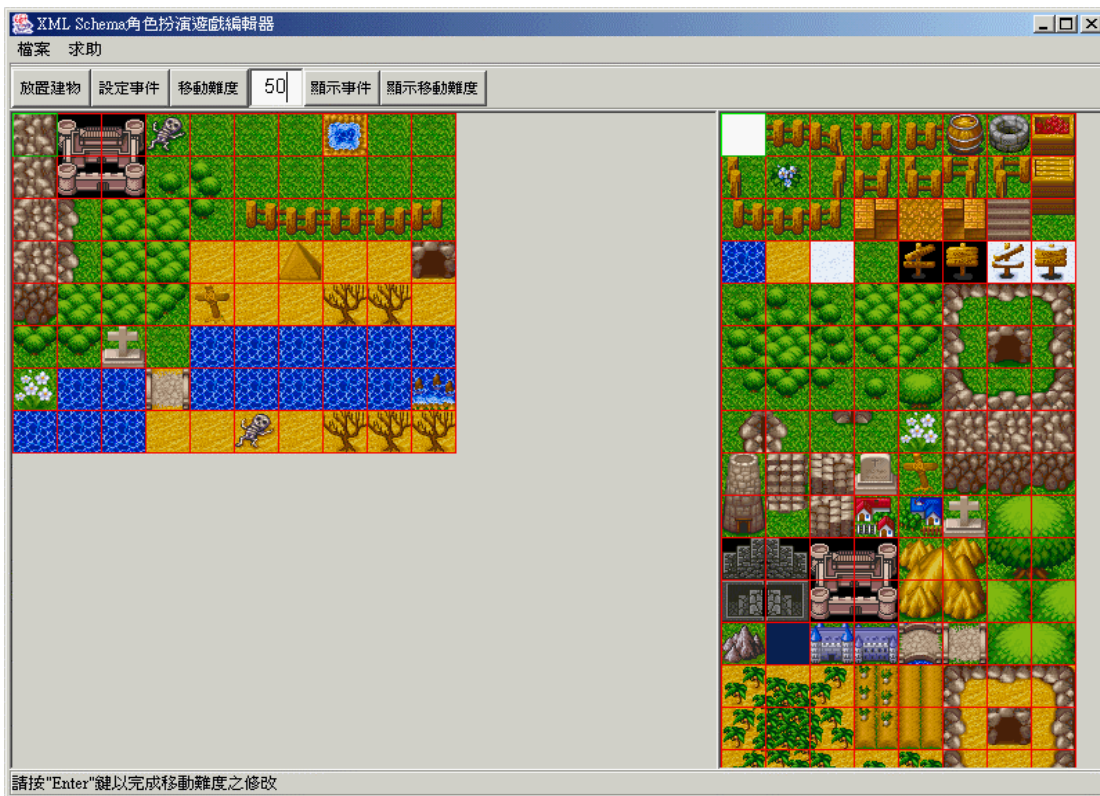
地圖編輯程式執行畫面 (Java 語言版本)



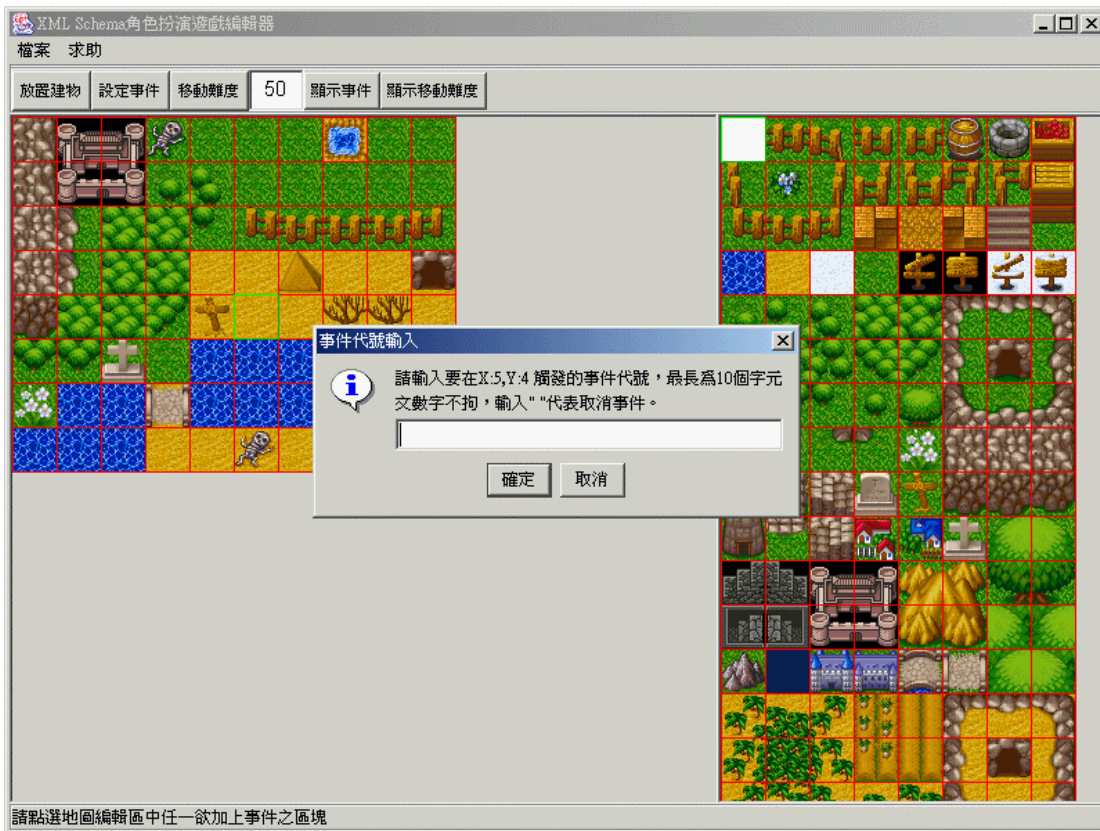
讀入 XML 地圖檔 (Java 語言版本)



