

國立交通大學

電機資訊學院 資訊學程

碩士論文

網路連線角色扮演遊戲產生器的設計與實作



The Design and Implementation of
a Network Role-Playing Game Generator

研究生：曾華堃

指導教授：陳登吉 教授

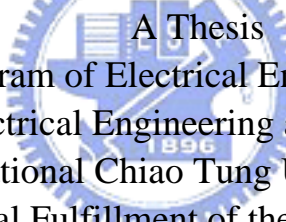
中華民國九十四年一月

網路連線角色扮演遊戲產生器的設計與實作
The Design and Implementation of
a Network Role-Playing Game Generator

研究生：曾華堃
指導教授：陳登吉 博士

Student : Hwa-Kuan Tseng
Advisor : Deng-Jyi Chen

國立交通大學
電機資訊學院 資訊學程
碩士論文



A Thesis
Submitted to Degree Program of Electrical Engineering and Computer Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Computer Science
January 2005
Hsinchu, Taiwan, Republic of China

中華民國九十四年一月

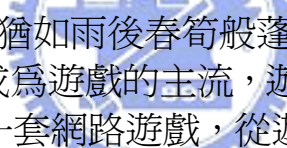
網路連線角色扮演遊戲產生器的設計與實作

學生：曾華堃

指導教授：陳登吉 博士

國立交通大學電機資訊學院資訊組

摘要



近年來網路遊戲猶如雨後春筍般蓬勃的發展，加上網路的普及，網路遊戲已經逐漸成為遊戲的主流，遊戲公司一一投入了網路連線遊戲的製作。但製作一套網路遊戲，從遊戲企劃、軟體的分析、設計到實作與測試，整個流程與傳統的單機版遊戲相去不遠，仍需耗費大量的人力及時間重覆同樣的步驟。

我們過去曾將針對單機版角色扮演遊戲提出使用一套遊戲編輯工具來協助遊戲的開發過程。這套工具有簡易的視覺化編輯介面，可以讓不懂程式設計的企劃人員透過這套工具來編輯遊戲。企劃人員可以隨時規劃及修改遊戲的腳本內容，而不會增加程式設計師的負擔。使遊戲的發展速度變快，同時遊戲的品質也能得到保證。

網路連線遊戲與單機版遊戲的主要差異，在於網路遊戲可提供多個玩家同時進行遊戲，其主要在於連線功能的支援與內部玩家資料結構上的變化，對於使用者介面的部分與單機版遊戲並無差別。以視覺化編輯介面來開發，同樣可以減少程式設計師與遊戲企畫人員之間的溝通問題，降低遊戲的發展週期。本論文將針對網路連線版的角色扮演遊戲，來驗證所提出的網路連線角色扮演的遊戲編輯系統的能力及應用。

The Design and Implementation of a Network Role-Playing Game Generator

Student : Hwa-Kuan Tseng

Advisor : Dr. Deng-Jyi Chen

Department of Computer Science and Information

Engineering National Chiao Tung University

Abstract

In recent years, the development of network game has become more and more flourishing. Along with the spread of network, network game has become the mainstream of all computer games, and many game companies invested in it. Like the traditional PC based game, the network game also has to invest a lot of man-power and time on game plan, analysis, design, implement, and testing, etc.

We had built a game generator to generate a PC based role playing game. It provides an easy and friendly visual interface to produce a game story for script designer who does not understand the program design. Script designer can design and modify script of games without the cooperation with UI programmers. It is not only speed up the game development, but also reduce the cost on UI implementation.

The main difference between network game and PC based game is that network game can support many players in the same time. On the user interface, the network based RPG game generated by the proposal network game generator is exactly the same as the PC based single player RPG game. Players do not need to spend extra effort to learn the network based RPG game generated by the proposal approach. The proposed approach also reduces the communication problems between programmers and script designers. Thus, it speeds up the game developing cycle. Specifically, examples are used to demonstrate the usefulness and applicability of the proposed network game generator.

誌謝

本論文得以順利完成，首先要感謝指導教授陳登吉老師的細心指導與諄諄教誨，使我在相關研究領域中獲益良多，在此對老師致上無盡的感謝。

此外，我更要感謝同學卓勇君、王家偉、林建中、劉柏昌等人，在這些年的求學過程中，於研究與課業上給我的指導及幫助，讓我得以完成本篇論文。

最後，感謝養育及栽培我長大成人的雙親及祖父母所給我的支持與鼓勵，使我得以在兼顧課業與工作的同時，能專心完成論文免去後顧之憂。謹將此論文獻給所有關心我及愛護我的人



目錄

中文摘要.....	I
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vii
圖目錄.....	viii
圖目錄.....	viii
一、緒論.....	1
1.1 網路遊戲軟體簡介.....	1
1.1.1 遊戲軟體的種類與市場規模.....	1
1.1.2 國產遊戲軟體發展史.....	2
1.1.3 網路連線遊戲的類型.....	3
1.1.4 單機版遊戲的連線化.....	4
1.2 遊戲開發的分工與流程.....	4
1.2.1 遊戲開發主要人員的分工.....	4
1.2.2 遊戲開發流程.....	6
1.3 研究動機及目的.....	8
1.3.1 研究動機.....	8
1.3.2 研究目的.....	9
1.4 研究步驟.....	9
1.5 章節說明.....	9
二、相關研究.....	10
2.1 角色扮演遊戲簡介.....	10
2.2 角色扮演遊戲的組成架構.....	11
2.2.1 玩家角色.....	11
2.2.2 場景.....	11
2.2.3 素材資源.....	12
2.2.4 程式模組.....	12
2.3 傳統方式的遊戲製作.....	13
2.3.1 遊戲劇情開發流程.....	13

2.3.2 撰寫遊戲劇情的方法.....	14
2.3.3 遊戲劇情開發時常遭遇的問題.....	16
2.4 視覺化方式的遊戲製作.....	16
2.4.1 視覺化方式遊戲劇情開發流程.....	17
2.4.2 劇情開發方式分析.....	19
三、遊戲系統需求分析.....	20
3.1 遊戲開發的成員關係.....	20
3.2 從玩家的角度.....	21
3.3 從遊戲設計者的角度.....	25
3.4 從系統製作者的角度.....	26
3.5 遊戲系統的需求分析.....	28
3.6 視覺化遊戲產生器的架構.....	28
四、網路連線角色扮演遊戲產生器之設計.....	32
4.1 在 3D 角色扮演遊戲編輯系統中所設計製作的遊戲產生器的能力.....	32
4.1.1 編輯場景的操作流程.....	33
4.1.2 編輯演員事件的操作流程.....	36
4.1.3 演員事件描述檔轉換.....	42
4.2 在 3D 角色扮演遊戲編輯系統之遊戲產生器中所待解決的問題.....	42
4.3 加入網路連線支援.....	43
4.3.1 網路連線遊戲的系統架構.....	43
4.3.2 選擇的連線架構.....	45
4.4 加入多重玩家支援.....	45
4.4.1 單機與連線環境中角色資料結構的差異.....	46
4.4.2 角色資料在連線遊戲中的特性.....	46
4.5 加入主角屬性與劇情編輯支援.....	47
4.5.1 編輯主角屬性資料.....	48
4.5.2 編輯主角劇情資料.....	48
4.6 改善 3D 角色扮演遊戲編輯系統後之遊戲產生器.....	50
五、網路連線角色扮演遊戲產生器之實作.....	53
5.1 遊戲播放器之網路連線模組實作.....	53
5.1.1 連線管理.....	53

5.1.2 連線流程.....	54
5.1.3 訊息處理.....	55
5.2 多重角色資料結構實作.....	56
5.2.1 多重玩家角色的資料結構.....	57
5.2.2 連線中的多重玩家角色管理.....	58
5.3 主角屬性與事件編輯功能實作.....	59
5.3.1 主角屬性資料編輯.....	59
5.3.2 主角劇情事件編輯.....	60
5.4 系統模組架構.....	62
六、系統展示.....	63
6.1 視覺化編輯階段.....	63
6.1.2 編輯主角屬性資料的操作流程.....	63
6.1.2 編輯主角劇情事件的操作流程.....	66
6.2 遊戲播放階段.....	68
七、結論與未來發展方向.....	74
7.1 結論.....	74
7.2 未來發展方向.....	74
參考文獻或資料.....	75



表目錄

表 1	1998~2004 年全球遊戲市場銷售額[1]	1
表 2	遊戲開發流程中的人員分工[14]	5
表 3	劇情描述檔格式	15
表 4	傳統方式與視覺化方式設計遊戲劇情的比較	19



圖目錄

圖 1	亞太地區線上遊戲市場佔有率	3
圖 2	遊戲開發流程.....	8
圖 3	角色扮演遊戲的標準流程.....	10
圖 4	角色扮演遊戲的基本組成架構	12
圖 5	傳統方式的遊戲開發流程.....	14
圖 6	視覺化方式的遊戲開發流程	17
圖 7	以視覺化工具自動轉換描述語言	18
圖 8	遊戲開發過程中的成員關係圖	21
圖 9	從玩家角度來看一個角色扮演遊戲的組成要素	22
圖 10	從遊戲設計者角度來看一個角色扮演遊戲的組成要素.....	26
圖 11	從程式設計師的角度來看遊戲的成員關係圖	27
圖 12	視覺化遊戲產生器的架構圖	29
圖 13	以視覺化遊戲產生器開發遊戲的流程.....	30
圖 14	在 3D 角色扮演遊戲產生器內模組的資料流程圖	32
圖 15	場景編輯器的主畫面	33
圖 16	場景編輯器 -- 新增一個場景.....	34
圖 17	場景編輯器 -- 設定演員的資訊	35
圖 18	事件編輯流程(1)	36
圖 19	事件編輯流程(2)	37
圖 20	編輯事件範例(1)	37
圖 21	編輯事件範例(2)	38
圖 22	編輯事件範例(3)	39
圖 23	編輯事件範例(4)	39
圖 24	編輯事件範例(5)	40
圖 25	編輯事件範例(6)	40
圖 26	編輯事件範例(7)	41
圖 27	編輯事件範例(8)	41
圖 28	PEER-TO-PEER 架構	44
圖 29	SERVER-CLIENT 架構	45
圖 30	網路遊戲處理玩家資料結構的改變	46
圖 31	在不同 PEER 中玩家所使用角色的關係	47
圖 32	主角劇情事件的組成結構.....	49
圖 33	改善後的遊戲產生器內模組的資料流程圖.....	51
圖 34	改善後的遊戲播放器的網路連線架構圖	52

圖 35	DIRECTPLAY連線流程	55
圖 36	遊戲進行中多重角色的類別圖	57
圖 37	遊戲進行中資料封包的交換方式	59
圖 38	事件編輯器的類別圖	61
圖 39	新版遊戲產生器的系統類別圖	62
圖 40	編輯主角遊戲屬性	64
圖 41	套用主角遊戲屬性(1)	65
圖 42	套用主角遊戲屬性(2)	65
圖 43	編輯主角劇情事件(1)	66
圖 44	編輯主角劇情事件(2)	67
圖 45	編輯主角劇情事件(3)	67
圖 46	網路連線流程(1)	68
圖 47	網路連線流程(2)	69
圖 48	演員事件播放流程(1)	69
圖 49	演員事件播放流程(2)	70
圖 50	網路連線流程(3)	71
圖 51	使用主角劇情事件(1)	71
圖 52	使用主角劇情事件(2)	72
圖 53	使用主角劇情事件(3)	73



一、緒論

1.1 網路遊戲軟體簡介

1.1.1 遊戲軟體的種類與市場規模

全球遊戲軟體市場大致可以區分為大型遊戲機(Arcade)、電視遊戲器(Video Game)、個人電腦遊戲(PC Game)及線上遊戲(On-line Game)。線上遊戲指的是由網路伺服器及使用者端個人電腦共同完成的遊戲架構，和傳統單機版遊戲或以區域網路多人對戰方式之區域網路型遊戲(LAN Game)不一樣的是，線上遊戲業者主要的收益來自遊戲玩家連線帳號的使用時數費，和單機版遊戲或區域網路型遊戲主要收益來源為販售遊戲套裝軟體(Package)的經營模式並不相同。

根據 MIC 在 2002 年 1 月時所做的「1998-2004 年全球遊戲產值」統計資料來看[1]，觀察 1998-2004 年間各個遊戲產業的成長率，可以發現以線上遊戲成長幅度達 44% 為最大，已超越了傳統的單機版遊戲。

表 1 1998~2004 年全球遊戲市場銷售額[1]

1998~2004 年全球遊戲市場銷售額 單位：百萬美元

	1998	1999	2000	2001	2002(F)	2003(F)	2004(F)
Video Game	14,135	14,837	13,526	18,461	21,295	23,398	24,461
PC Game	2,388	2,644	2,871	3,024	3,187	3,396	3,536
Arcade Game	1,776	2,008	2,345	2,735	3,156	3,602	3,937
On-line Game	803	1,004	1,476	2,078	3,003	3,912	4,521
合計	19,102	20,493	20,218	26,298	30,641	34,308	36,455

在屈指可數的幾個能給網路帶來利潤的業務當中，網路遊戲的收費被認為是網路歷史上最成功的盈利模式之一。從遊戲開發公司、發行營運商到 ISP 業者都可在此一市場獲得巨額利潤。遊戲開發業者從中收取販售與遊戲版權費，與後續的更新版本的維護費用。遊戲營運商經由架設的網路平台向玩家提供服務及收取月費。而 ISP 業者則透過提供網路機制從玩家身上收取網路撥接費用。加上週邊產品系列的書刊雜誌、紀念品、玩偶、電影等等.....整個產業的獲利相當驚

人。在韓國，網路遊戲產業的發展甚至已經超越了他們向來引以為豪的汽車工業。而在國際上，整個遊戲業的收入已經超過了電影業，成為娛樂業最炙手可熱的賺錢明星。

網路遊戲之所以賺錢，主要在於與單機版遊戲營利方式的差別。單機版遊戲的主要收益來源為販售遊戲套裝軟體(Package)的經營模式，但由於盜版率過高，達 95% 的使用者所用的都是盜版軟體，嚴重影響到產品的銷售狀況。而網路遊戲的收費方式來自遊戲玩家連線帳號的使用時數費。用戶註冊後得到一個 ID 帳號，但要使用這個帳號玩遊戲，還必須再去購買點數卡，將卡裡的點數(錢)存到帳號裡才可以開始玩。帳號裡的錢用完了，就要再去買點數卡來充值，或者玩家也可以購買包月制，有效地防止了盜版的衝擊，遊戲廠商自然是趨之若鶩。因此『如何在有限時間內、以有限的資源開發出足以吸引玩家的遊戲，進而獲取最大利潤』將是擠身全球遊戲市場最大的挑戰。

線上遊戲受到投注的主要原因，除了抑止盜版問題之外，並且也提供了過去賣斷整套軟體所缺乏的成熟營運模式。另外，由於國際上動畫電影的熱賣，以炫人的 3D 動畫及特效後製公司，結合在遊戲軟體吸引顧客的需求，使得遊戲軟體產業與 3D 動畫及特效後製產業，可說是數位經濟時代明星產業中的明星。預測顯示，在互聯網這一新時代媒介的推動下，全球娛樂業與傳媒業今後幾年都將以每年 7.2% 的速度增長，2005 年將達到 1.2 萬億美元的規模。其中，全球遊戲市場的收入將增至 499 億美元。

1.1.2 國產遊戲軟體發展史

國產遊戲的歷史始於 Apple II 的時代，就有一些簡單的遊戲作品。但因當時電腦運算能力與中文系統不發達的限制，多為一些英文的簡單小遊戲，也未成為市場主流。玩家所玩到的遊戲仍以國外遊戲為主[2]。

隨著個人電腦從 Apple II 的時代進入 IBM PC 的時代，國內的遊戲產業也開始有了變化。原本以代理國外遊戲為主的國內遊戲市場中，開始出現了一些國人自製的本土中文遊戲。隨著電腦螢幕由單色進入到彩色的時代，中文遊戲的產品也漸趨成熟，出現了許多知名的遊戲如《軒轅劍》、《大富翁》、《仙劍奇俠傳》、《俠客英雄傳》等，使國產遊戲的開發在市場上成功的佔住了一席之地。

以當時台灣中文遊戲產業的現況來看，我們會發現國內的遊戲類型大多集中在幾種類型上。角色扮演遊戲與策略遊戲等為市場上銷售的主力，每年都有許多同類的新產品推出。這幾類作品也成了國產遊戲中製作技術與市場最成熟的類

型。

而隨著電腦科技的進步以及國內設計遊戲水準的提升，這幾年來，國內的遊戲也開始逐漸的朝 3D 圖形以及網路連線的方向前進。國內線上遊戲始自於西元 1999 年華彩推出的「萬王之王」，但初期並未形成熱潮，一直到西元 2000 年，華義國際、遊戲橘子、智冠相繼推出各種不同題材的線上遊戲後，國內網路遊戲蔚為風潮，各遊戲的會員人數迅速爆增，成千上萬的玩家們亦能樂在其中。根據 IDC(國際數據資訊)研究報告顯示[3]，亞太地區線上遊戲市場佔有率中，台灣僅次於韓國居於第二。從購買線上遊戲服務的營收來看，不但顯示出玩家相當樂意以付費方式在網路上與其它玩家共同對打，整個遊戲產業的市場規模也出現了革命性的轉變。因為線上遊戲解決了單機遊戲長久以來的盜版問題，直接促使遊戲軟體業的市場產值迅速成長，然而隨著產值的擴大，不僅激勵了產業的成長，更再度引領業者跨上國際舞台。



圖 1 亞太地區線上遊戲市場佔有率

1.1.3 網路連線遊戲的類型

留意一下目前網咖所流行的電腦遊戲，我們會發現幾乎都是連線遊戲的天下。不論是幾可亂真的第一人稱射擊對戰遊戲(FPS)、或是多人線上角色扮演遊戲(Multi-User Online RPG)，這種透過網際網路的連結來讓世界各地的人一起互動、一同分享冒險過程的遊戲方式，已經在電腦遊戲中(或是未來的 Console Game)占據了一個十分重要的地位。

所謂的網路多人互動遊戲，簡單的說，就是透過網路把所有參與互動的電腦

(或說是玩家) 聯結起來。透過網路的聯結與互動訊息的傳遞溝通，讓分散各地的電腦組成了一個共享的虛擬世界(Shared Virtual World)，讓每個玩家各扮演一個角色與其他人互動。這類的遊戲因為是與真人互動，所以變化更加的多端，往往還摻雜了複雜的人性在裡面。

目前的網路多人互動遊戲大致可以分成三大類：一種是益智類的線上遊戲。此種遊戲在進行過程中同時參與的玩家人數不多，遊戲內容以益智類為主如橋牌或麻將。目前有包括宏碁戲谷、KIKI 歡樂王國、iGame 遊戲網間等業者提供此遊戲服務。第二種為即時連線對戰遊戲。此類遊戲強調以第一人稱的展現方式與即時回應的處理速度，來讓玩家體驗真實戰鬥的逼真與臨場感。目前在網咖中最紅的遊戲-戰慄時空之絕對武力(Counter-Strike)，即屬於此類的遊戲。第三種則是多人線上角色扮演遊戲 (Massive Multi-player RPG)。多人連線的角色扮演遊戲和過去單人的角色扮演遊戲最大不同就是在於，玩家們可以彼此結盟、即時線上分享心得與彼此交換寶物等。這種有長期連盟(或是群體)特性的遊戲方式與不斷改變的任務內容，讓許多這類遊戲線上遊戲歷久不衰，成為最受歡迎的網路遊戲類型。國內最有名的例子即是仙境傳說(RO)、天堂、石器時代等遊戲。

1.1.4 單機版遊戲的連線化

隨著網路遊戲的蓬勃發展，不但線上連線遊戲蔚為盛行，連傳統的單機版遊戲也開始在功能中加入連線能力。單機版遊戲朝向連線化的優點就在可以抵制盜版的日益猖獗。例如玩家可在購買遊戲時得到一組帳號密碼，在安裝好遊戲軟體之後需以連線的方式用該組帳號密碼進行註冊後才可開啓遊戲功能；或是該遊戲軟體只包含基本的遊戲內容，若想進行其他的遊戲內容則需以連線付費下載的方式來取得，以增加該遊戲的趣味性等。又因為具備連線功能，也可以和其他玩家進行互動。

1.2 遊戲開發的分工與流程

以下我們整理一些經驗豐富的遊戲開發者所提出關於遊戲開發的人員分工及開發流程，來探討遊戲開發的程序[4][5][6][7][8][9][10][11]。

1.2.1 遊戲開發主要人員的分工

開發一套遊戲必須有各種不同專業領域的人員來分工合作開發，並且在開發時可以讓各組人員的工作平行進行，來減少開發的時間與成本[7][9][10][11][12][13]。一套遊戲的主要開發成員，一般分為程式設計、遊戲企畫、

與美工、聲音人員等製作人員。並有製作人負責統籌及協調各方的合作情形、意見糾紛與進度掌控。

表 2 遊戲開發流程中的人員分工[14]

	職務	工作性質	需求人才背景
製作 團隊	製作人	製作團隊總指揮工作協調及製作進度的掌控	資深設計人員轉任，對程式、美術企劃、音樂有一定程度了解，有團隊管理能力
	執行製作	產品經理以及產品概念的規劃與管理	資深設計人員轉任，對程式、美術企劃、音樂有一定程度了解，有團隊協調能力
	遊戲開發企劃	產品原創設計者，遊戲腳本製作	有創意，有組織規劃能力，有敏銳市場感覺
	美術設計	產品視覺呈現者	有創意，有美術技術及設計能力佳者，美術相關設計科系背景
	程式設計	遊戲內容互動性與動畫、AI以及場景的程式撰寫	有創意、對程式設計熱愛、資訊工程、軟體工程、資訊管理、系統設計相關人員

◎ 企劃與程式：

企劃人員負責提出遊戲的創意。在遊戲開發初期，企劃人員最重要的工作，是與程式設計人員規劃遊戲系統，企劃人員定出要求，程式設計人員就現有技術與時間提供不可行與替代方案給企劃。一旦定案，程式設計人員便負責提供企劃人員各種場景、劇本、人物與物品等編輯器，讓企劃人員用以輸入一個遊戲的劇情資料。最後，重點是完成主程式的開發，確定雙方同意的創意無誤的在電腦上執行。

◎ 程式與美工：

美工內部的分工，一般分為角色、物品、怪物等素材設計與電腦動畫(Computer Graphic)、廣告海報等美術設計。美工通常比較不參與系統的討論，大多是在遊戲定案後，給予企劃人員與程式設計人員美術上的專業諮詢，譬如遊戲圖形採取何種格式、色盤如何安排、風格走向、製作測試圖形，讓程式試做無誤等等，工作量十分的大。一般與程式的互動，從圖形格式、命名原則的規定到

3D 模型的複雜度與動畫張數等，都需要經雙方確認無誤後，方能正確應用於遊戲中。

◎ 企畫與美工：

兩者的關係互動性很大，企劃人員必須隨時關心美工設計的圖形是否符合遊戲風格，並確定其設計的遊戲地圖是否符合劇情所需。

1.2.2 遊戲開發流程

開發一個遊戲需要歷經規劃、分析、設計、實作等階段。開發時必須確認一個階段正確無誤後再進入下一個階段。因此開發遊戲的流程如同在軟體工程中提到的軟體生命週期[7][8][9][12][13]。遊戲軟體發展的流程可分為遊戲提案期、專案企劃期、製作開發期、測試後製期、以及發行檢討等五個階段：

(1) 遊戲提案期：遊戲提案期即為製作遊戲的提案時期，企劃應先就市場現況、目標消費群、遊戲定位與類型來做判斷與決定是否提案，接著對於此遊戲的大致內容、玩法特色、遊戲性，以及與其它遊戲的區隔等進行提案的撰寫，並規劃出估算的人力配置、資金、遊戲製作時間。

當完成這樣的提案撰寫之後，接著就是與投資人、經營者、行銷部門、美術總監、程式總監等，進行開會與討論。經過討論之後，通常會提出許多的問題點，關鍵點是「市場性」、「可行性」與「遊戲性」，並研究上述的問題點是否能克服或予以修正，可能最終的結果是此提案遭退件，另行再擬新案等。

(2) 專案企劃期：如果提案獲得通過，就會正式開立專案，並且指定專案名稱與負責人，接著就進入企劃期。專案企劃期可分為前期、中期與後期，在前期的部分，主要在於資料蒐集與撰寫。遊戲企劃此時需開始將這次的提案改寫成詳細的初步企劃書，並將背景世界觀、故事大綱、遊戲主畫面、遊戲玩法、人物設定、場景的美術風格、音樂風格等，在此時與其他三個小組成員密集的進行開會與討論，並互相激盪彼此之間的創意，並且讓各小組成員對此次的遊戲能有更深一層的了解與認同，以建立起參與感。

在經歷這段時期的討論之後，遊戲企劃便必須彙整各小組的意見，進行企劃書的修正，並且開始分析工作量、場景的大小與規模、視覺畫面、遊戲規則、物品種類、視窗開啓方式、角色與非玩家角色(Non-Player Character, NPC)配置等。此時，美術設計與程式設計小組，也開始作其前置工作。美術設計需開始了解與揣摩美術風格與相關設計，程式設計也需試做可能使用的各項程式與效果，並且對於遊戲主程式是否需要重新撰寫，還是延伸改進之前的舊引擎做出決定。

專案企劃期的中期，始於一個原型(prototype)的製作，用以大致表達此遊戲的模擬畫面，並可確立之前所討論的角色造型與遊戲風格、遊戲的進行方式與介面，在實際的遊戲動作中，是否可以確實表現出遊戲性，驗證企劃與提案時的各項特色與重點。完成後，再與投資人、經營者、行銷部門、美術總監、程式總監等進行審查會議，此時可能加入代理商與發行商，共同對此遊戲進行評估。