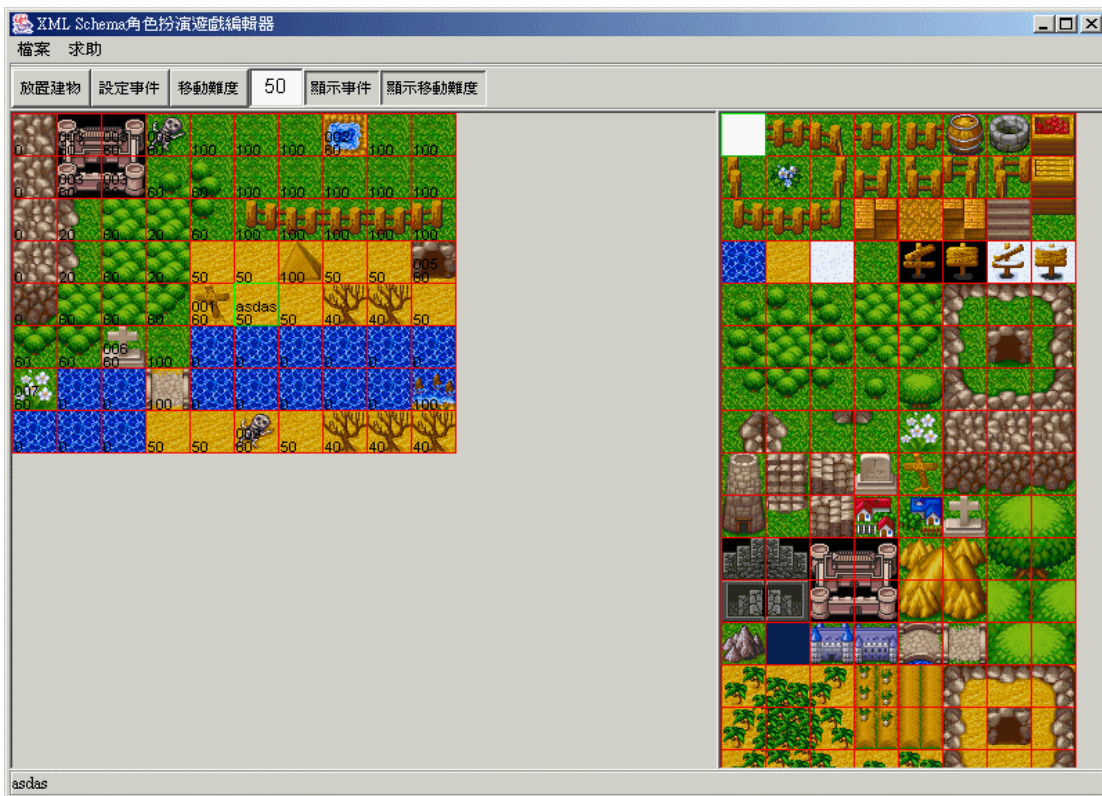
讀入 XML 地圖檔後 (Java 語言版本)