經過上述的審查並確定無誤之後，才會進入專案企劃期的後期，完成完整的企劃書撰寫，確定的工作項目與時程規劃，安排人力與彼此間的支援，並正式進入製作開發期。

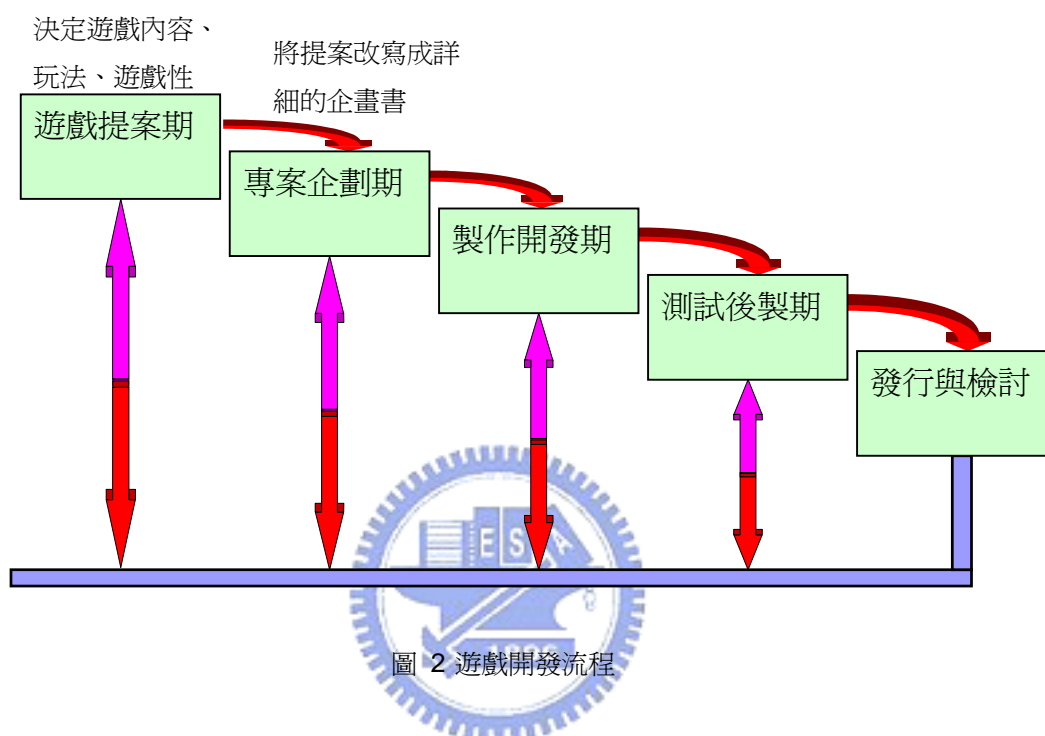
(3) 製作開發期：開發時期的部分，最重要的就是管控時程，必須讓每項工作準時完成，並且處理實際開發時，所遇到的各種狀況。甚至，盡可能在不影響整體遊戲的原貌與工作量之下，適時的加入更多新的想法與做法。此時目標的第一步為 **First Playable** 遊戲 demo 版本的製作。此版本應將該有的基本功能加入，但關卡與場景可以只完成部分，demo 版比原型更加嚴謹而詳細，並且可為開發期間舉辦的相關展覽，預作宣傳展示準備。

在整體遊戲開發完成度約至 60%時，便可釋出部分相關的設定圖與遊戲故事劇情，並配合行銷與宣傳的需要，進行新聞雜誌稿的撰寫，以及圖片刊登，並繼續著手完成 α 版。 α 版是接近完成的全功能版本，未加入的內容只剩遊戲片頭與結局畫面，或其他非關程式系統功能與畫面表現的內容。

(4) 測試後製期：測試後製期主要分為 α 版測試、 β 版測試與遊戲相關後製。 α 版測試為遊戲開發小組所進行的內部測試，主要測試遊戲內的各項系統功能、遊戲性以及平衡度等，必須將關鍵性與顯而易見的程式瑕疵(bug)與畫面顯示的錯誤，予以修正。完成 α 版測試並修正之後，便進入 β 版測試，主要進行遊戲穩定度的測試，藉由不同玩家或發行商進行外部測試，並觀察不同軟硬體規格上的遊戲運作情形。由測試樣本回覆之結果，發現是否有任何遺漏的瑕疵、錯字、圖形等，修正後，便完成最後的 **Gold** 版，也就是進行壓片並上市銷售的版本。關於後製部分，包括遊戲的封面與盒裝設計、說明書的編製以及宣傳海報等等，也都在遊戲製作期與測試後製期同步進行，並與壓片廠、發行商等溝通完成，準備相關的製作、裝箱與發行事宜。

(5) 發行與檢討：遊戲在完成之後，並不代表開發流程的結束，尚有發行與檢討的一環。發行部分非本次主題，不多加闡述，此處將討論遊戲發行後的檢討工作。

遊戲產品檢討的時間，可定在遊戲發行後 2 個月進行，因為之前或許還有上市後才發現的瑕疵必須修正，或是根據市場反應狀況進行後續的行銷與宣傳配合。此時便可由企劃、美術、程式與音樂四組人員進行開會討論，針對此次遊戲開發中，工具程式的使用，與消費者在題材、操作、心得等方面的回應等，討論下一次進行開發時，應如何改進，使開發更加順利，更符合市場需求，並將此次遊戲開發的經驗彼此分享，至此整個遊戲發展流程才算完整結束。



1.3 研究動機及目的

1.3.1 研究動機

由於網路環境的進步，網路連線遊戲近年來已成為最炙手可熱的遊戲產業，其中又以由於線上角色扮演遊戲的市場最大。但開發一套遊戲所需的流程與時間仍傳統的單機遊戲相去不遠，從遊戲的企劃、分析、設計到實作與測試，仍需耗費大量的人力及時間。且每次開發時所使用的檔案格式常產生變化，使圖形、音效、人物等素材資源無法共用，常造成同樣的素材要重覆開發，形成人力上的浪費。若能有一套遊戲開發工具來協助遊戲設計者編輯劇情及管理素材，將能有效減少開發的時間，快速的掌握及佔有市場的先機。

此外，網路連線遊戲與單機版遊戲最大的不同之處，即在於遊戲中同時擁有許多玩家在進行遊戲，遊戲中所發生的事件必須即時的讓所有參與的玩家知道。玩家之間也可以彼此溝通交流，進而產生合作與對立的關係。此種玩家之間的互

動是傳統單機版遊戲所無法提供的，也是設計一套網路連線遊戲時所需注意的重點。

1.3.2 研究目的

本文的目標是設計一套視覺化方式開發的網路連線遊戲產生器，以解決在製作網路遊戲時，仍要受到傳統單機版遊戲開發流程上的困難。此外，鑑於「角色扮演」為當今線上連線遊戲最受歡迎的類型，且遊戲內玩家彼此的互動性非常高，對於企畫設計遊戲中的互動劇情事件上相當重要，故選擇以角色扮演遊戲為遊戲產生器之提供類型。

1.4 研究步驟

為了達到上述的目標，我們提出的研究步驟如下：

- (1)分析劇情在一個角色扮演遊戲中的重要性
- (2)分析傳統的遊戲劇情設計方式
- (3)提出傳統的劇情設計方式所常遇到的問題
- (4)比較以視覺化方式設計遊戲劇情與傳統方式的優劣之處
- (5)分析網路連線遊戲產生器的設計需求
- (6)研究現有單機版遊戲產生器的架構，並分析其在設計網路連線遊戲時的不足之處
- (8)為遊戲產生器加入新功能，使其可以用來設計網路連線遊戲
- (9)利用新的遊戲產生器實作出一套連線遊戲

1.5 章節說明

第一章，分析網路遊戲的市場，並提出撰寫本論文的動機、目的與研究步驟

第二章，比較傳統方法與視覺化方式設計遊戲劇情的差異，提出使用視覺化設計遊戲劇情的原因。

第三章，討論遊戲產生器的需求分析及系統架構

第四章，檢查現有遊戲產生器所提供的功能與不足之處，提出如何讓遊戲產生器可以支援網路遊戲的設計

第五章，介紹系統的設計與實作，描述如何來實作編輯系統的各個模組。

第六章，結論與未來發展方向

二、相關研究

本章將研究角色扮演遊戲中劇情的重要性，並分析傳統方式與視覺化方式設計遊戲劇情的比較。

2.1 角色扮演遊戲簡介

角色扮演遊戲(Role Playing Game)一般簡稱為RPG。角色扮演遊戲的歷史悠久，自個人電腦出現後，角色扮演遊戲一直居於遊戲市場中的主流地位。網路遊戲時代來臨後，角色扮演遊戲更幾乎成爲了當紅線上遊戲的代名詞。舉凡國內流行的線上遊戲「天堂」、「仙境傳說」、「石器時代」等皆爲線上型態的角色扮演遊戲。

角色扮演遊戲歷久不衰的原因，就在於其遊戲劇情內容的豐富性。遊戲一般會提供一個虛擬的幻想世界[15][16]，玩家在進行遊戲時，必須扮演遊戲裡頭的主角，遊歷各處城鎮村落收集情報，觸發種種的事件。經由打倒怪物練功升級，提昇自己的能力以完成劇情所交付的任務，最後到達魔王城堡消滅大魔王結束遊戲。在連線版的角色扮演遊戲中，玩家除了可以自己挑戰遊戲中的任務劇情外，也可以與其他玩家一同組隊，互相協調合作，共同挑戰更高難度的劇情，大大增加了角色扮演遊戲的遊戲性。

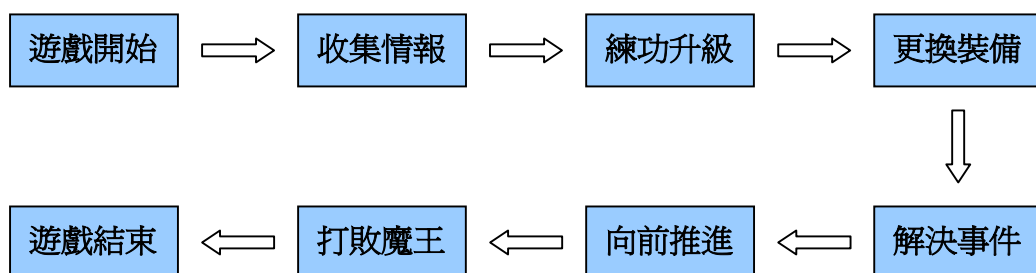


圖 3 角色扮演遊戲的標準流程

2.2 角色扮演遊戲的組成架構

以下首先分析一個角色扮演遊戲常見的組成架構，並觀察各個成分在遊戲中所佔的重要性，以瞭解一個角色扮演遊戲的核心。

2.2.1 玩家角色

一個遊戲不能沒有人玩。玩家在遊戲中扮演的角色隨時代表著主角的地位，負責觸發及推動遊戲的進行。網路連線遊戲與單機版遊戲一個最大不同之處，就在於網路遊戲提供了讓許多不同玩家同時進行遊戲的能力。對連線遊戲而言，遊戲中每個玩家都可以有各自扮演的不同角色，在不同的區域進行各自的冒險，而不會影響到其他玩家扮演角色的操作。玩家之間可以互相合作甚至競爭，使遊戲增加了許多變化，提高遊戲的耐玩度。每個網路連線所可以提供的同時進行遊戲人數，往往隨著遊戲類型而變。遊戲企畫人員必須在一開始決定遊戲種類時，同時確定開發的遊戲要容納的最大人數，以便程式設計師撰寫程式時可以決定合適的資料結構，以達到遊戲速度與品質的最佳化。

2.2.2 場景

幾乎所有的角色扮演遊戲都是由一張又一張的地圖所組合建構而成。場景代表著遊戲中的玩家所存在的虛擬世界，玩家透過穿梭各個不同的場景，與其他角色互動取得情報，打倒阻擋在路上敵手賺取金錢財物，再到商店中購買新的裝備、物品。遊戲的劇情、事件也同樣存在於場景之間，玩家往往必須在特定時間地點，取得指定的裝備或打倒特定敵人，以完成故事中所交付的任務。故企劃人員必須在規劃劇情大綱時，決定好遊戲中要使用的場景與場景中的角色、怪物，交由場景與編劇人員定義相關的地圖資料與劇情事件等細節。一個完整的場景由下列元素組成：

(1) 演員(Actor)

一個場景的外觀不僅僅是只有地形，還要有擺設於地形上的居民、怪物與物品道具等元素所形成，並可以讓玩家所使用的角色去接觸以觸發劇情。這些擺設在地圖上的人與物，我們將之通稱為「演員」。

(2) 劇情事件(Event)

每個演員都可以有自己的一套劇本，描述這個演員跟玩家產生接觸時所能引發的事件，讓玩家感受到遊戲的進行。舉例來說，玩家可以與一名村民談話後得到一件任務要求他去殺死一隻危害鄉里的盜賊取回寶物。在殺死名為盜賊的演員怪物後，玩家可以取得任務需要的寶物。在將寶物放回指定的地點後，玩家就會完成這件任務並得到獎賞。這些與任務

進行有關的事件描述，都可以記錄在演員的劇本中。

(3) 特效

每個角色扮演遊戲都是由一個故事來當作背景來發展而成，一個遊戲的進行可說是由玩家來帶領遊戲的主角進行這個虛擬的故事。因此在遊戲中如何將故事生動的表現出來，是遊戲企畫人員必須要費心去設計的。早期的角色扮演遊戲大多都是以靜態的文字與圖片來表達故事的進行。近年來多媒體及動畫的技術快速發展，因此大多數的遊戲都使用華麗的動畫及音效等特效來處理，讓玩家留下更深刻的印象。

2.2.3 素材資源

不論遊戲的玩家角色還是場景地形，以至於構成場景的演員、動畫、音效等效果，都必須由美工人員設計而成。沒有美工設計遊戲中使用的圖形與音效等素材資源，遊戲只不過是一堆文字與資料的集合體，無法吸引玩家去玩這個遊戲。一個遊戲有良好的圖形畫面，可以吸引玩家在看到這個遊戲第一眼就達到想要去玩的念頭。

2.2.4 程式模組

除了劇情與美工以外，一個遊戲也少不了遊戲的程式本身。沒有程式設計師負責做出一套能處理劇情與美工素材的程式來，則遊戲始終只是一個企畫，無法成為最終的產品。

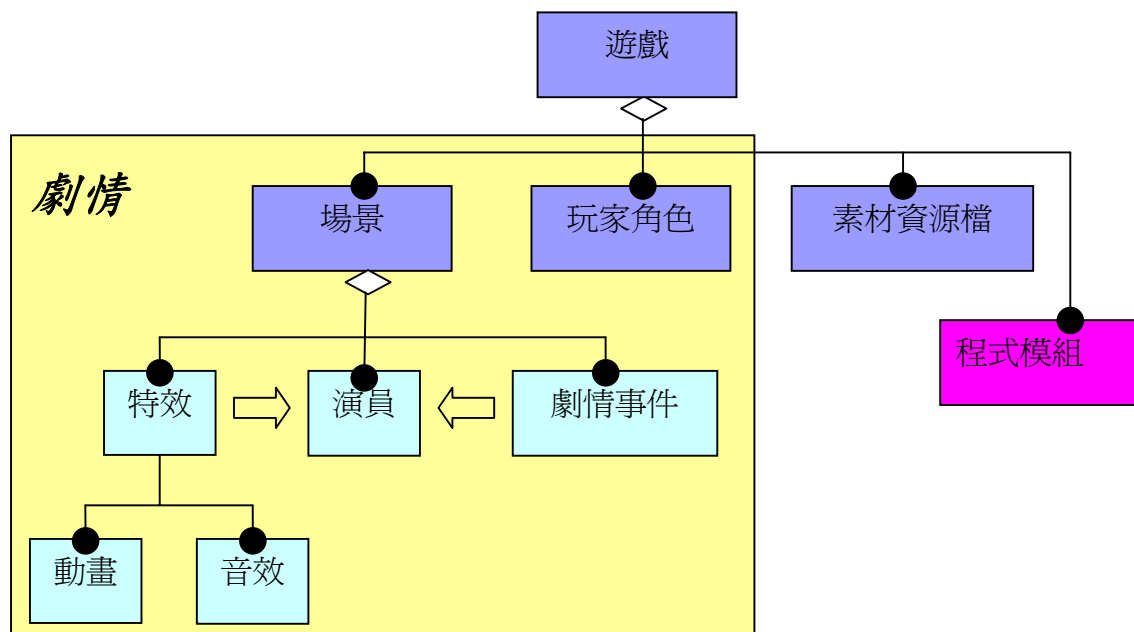


圖 4 角色扮演遊戲的基本組成架構

圖 4 是一套角色扮演遊戲中各個組成元素的架構圖。其中玩家角色與場景加起來，則組合成一個遊戲的劇情部分。劇情佔一個角色扮演遊戲的比重相當大，一個遊戲的劇情深度如何，是否能使玩家得到共鳴，往往就關係到遊戲的成功。

2.3 傳統方式的遊戲製作

本節首先觀察傳統方式開發遊戲的流程，與使用的劇情場景等文檔格式範例，從而分析其開發流程中常遇到的問題，並提出改進的方法。

2.3.1 遊戲劇情開發流程

以下是一般傳統上設計一個角色扮演遊戲劇情的標準流程：

1. 企畫人員調查市場需求，決定要製作什麼樣的遊戲，並提出遊戲腳本大綱的企畫書
2. 根據企畫提出的遊戲腳本大綱，分析遊戲中要用到的人物、場景等劇情元素，以及需要哪些圖形、音效、動畫等美工素材資源等需求，然後交由美工與編劇人員設計相關細節
3. 美工人員用繪圖、影像及錄音編輯工具，製作出圖形、聲音、影像、與 3D 模型等素材檔案。並根據劇情的需要製作遊戲中虛擬世界的各處地圖場景與角色
4. 程式設計師根據上述需求，定義出一套關於劇情設計用的描述語言，提供劇情編輯
5. 劇情編輯以程式設計師提供的描述語言語法撰寫完成劇情後，交由程式設計師放入遊戲內使用及測試

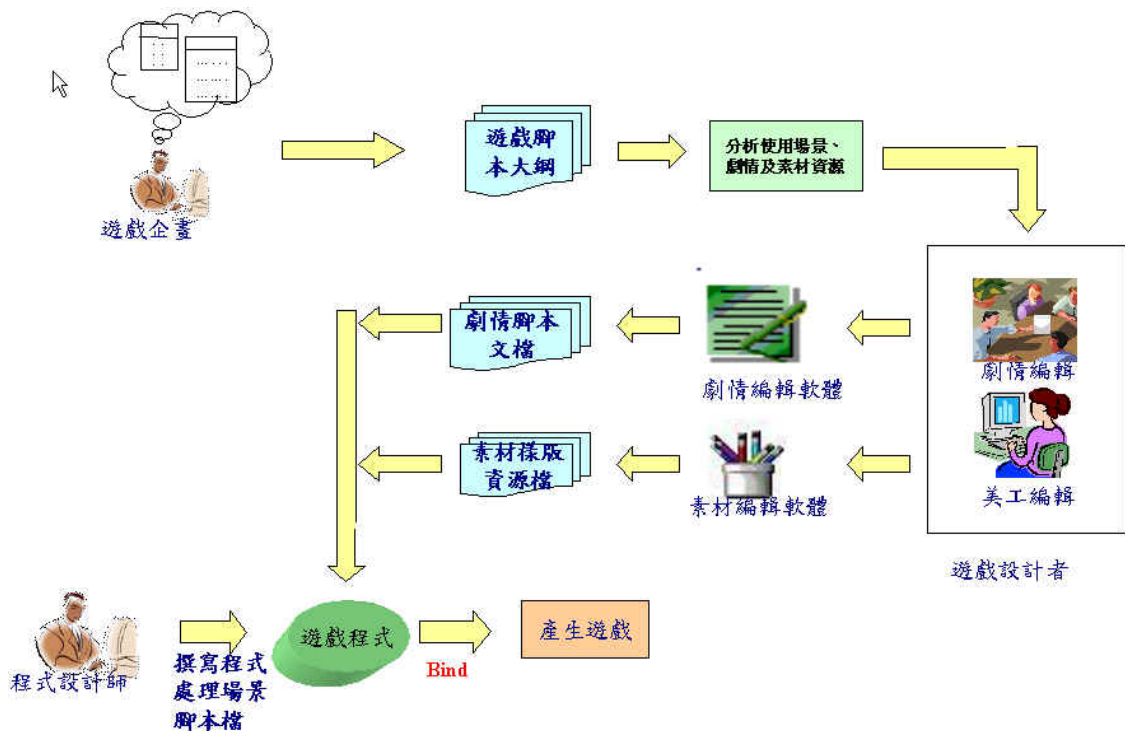


圖 5 傳統方式的遊戲開發流程

2.3.2 撰寫遊戲劇情的方法

在開發一個大型的角色扮演遊戲的過程中，對於遊戲的劇情資料的設定必定是相當繁複且修改頻繁。因此為了劇情設定的方便，也為了讓遊戲的製作更有彈性，由程式設計師與編劇人員互相定義出一套關於遊戲劇情的描述語言是必要的。程式設計師先在系統的分析階段根據企畫人員所提出的需求，設計出一套可以包含整個遊戲劇情設定的描述語言，再交給編劇人員依描述語言的格式來撰寫描述檔。描述語言可以描述在遊戲裡所使用到的 3D 動畫、背景音樂及音效撥出、遊戲事件的產生、遊戲的規則與人工智慧等等。下面介紹一個角色扮演遊戲所使用的地圖場景描述檔語言格式與範例。

2.3.2.1 劇情描述檔

劇情描述檔是由編劇人員設計。用來描述一個場景與擺放在上面的演員資料關係，演員資料包括演員本身的識別 ID、演員放在場景中的座標位置以及此演員包含的行動方式。以下是一個劇情描述檔的格式範例：

表 3 劇情描述檔格式

<pre>#BEGIN <X:number Y:number> <Actor:string> actiontrigger <行動方式> <行動方式> . . endtrigger #END</pre> <p>行動方式:</p> <table><tr><td>SAY <說話內容></td><td>: 說話</td></tr><tr><td>GIVE <物品ID></td><td>: 給玩家一個道具。ID表示此道具的編號</td></tr></table>	SAY <說話內容>	: 說話	GIVE <物品ID>	: 給玩家一個道具。ID表示此道具的編號
SAY <說話內容>	: 說話			
GIVE <物品ID>	: 給玩家一個道具。ID表示此道具的編號			

說明

- (1) #BEGIN、#END、actiontrigger、endtrigger、X、Y、Actor 皆為關鍵字 (keyword)
- (2) #BEGIN、#END 表示此場景檔案描述資料的開始與結束範圍
- (3) X、Y 用以描述此演員放在地圖上的座標數值
- (4) Actor 用以描述此演員的辨識 ID
- (5) actiontrigger、endtrigger 用以描述此演員所包含的事件。包括與玩家交談 (SAY)或送給玩家一個道具(GIVE)等

以下是用此格式所寫出來的劇情文檔範例：

<pre>#BEGIN <X:10,Y:10> <Actor:村民> actiontrigger SAY 你好嗎? GIVE ITEM_17 endtrigger #END</pre>
--

2.3.3 遊戲劇情開發時常遭遇的問題

傳統方式在使用劇情描述檔的方式開發遊戲時，常有下列的問題：

1. 程式設計師與劇情編輯之間的溝通問題

編劇人員編寫劇情時要記憶專用的語法去撰寫劇情描述檔，不夠直覺，更容易因文法格式錯誤而出錯。此外，劇情的描述由劇本編輯撰寫，但描述檔的格式及語法卻由程式設計師決定，導致編劇人員難以隨心所欲撰寫劇情。一旦程式設計師修改程式的資料結構或關鍵字(Keyword)，編劇人員就得因應修改寫法，造成開發上的困擾

2. 素材資源檔案重覆運用的問題

地圖、角色、事件、動畫等所用的素材資源檔案分散，缺乏統一性的管理，難以重覆使用，導致每次開發時必須一再重新設計相似的素材，造成人力與時間上的浪費。

2.4 視覺化方式的遊戲製作

鑑於以上問題，我們設計了一套組織化的遊戲產生器，來統一處理劇本的寫作、和資源的有效運用，以方便劇情設計的流程。這套遊戲產生器具有簡易的視覺化編輯介面，可以讓不懂程式設計的編劇人員透過這套工具來編輯遊戲。擁有這樣的一套遊戲編輯工具，遊戲劇情的內容可以讓編劇人員隨時修改，減輕了程式設計師的負擔，也降低了兩者之間的溝通問題。同時讓遊戲的發展速度得以提升，遊戲的品質也較能得到保證。

使用遊戲產生器來開發遊戲，可以解決下列的問題：

1. 解決程式設計師與編劇人員的溝通問題

在遊戲開發的系統分析階段，程式設計師即製作出一個可供劇本與場景編輯撰寫劇情用的編輯工具，則程式設計師和編劇人員可獨立、平行進行自己部分的開發

2. 簡化遊戲劇情的撰寫方式

透過視覺化的介面排設場景與劇情事件，使編劇人員不用去記憶繁瑣的描述語言文法，提供一個方便的編輯環境。

3. 素材資源的管理

提供一套素材資源庫管理與查詢的資料庫化，可提高各項元件的重覆使用結合。

2.4.1 視覺化方式遊戲劇情開發流程

以下是用視覺化遊戲產生器開發劇情的流程：

美工用影像及聲音編輯軟體製作遊戲素材後，編劇人員使用遊戲產生器將素材資源匯入，直接在地圖場景上用滑鼠拖拉的方式擺設演員，並可設定演員屬性與事件，最後用遊戲產生器的描述檔轉換功能自動產生程式設計師所要的描述檔格式。整個過程中劇情編輯只需要用滑鼠點選，不需去記憶劇情描述檔的語法。