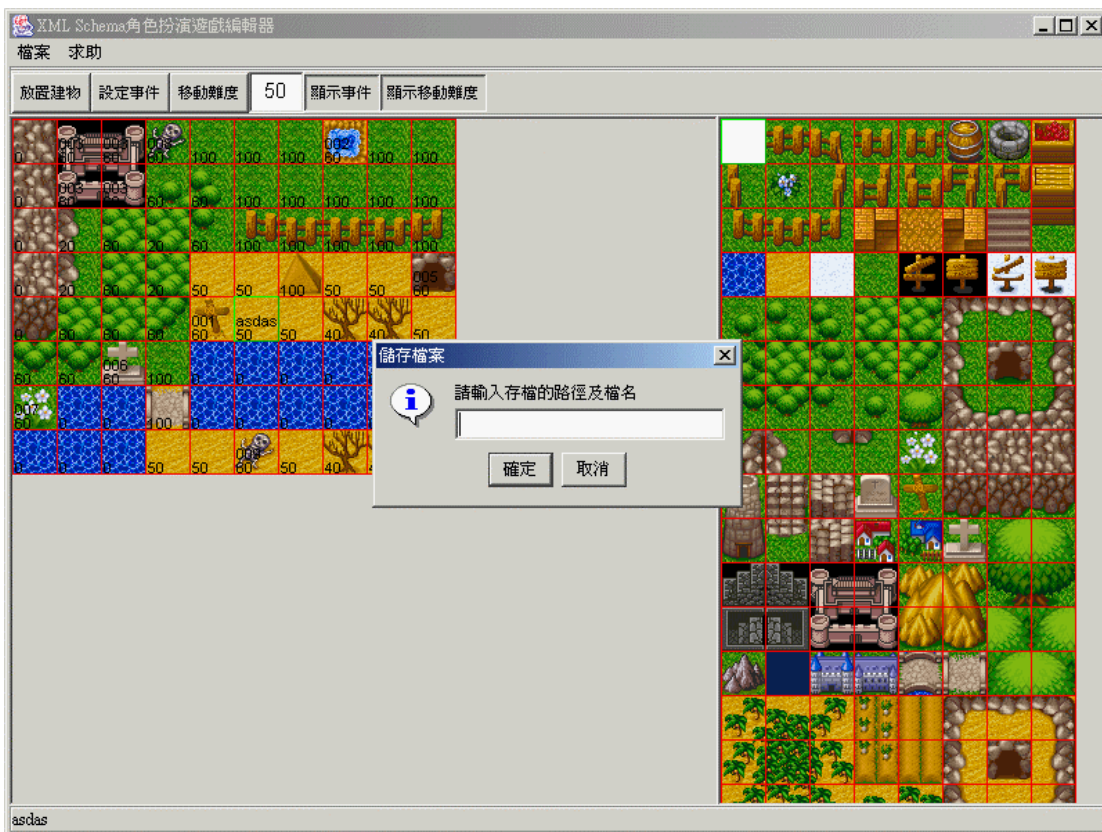
修改移動難度畫面 (Java 語言版本)



修改事件編號 (Java 語言版本)



顯示事件編號及移動難度 (Java 語言版本)



存檔畫面 (Java 語言版本)