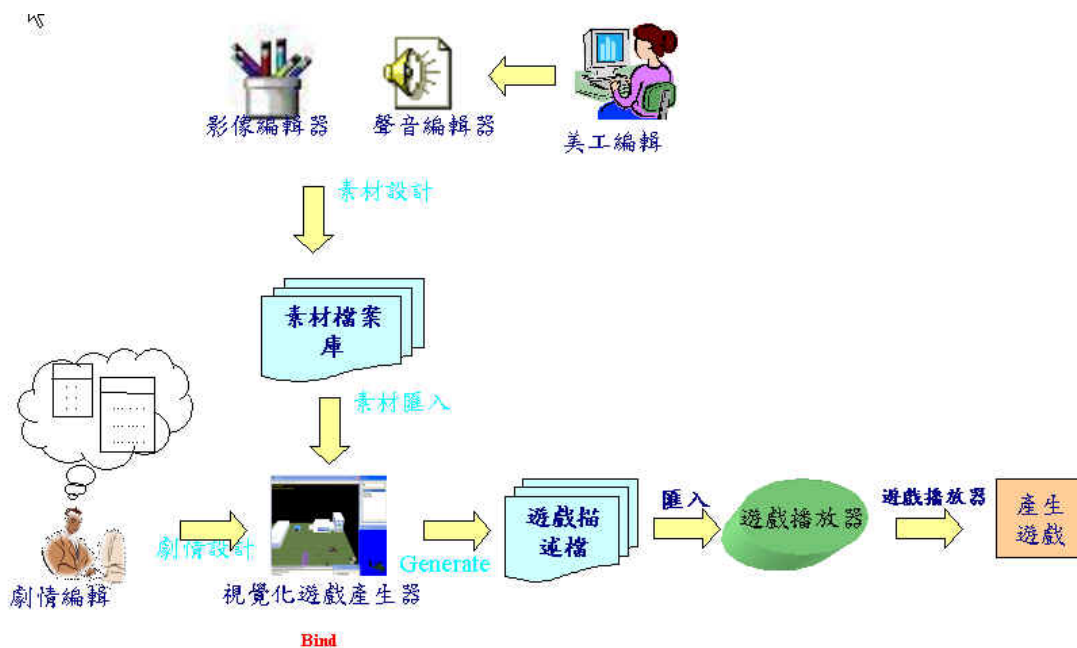


圖 6 視覺化方式的遊戲開發流程

使用視覺化方式[17][18][19][20]來設計劇情的一大優點，就在於視覺化的設計環境提供了一個所見即所得(What you see is what you get)的開發方式，編輯者在設計階段看到的地圖場景畫面就等於遊戲進行時的畫面。編輯者直接在畫面上點選演員後拖拉就可改變演員的擺設位置，並可以立刻看到改變後的畫面結果。

在編輯演員的劇情事件時，編輯者僅需從事件編輯畫面的指令列表中選擇要

加入的指令拉到事件視窗就可以為演員加入一段新的事件並編輯內容，遊戲產生器會自動產生符合遊戲中所要使用的劇情描述語法，透過遊戲播放器來形成我們所要開發的遊戲。在編輯畫面中則使用自然語言的方式顯示，讓編輯者可以直覺地了解每一段事件的內容。整個過程中編輯者完全不用去記憶程式設計師定義的描述語法，讓編輯者可以全心專注於劇情的設計之上。

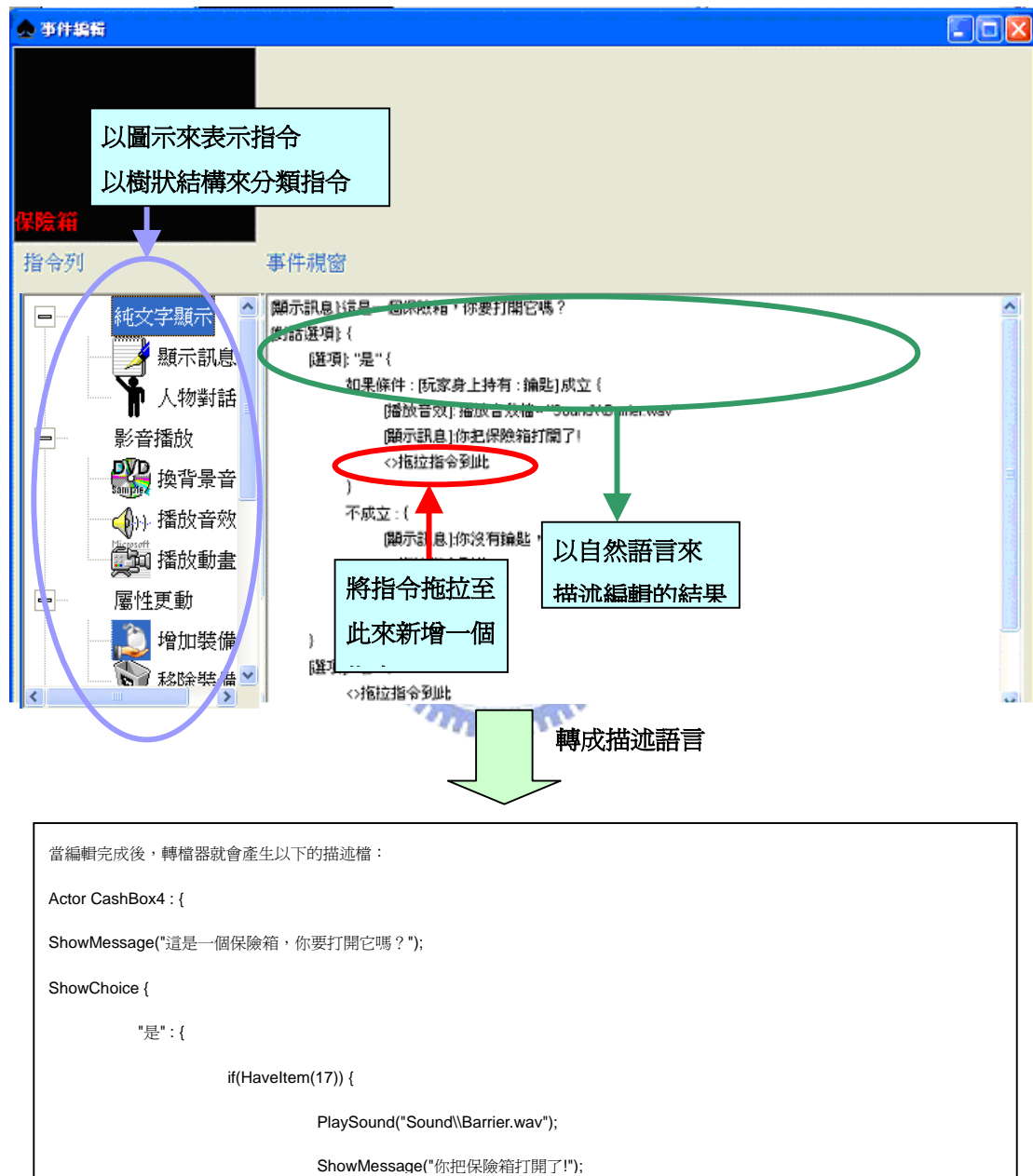


圖 7 以視覺化工具自動轉換描述語言

此外，視覺化遊戲產生器還包括一個遊戲播放器。編劇人員可以用這套遊戲編輯工具將自己做好的劇情企畫編出來，配合遊戲播放器就可以呈現隅測試所要開發的遊戲。在程式設計師實作遊戲播放器的同時，編劇人員可以平行的設定遊戲劇情，增進開發的速度。當遊戲完成並進行測試後，編劇人員可以針對遊戲需

要修改的地方立即進行修改，如此加快了遊戲開發的時間。

2.4.2 劇情開發方式分析

對於傳統方式與視覺化方式的遊戲劇情開發流程差異，可做一圖表來比較其優劣：

表 4 傳統方式與視覺化方式設計遊戲劇情的比較

	傳統方式	視覺化方式
開發流程	劇情的設計必須編劇人員和程式設計師不斷溝通才能產生	將劇情的設計全部交給編劇人員，減少雙方溝通上的問題，並減輕程式設計師的負擔。
劇情描述檔撰寫方式	編劇人員必須記憶固定的描述語言格式才能撰寫劇本描述檔，編輯時容易產生語法錯誤	編劇人員透過視覺化的介面直接從畫面上拖拉即可自動產生符合語法的劇本描述檔。不需去記憶描述語言的格式，亦不會有語法錯誤問題。
測試與修改	完成的劇本必須等程式設計師製作測試版才能測試	透過內建的遊戲播放器即可立即測試，縮短測試週期，並保證遊戲的品質
素材資源	各種素材檔案類型不一，難以分類與管理	提供預覽、查詢、匯入匯出等方式管理與重覆使用的便利性

三、遊戲系統需求分析

在第二章的討論中，我們看到了傳統方式開發遊戲的流程與方式所會產生的問題，與使用視覺化遊戲產生器來開發遊戲所帶來的好處。在本章中將會對視覺化遊戲產生器的系統需求作一分析，以了解一個視覺化遊戲產生器需要一些什麼功能來符合我們開發遊戲時的需要。首先我們將分析遊戲開發時的成員關係，並從每個不同成員的角度來看一個遊戲的組成要素與需求分析，從而導出視覺化遊戲產生器應有的成員架構。

3.1 遊戲開發的成員關係

首先我們來看一個遊戲開發過程中的相關人員組成。一個遊戲的開發團隊主要會由下列成員所組成：

- ◇ 玩家：遊戲完成後的最終使用者，也是需求的最先提出者
- ◇ 遊戲企畫：調查玩家想要的遊戲需求，將之轉成製作上的故事規格
- ◇ 場景編輯：根據遊戲大綱訂出使用到的地圖場景
- ◇ 劇本編輯：根據遊戲大綱訂出劇本內容
- ◇ 美工人員：提供場景及劇本所需要的圖形、音樂、動畫等資源素材
- ◇ 程式設計師：遊戲系統程式實作者
- ◇ 測試人員：遊戲完成後的第一手使用者

其中遊戲企畫、場景編輯、劇本編輯與美工人員為遊戲劇情的開發者，皆可視作遊戲設計者的一部份。程式設計師負責遊戲程式系統的開發，同時遊戲產生器也必須由程式設計師製作，因此其代表了系統製作者的角度。以下是遊戲開發團隊的成員關係圖：

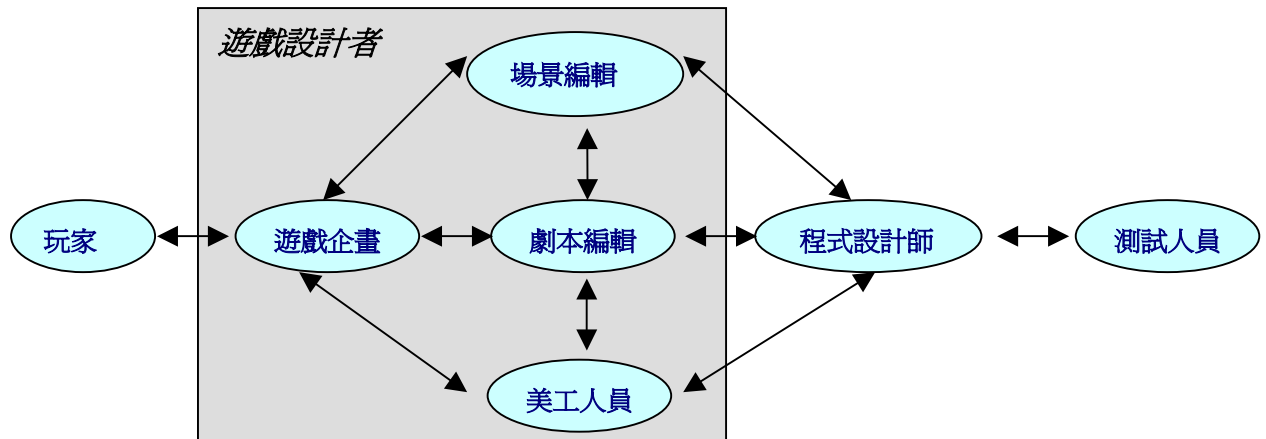
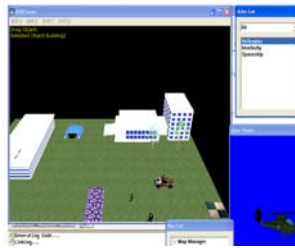


圖 8 遊戲開發過程中的成員關係圖

根據圖 8，我們將從玩家、遊戲設計者，以及遊戲產生器之系統製作者等下三個不同的角度來觀察一個遊戲所應有的組成要素與需求分析。得到遊戲的需求分析之後，便可以知道一個遊戲產生器所應該要支援的功能。

3.2 從玩家的角度

玩家是整個遊戲需求的發端，玩家的需求直接影響到一個角色扮演遊戲應該有的成分。對玩家來說，一個角色扮演遊戲的成分，不外乎由外在的畫面、音效、動畫等聲光效果，與內在的故事劇情、難易度等遊戲內涵所組成。



角色扮演遊戲

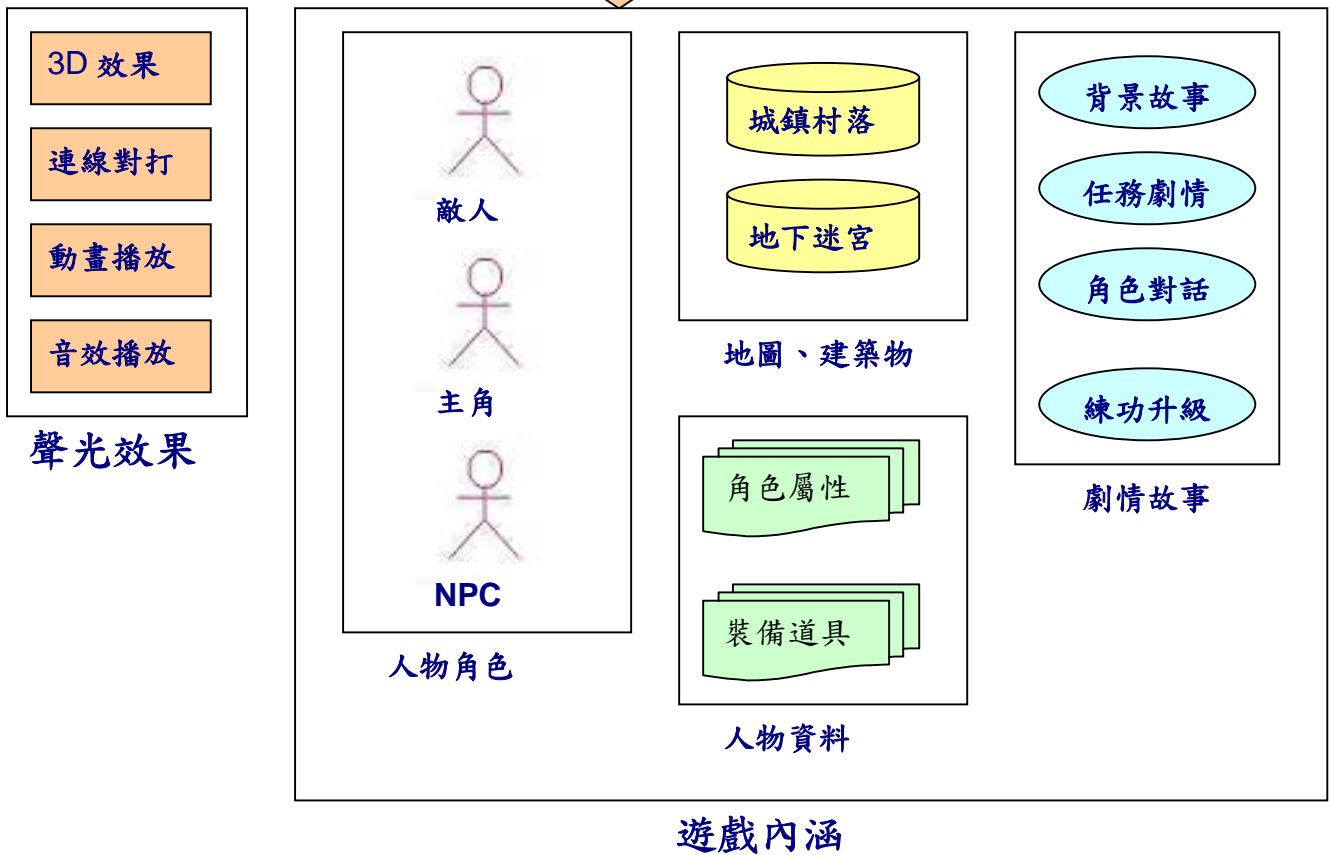
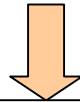


圖 9 從玩家角度來看一個角色扮演遊戲的組成要素

一、 聲光效果

有良好的外表包裝對於一個遊戲是相當重要的，一個畫面精緻的遊戲，遠比一個作畫品質差勁的遊戲更容易吸引玩家的注意力。聲光效果可以讓玩家在看到第一眼時就受到吸引而湧起想玩這遊戲的念頭。目前的遊戲幾乎都支援下列的聲光效果：

1. 3D 效果

隨著 3D 圖形技術的成熟及電腦顯示配備的進步，近年來的遊戲幾乎都大量運用 3D 效果來製作。市場上可見的角色扮演遊戲，超過一半以上都是採用 3D 的表現方式。原因即在於 3D 遊戲的展現方式，可以將畫面上參與演出的所有物件以 3D 的方式來呈現，不但比 2D 的呈現方式更能表現出場景畫面的深度與物體的遠近及相對大小，在呈現現實中存在的或是虛構、不存在的物體時都能比 2D 的方式顯得更為栩栩如生。且對於以 2D 的方式難以解釋或表現的觀念與物體，以 3D 方式來表現則來得更容易理解，因此能帶給使用者更實在的參與感和身歷其境的感覺。因此使用 3D 效果來開發遊戲已成為整個遊戲產業的趨勢，沒有 3D 效果的遊戲將難以滿足玩家的感官要求而被市場淘汰。

2. 連線對打

隨著網路的發達，網路連線遊戲大量興起。由於連線遊戲提供了玩家彼此之間的合作及對戰能力，讓玩家享受到有如現實中虛實交錯的人性爾虞我詐，大大提高了遊戲的刺激度和互動性，也讓玩者能夠完全地融入虛擬的角色當中。使得網路連線功能已成為了目前許多遊戲都標榜提供支援的種類。

3. 動畫播放

在遊戲中所使用的動畫主要有片頭展示動畫與劇情過場動畫兩類，用以描述故事起源或劇情的切換。動畫的播放可以讓玩家有如看一部電影般的欣賞一段故事的發展，使得遊戲的劇情呈現更加地生動。此外，在某些遊戲中對角色的動作如戰鬥攻擊、使用魔法等招式也使用動畫播放，使得角色的動作有更好的流暢感。

4. 音效播放

除了視覺上的效果外，音效播放也是遊戲聲光效果中不可欠缺的一環。音效的變化可以達到與遊戲畫面背景與舉情的結合，例如走在草地、沙灘、海邊、雪地等場景時都會切換不同的音效，當劇情事件發生時也會隨著故事發展變更背景音樂的風格，從而帶給玩家更多身歷其境的感覺。

二、 遊戲內涵

如果說遊戲的聲光效果是吸引玩家去玩這個遊戲的原因，那遊戲的內涵就是吸引一個玩家持續玩下去的動力了。一個空有外表沒有內涵的遊戲，玩家在經歷過初期聲光效果的震撼後，很快就會覺得食之無味。所謂的遊戲內涵，也就是讓玩家在玩遊戲的同時，能夠獲得啓發、學習到知識甚至能夠給玩家更多的思考空間。以一個角色扮演遊戲來說，最大的遊戲內涵就在於它的劇

情故事內容的變化。玩家對一段劇情事件所做的不同選擇，可能影響到玩家在接下來的故事中成為正義的英雄或惡名昭彰的壞蛋，而給予了玩家許多的思考與想像空間。一個遊戲的故事內涵，主要會由下列幾項要素組成：

1. 劇情故事

遊戲的劇情首先以故事的劇情背景為架構，規劃出一個故事發生的時代、場合與社會結構，而定義出一個虛擬的世界來。再在這個虛擬世界中，細分出相關的背景故事、劇情事件、人物對話等。此外，一個角色扮演遊戲往往能讓玩家所扮演的遊戲角色在故事中成長，而使用一套練功打怪升級的規則，使玩家可以逐步提升角色的實力，前往新的區域挑戰更高難度的任務。

2. 地圖、建築物

遊戲的虛擬世界必定有許多的地圖與建築物。這些地圖與建築物的組合可以分為讓玩家獲得情報資訊與裝備補給的城鎮村落，與探索冒險、練功打怪用的地下迷宮等地區。

3. 人物角色

不論是在遊戲中的哪一種場景地區，上面都會有一些人物角色。一個遊戲中包含的角色有玩家所扮演的主角自己、由電腦控制的非玩者角色(Non-Playing Character)、以及在地下迷宮深處讓玩家挑戰的敵人等。

4. 角色資料

如上所述，角色扮演遊戲的一大特性就在於玩家所扮演角色在故事中能力的成長。影響一個角色能力的重點，就在於角色本身的屬性資料與持有的道具裝備。尤其道具與裝備，在角色扮演遊戲中佔有重要的地位。

在玩家的屬性能力資料來說，基本上常分為玩家的生命力、法力、攻擊力、防禦力、經驗值、等級等數值。玩家在冒險中會遭遇敵人戰鬥，以本身的攻擊力進行攻擊，削減敵人的生命力。在受到敵人攻擊時以自己的防禦力抵擋對方的傷害，同時也可以消耗法力來施展魔法，回復自己的生命或打擊對手的生命力。當其中一方生命值降到0時則代表死亡，若是自己打倒對手，則可以獲得勝利的經驗值。累積經驗值到一定程度後就會提升等級，同時角色的生命力、法力、攻擊力、防禦力等屬性數值也會提升變得更強。

擁有多樣的道具與裝備的角色扮演遊戲，可以讓角色的屬性有許多的變化空間，而使遊戲的內容更為豐富且更吸引人。以道具的種類來說，通常可以將它分為幾個大類，首先是可以大略的分為人物用的裝備以及使用道具，接下來裝備有可以依功用的不同而再區分為武器、護甲、頭盔、盾牌等等，當角色穿上這些裝

備後，本身的攻擊力與防禦力就可以得到提升。至於在使用的道具中，也可以大略分為治療用道具、移動用道具、輔助型道具、以及角色扮演遊戲中過關所需的必要道具等等。

3.3 從遊戲設計者的角度

上一節我們站在玩家的角色看一個遊戲的組成要素後，接下來我們站在遊戲劇情、場景、與美工編輯等遊戲設計者的角度，來將這些遊戲組成要素分類及歸納，轉換成製作開發過程中的名詞。

- ◇ 聲光效果：舉凡 3D 效果、動畫、音效，皆屬於美工人員所設計的圖形與音效等素材資源(Resource)
- ◇ 遊戲內涵：遊戲的故事背景、劇情，與構成遊戲內虛擬世界的地圖場景和人物角色等，皆屬於遊戲企畫與編劇人員所設計的劇情要素。其中的
 1. 人物角色與角色資料等，可以歸納為編劇人員設計的演員(Actor)與演員的屬性(Actor Attribute)。
 2. 背景故事、任務劇情、人物對話等，即為編劇人員所設計的遊戲劇情腳本 (Script)。
 3. 村落與地下迷宮等建築物，為編劇人員所設計的地圖場景(Scene)

因此將玩家所看到的遊戲組成要素換成遊戲設計者的角度來看，也就是一套角色扮演遊戲要有場景、演員與劇情等遊戲內涵的劇情要素，以及提供遊戲外在聲光效果用的圖形與音效等素材資源。

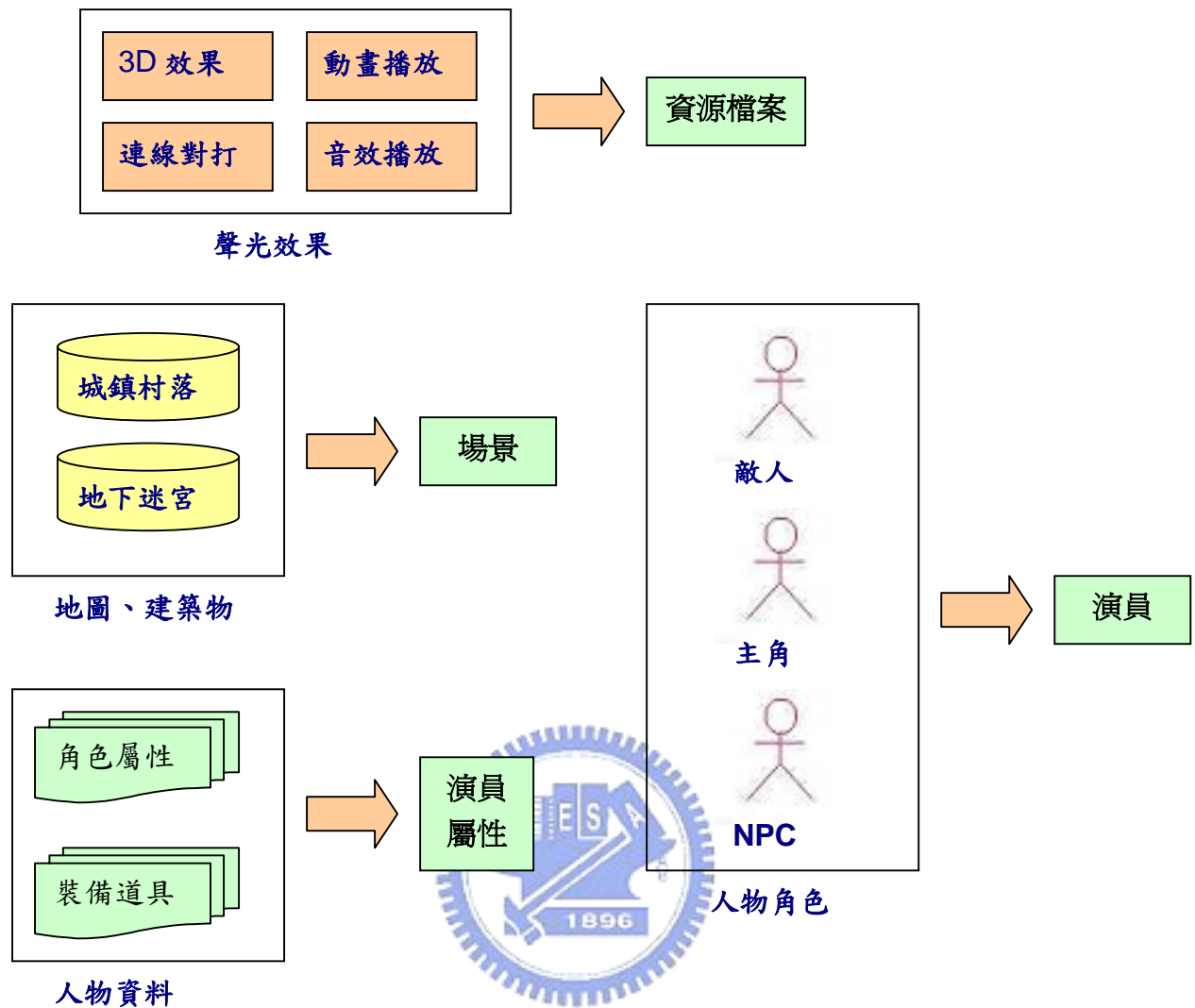


圖 10 從遊戲設計者角度來看一個角色扮演遊戲的組成要素

3.4 從系統製作者的角度

在得出一套設計一套遊戲時所需要有的組成要素後，最後以系統製作者，即程式設計師的角度，來看如何設計一個遊戲的程式系統，可以將上述的要素放入而組合成一套完整的遊戲架構。圖 11 是以程式設計師的角度來看的遊戲架構圖：