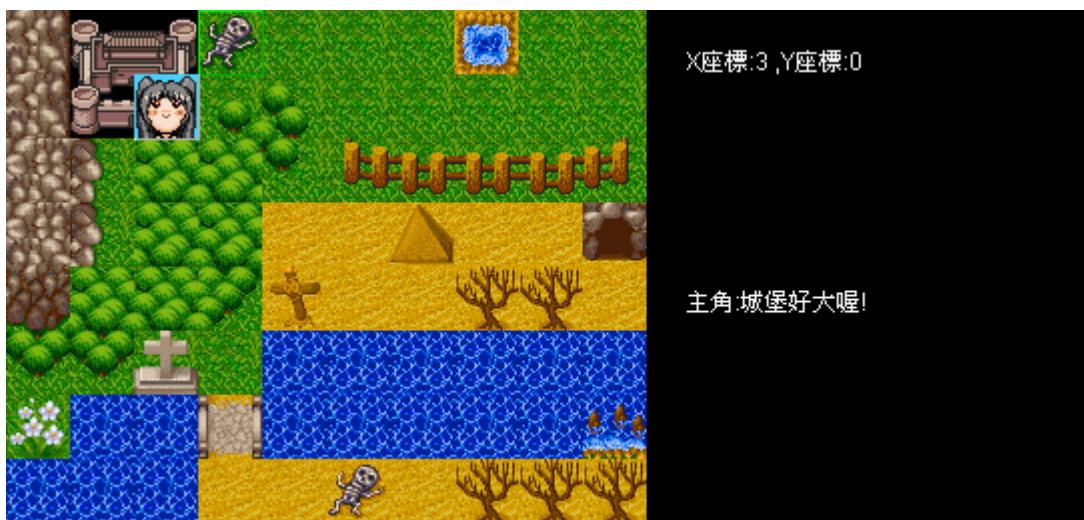
附錄 B：示範程式執行畫面



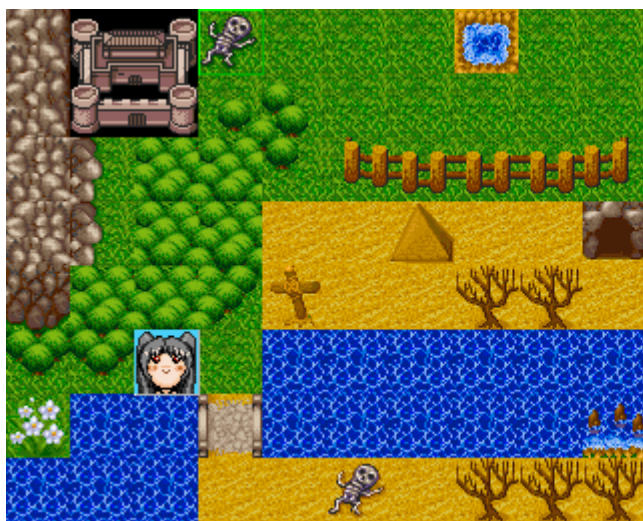
示範程式執行畫面 (Java Applet)



各種移動難度(Java Applet)



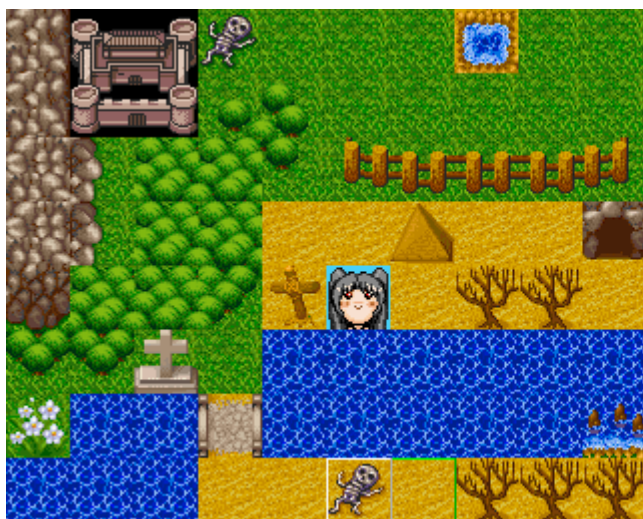
觸發事件 (Java Applet)



X座標:3,Y座標:0

主角:這個是紀念誰的呢?

觸發事件 (Java Applet)



X座標:6,Y座標:7

主角:哇哇!我過不去!!

無法通過的地形 (Java Applet)

附錄 C : 地圖編輯類別

xml4map

Class RPGMap

public class RPGMap

Title: XML RPG Map

Description: For R.P.G. Map use

Copyright: Copyright (c) 2003

Version:

1.0

Method Detail

RPGMap

```
public xml4map.RPGMap RPGMap()  
    Init some value for this class * @return RPGMap
```

readRPGMapFile

```
public boolean readRPGMapFile(File Fin)  
    Read a XML type RPG Map file
```

Parameters:

Fin -

Returns:

boolean for reads result.

writeRPGMapFile

```
public boolean writeRPGMapFile(File Fout)  
    Write a XML type RPG Map file
```

Parameters:

Fout -

Returns:

boolean for writes result.

setMoveLevel

```
public void setMoveLevel(int aIndex,  
                        int newMoveLevel)
```

Set a Move level depended on array index

Parameters:

aIndex - Which Index of MoveLevel Array will be change.

newMoveLevel - New value of MoveLevel

setEventID

```
public void setEventID(int aIndex,  
                      String newEventID)
```

Set a Event ID depended on array index

Parameters:

aIndex - Which Index of EventID Array will be change.

newEventID - New EventID

getMoveLevel

```
public int getMoveLevel(int aIndex)
```

Return a move level value form Array

Parameters:

aIndex - Which Index in MoveLevel array

Returns:

MoveLevel

getEventID

```
public String getEventID(int aIndex)
```

Return a Event ID value form Array

Parameters:

aIndex - Which Index in EventID array

Returns:

EventID

RPGMapXMLparser

private void RPGMapXMLparser()

For internal use- parse XML File and store value in this Object

toBufferedImage

private BufferedImage toBufferedImage(Image image)

For internal use - Get Image file and transform to buffered image for next process

Parameters:

image -

Returns:

BufferedImage



誌 謝

兩年的學習生涯，令我生命中添加了許多色彩與回憶，師長、同窗、工作夥伴及親愛的家人，首於篇幅之初將感謝獻給我所愛的人。

我能順利的取得碩士學位，必須感謝我的指導教授朱正忠老師，在老師身上我不但學到完整的研究過程，更重要的是老師圓融的處事技巧以及良好的人際關係，都是我所努力學習的對象，而在學習的過程中，老師更啟發了我許多的研究方向及靈感，讓我受益良多。

最後，更感謝資訊技術實驗室的所有學長、同學及學弟，陪伴我渡過這一段美好的時光。