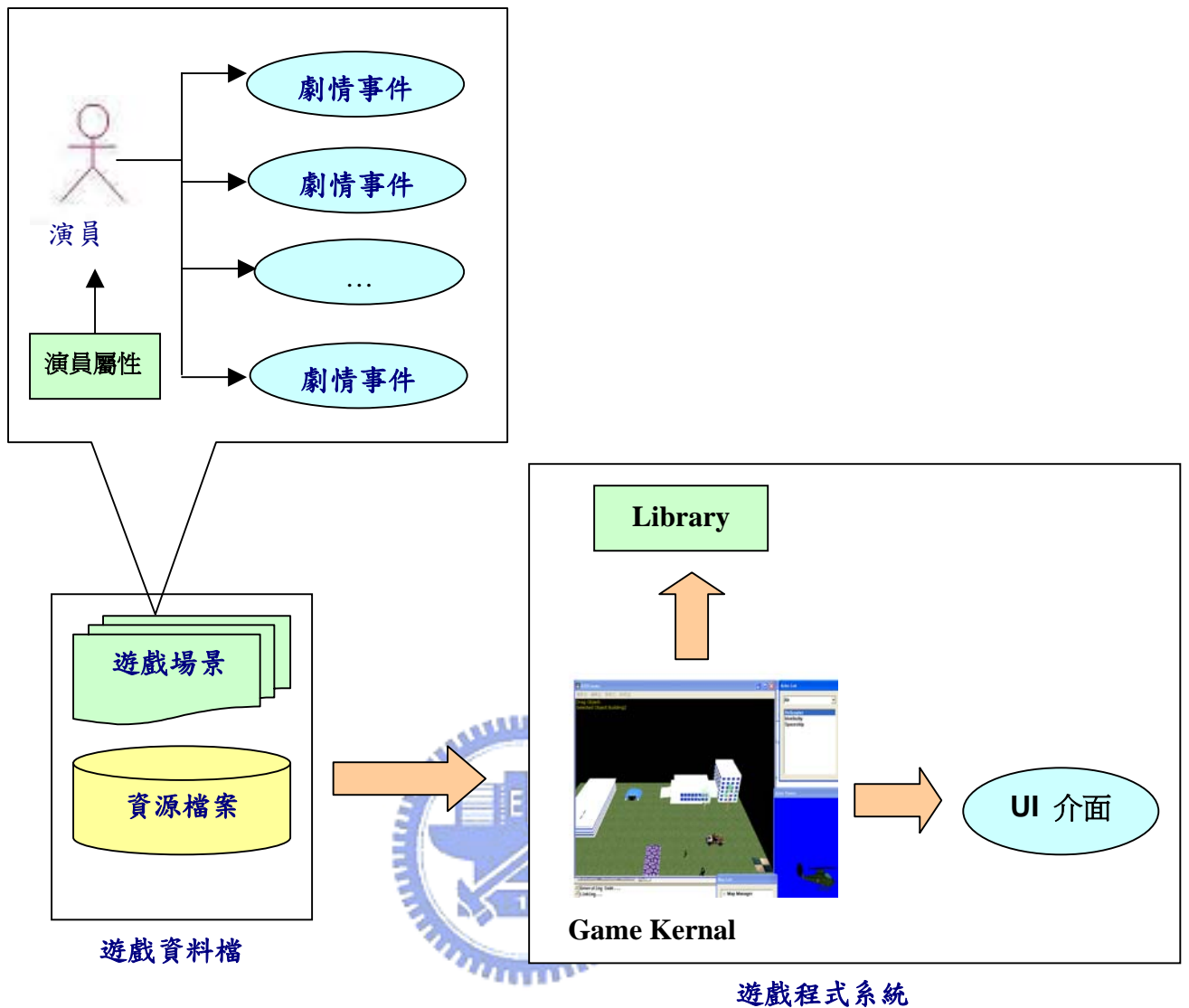


圖 11 從程式設計師的角度來看遊戲的成員關係圖

程式設計師負責製作遊戲的程式引擎[4][5]。一個遊戲引擎相當於遊戲的心臟。它們處理的工作包含圖形音效檔案的處理、劇本的解析、地圖與地形的設定、道具物品的使用等程式庫；以及和玩家溝通的遊戲輸入操縱方式(例如搖桿)與輸出畫面等使用者介面。程式所使用的遊戲資料檔，則來自於美工人員設計的素材資源檔案，以及編劇人員所製作的遊戲劇本檔案。在程式設計師的角度來看，遊戲的虛擬世界是由一個又一個的地圖場景檔案所組成，每個場景檔案內又包含了擺設在上面的人物與物品等遊戲演員，每個演員則各自擁有自己的屬性資料與觸發遊戲劇情用的事件腳本等。

3.5 遊戲系統的需求分析

結合 3.2、3.3、3.4 節對遊戲系統的需求分析結果，一個遊戲產生器的系統架構應該要能夠提供下列功能，才能製作出一套角色扮演遊戲：

(1) 產生場景

一套角色扮演遊戲由許多不同的場景所組成。在遊戲進行時必須讓玩家控制遊戲的主角穿梭遊戲中每個場景來進行故事。企畫人員必須要能利用遊戲產生器設定場景中的景觀、障礙物、地形，以及每個場景的大小與場景彼此之間的連結。

(2) 產生演員(人物、道具)

遊戲中的演員分為人物與物品兩種類型。企畫人員必須能透過遊戲產生器的編輯功能，設定遊戲中要出現的人物與物品等演員的外觀造型、屬性、以及與它們互動所產生的事件。

(3) 產生事件

在遊戲中由玩家所控制的遊戲主角跟遊戲中的其他演員進行的互動，來觸發劇情事件的發生。玩家可以藉由觸發不同的事件時的選擇，使遊戲以不同的路線來進行下去，而形成多線式的劇情。

(4) 連結素材資源

不論場景或演員所呈現給玩家感受到的視覺與音樂效果，都必須由美工提供的多媒體素材檔案來組成。此外由於多媒體動畫技術的快速進步，遊戲也需要生動的動畫搭配音效來呈現過場的劇情。

3.6 視覺化遊戲產生器的架構

根據上面對角色扮演遊戲系統的需求分析，我們可以得出一個用來產生角色扮演遊戲用的視覺化遊戲產生器，應該要包含下列架構及功能：

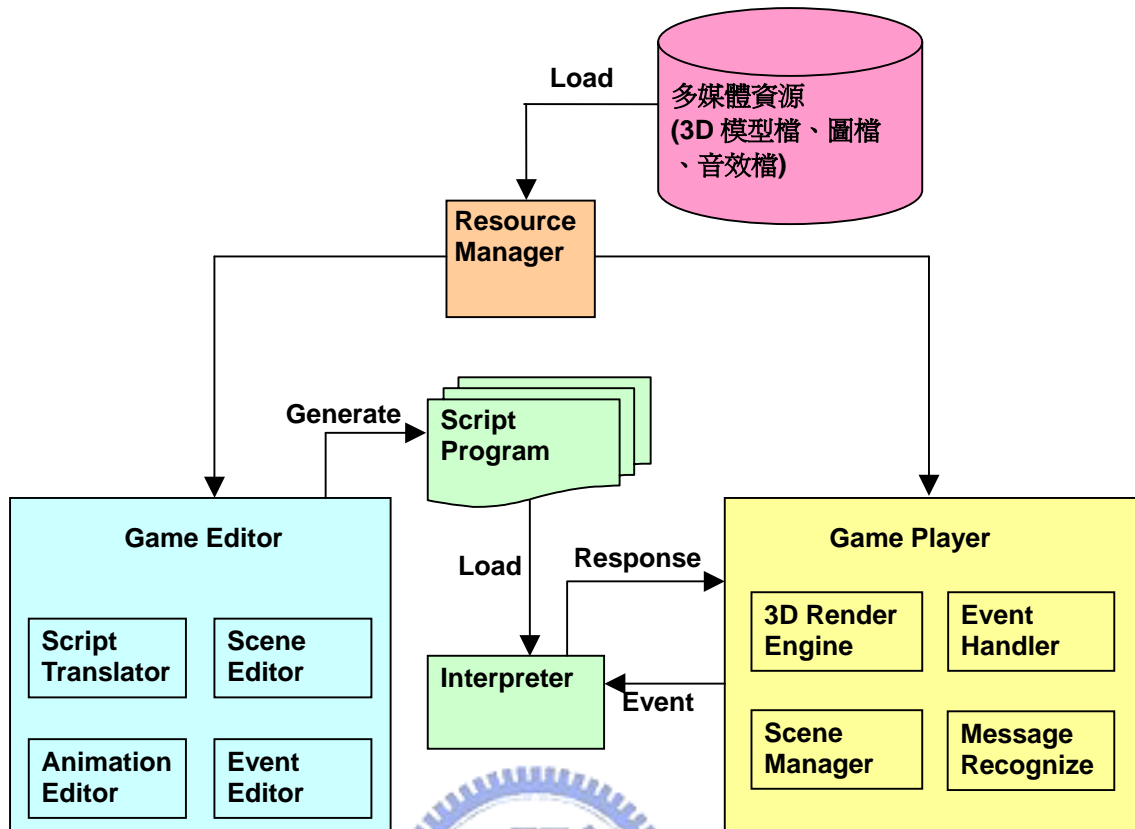


圖 12 視覺化遊戲產生器的架構圖

1. 遊戲編輯器：負責設計場景與演員關係，以及演員的事件劇情等。
 - (1) 場景編輯
 - 一、列出所有場景、切換編輯場景，新增、刪除場景。
 - 二、設定場景的基本資料：場景的大小、地面的貼圖、背景音樂。
 - 三、編輯視角的轉換：可以讓使用者自由移動視角，以不同的視角來作編輯。
 - (2) 演員編輯
 - 一、以滑鼠拖拉來設定場景中演員的位置。
 - 二、在場景中加入及刪除演員。
 - 三、設定演員的大小、角度。
 - (3) 事件編輯：對於演員事件的編輯，讓企畫人員不需去輸入描述語法就能編輯出事件。幫助企畫人員方便修改及檢視目前的劇情。
2. 資源檔匯入管理工具
 - (1) 管理美工人員所製作的多媒體素材。如圖檔、音效檔、3D 模型檔等。讓使用者可透過資源管理系統來使用這些資源進行編輯。
 - (2) 將這些資源分類，方便使用者來找到想要用的資源。
 - (3) 在同一個頁面同時展示這些資源，可以一次將同類型的所有資源列

出。

3. 劇本描述檔轉換工具

- (1) 能夠將以上的編輯結果轉成描述檔。
- (2) 將描述檔讀入編輯系統，還原回編輯成果。

4. 遊戲播放器

將遊戲設計者所編輯的遊戲，透過繪圖引擎及場景控制模組來顯示在畫面上。並且藉由輸入介面接受遊戲玩家的控制。此外，使用一個內建的解譯器(Interpreter)來解析與執行遊戲的劇情事件。圖 13 是以視覺化遊戲產生器來開發一套角色扮演遊戲的流程圖：

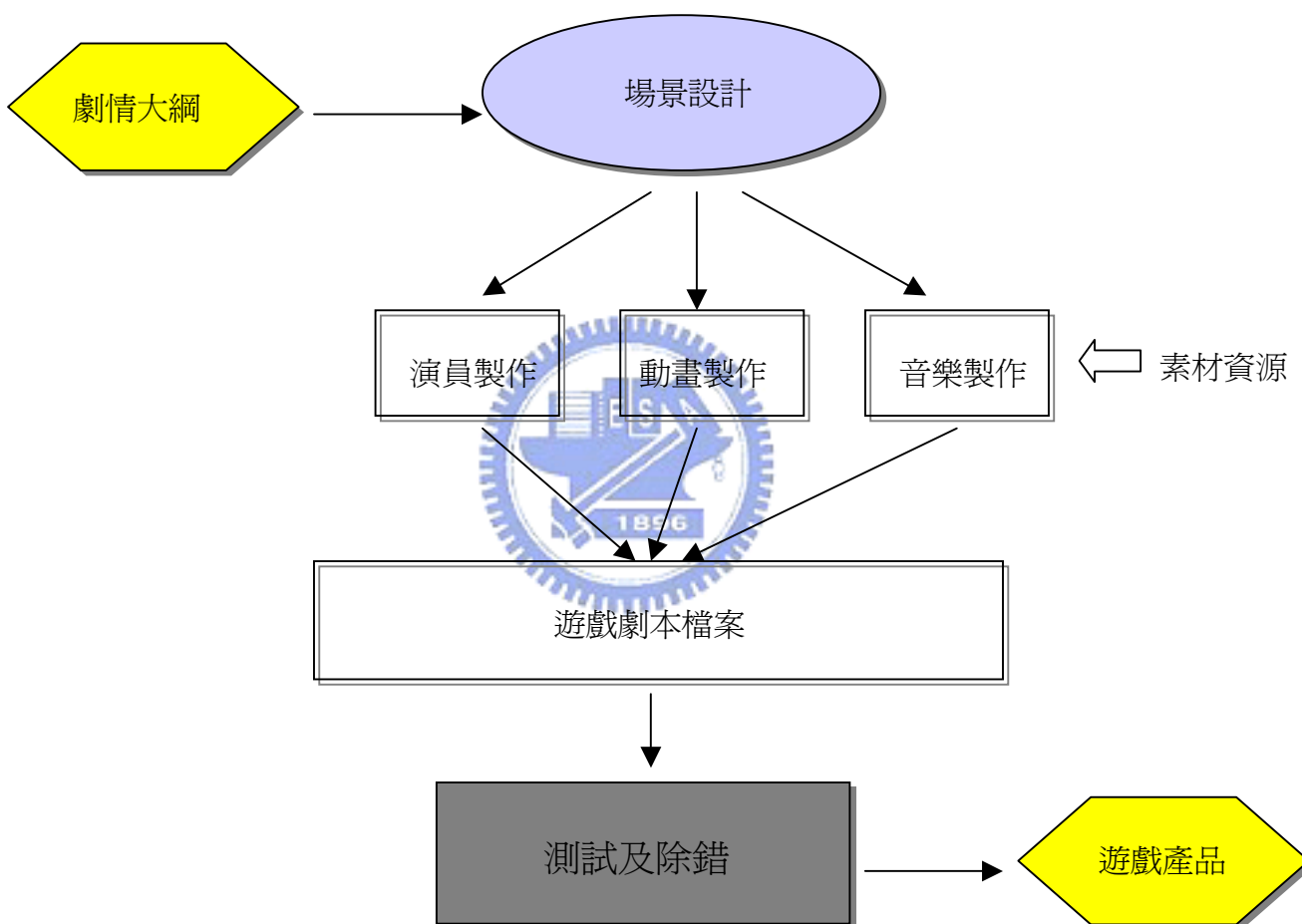


圖 13 以視覺化遊戲產生器開發遊戲的流程

在卓勇君的 3D 角色扮演遊戲編輯系統[21]中已對此視覺化遊戲產生器做了系統設計及規劃，唯尚未提供：

1. 編輯主角個人屬性與劇情事件的能力
2. 讓多位玩家同時連線上網一起進行遊戲互動的能力

故本研究將專注於：

1. 在遊戲產生器編輯主角屬性與劇情事件
2. 在遊戲播放器中使用編輯好的主角屬性與事件資料
3. 使用兩部以上電腦互相連線進行多人遊戲的研究與規劃



四、網路連線角色扮演遊戲產生器之設計

本章我們將剖析在3D角色扮演遊戲編輯系統的設計與製作[21]中之遊戲產生器的架構與產生遊戲的操作流程，藉以瞭解此工具所能達到的能力，並分析此遊戲產生器用在製作網路連線遊戲時所缺乏的功能，從而導出進階版遊戲產生器所需要增加的部分。

4.1 在3D角色扮演遊戲編輯系統中所設計製作的遊戲產生器的能力

接下來我們將分析此遊戲產生器的能力，觀察其中各模組的互動關係與使用方式，並找出此遊戲產生器的不足之處。圖 14 是現有的遊戲產生器的模組關係與資料流程圖(Data Flow)：

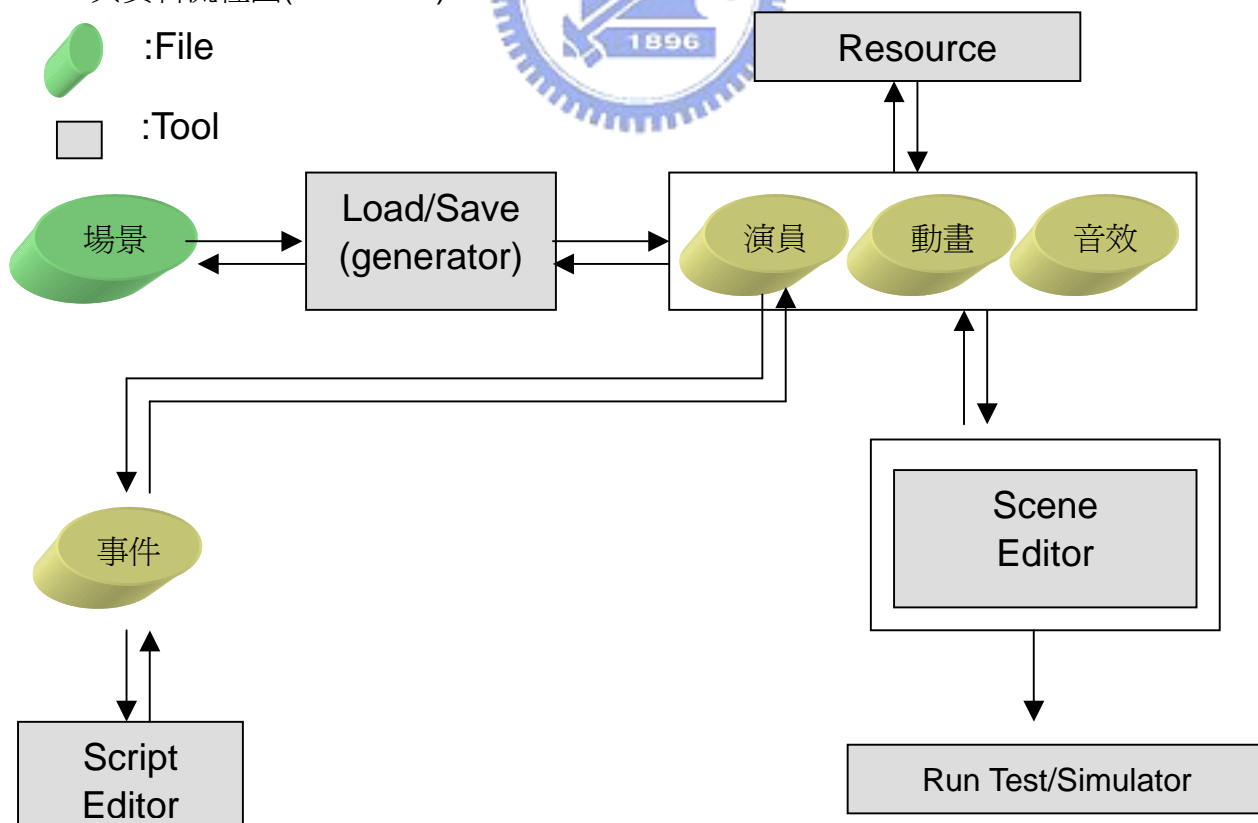


圖 14 在3D角色扮演遊戲產生器內模組的資料流程圖

遊戲設計者可開啓一張指定的場景地圖到編輯器中編輯內容，將素材資源庫中的演員、動畫與音效等加到場景中，並透過滑鼠以視覺化的方式來設定其位置與大小、播放方式等屬性內容。要編輯遊戲的劇情時，首先在編輯場景的主畫面中選擇想要編輯事件的演員，就可以透過事件編輯器編輯該演員相關的劇情事件。所有編輯的資料會自動轉換成遊戲播放器所要使用的描述語法儲存。遊戲設計者也可以隨時啓動遊戲播放器來看編輯好的場景效果。

4.1.1 編輯場景的操作流程

圖 15 是遊戲產生器的場景編輯畫面[21]，場景列表中列出了目前這個遊戲所有的遊戲場景。遊戲設計者可以在上面用滑鼠雙擊(Double Click)切換到其他場景進行編輯。此外也可以按方向鍵與上頁鍵(PageUp)、下頁鍵(PageDown)來移動編輯視角。如果想要編輯新場景可以在工具列上按新增場景的按鈕。



圖 15 場景編輯器的主畫面

進入新增場景的頁面後可以設定場景名稱、場景大小、以及場景地面的貼圖和背景音樂。按下 OK 按鈕後就可以進入編輯新場景的頁面來編輯。編輯後所產

生的場景即為描述語言中 Scene Definition Block 的內容。

新增場景：設定場景的長寬、地面貼圖、背景音樂

選擇場景的背景音樂

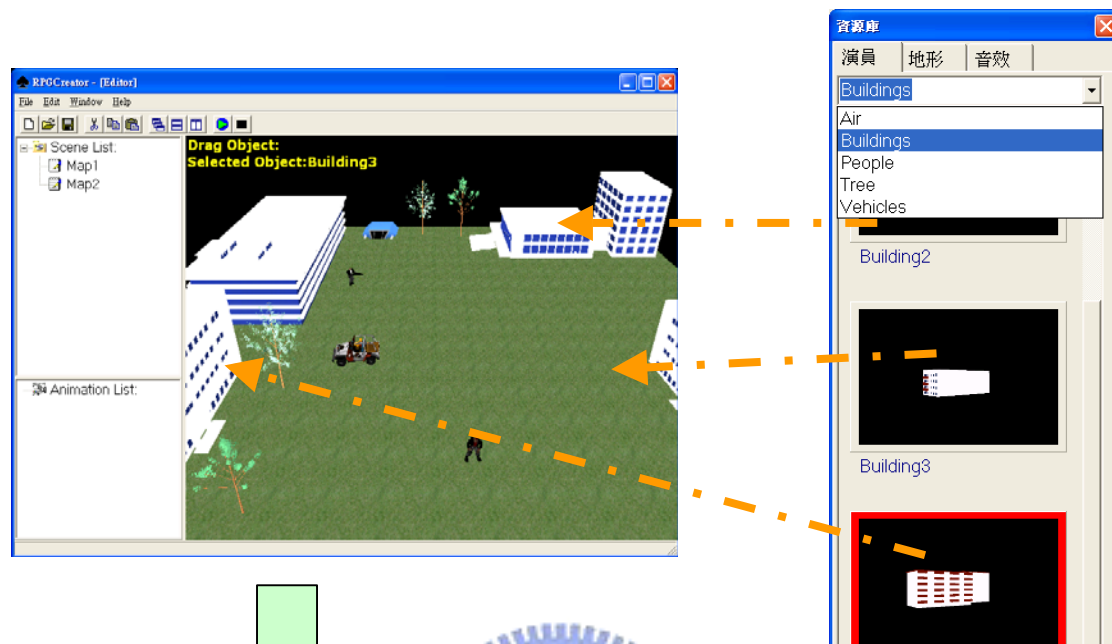
產生描述語法
Scene Definition Block

```
[SCENE_DEFINITION_BEGIN]
BGMusicPath = "Music\\Castle2.mid"
Height = 15
Width = 15
SceneID = "NewMap3"
Size = 15.000000
TextureFile = "GRAPHIC\\Texture\\Block.bmp"
[SCENE_DEFINITION_END]
```

圖 16 場景編輯器 -- 新增一個場景

新增一個場景後，接下來就可以從資源庫中選擇想要加入場景的演員。可選擇的演員包括人物、建築、家具、車子等類型。選擇演員後用滑鼠拖拉到場景畫

面內即可將此演員加到場景裡，並可接著用滑鼠來設定這些演員的擺設位置及大小。這些關於演員的資訊會自動轉換成劇情描述語法並被紀錄在描述檔中的 Actor Definition Block 內。



以滑鼠來設定演員的位置、大小等產生描述語法 Actor Definition Block

```
[ACTOR_DEFINITION_BEGIN]
Actor Building2 : {
  2DPicturePath = ""
  IModelFile = "GRAPHIC\MODELS\Buildings\B.x"
  ModelType = "XMODEL"
  Name = "第二大樓"
  Position = "(65.320680 0.000000 -339.796768)"
  Quaternion = "(0.000000 0.000000 0.000000 1.000000)"
  Size = "(9.200000 9.200000 9.200000)"
}

Actor Building3 : {
  2DPicturePath = ""
  ModelFile = "GRAPHIC\MODELS\Buildings\House.x"
  ModelType = "XMODEL"
  Name = "第一大樓"
  Position = "(53.794424 28.499978 -66.946600)"
  Quaternion = "(0.000000 0.999978 0.000000 -0.006135)"
  Size = "(1.900001 1.900001 1.900001)"
}

Actor Male1 : {
  2DPicturePath = "GRAPHIC\character\photo2.bmp"
  IniFile = "GRAPHIC\MODELS\Male\male.ini"
  ModelFile = "GRAPHIC\MODELS\Male\male.md2"
  ModelType = "MD2MODEL"
  Name = "總司令"
  Position = "(171.942320 9.599988 -51.198668)"
  Quaternion = "(0.000000 0.006136 0.000000 -0.999976)"
  Size = "(1.000001 1.000001 1.000001)"
  TextureFile = "GRAPHIC\MODELS\Male\colt.ppm"
}
[ACTOR_DEFINITION_END]
```

圖 17 場景編輯器 -- 設定演員的資訊

4.1.2 編輯演員事件的操作流程

首先在編輯場景的主畫面中點選想要編輯事件的演員，點選後在場景畫面中的 **Selected Object** 一欄會顯示出已選擇演員的名稱。此時按下右鍵，選擇選單中的『編輯事件』，即可進入編輯演員事件的畫面。

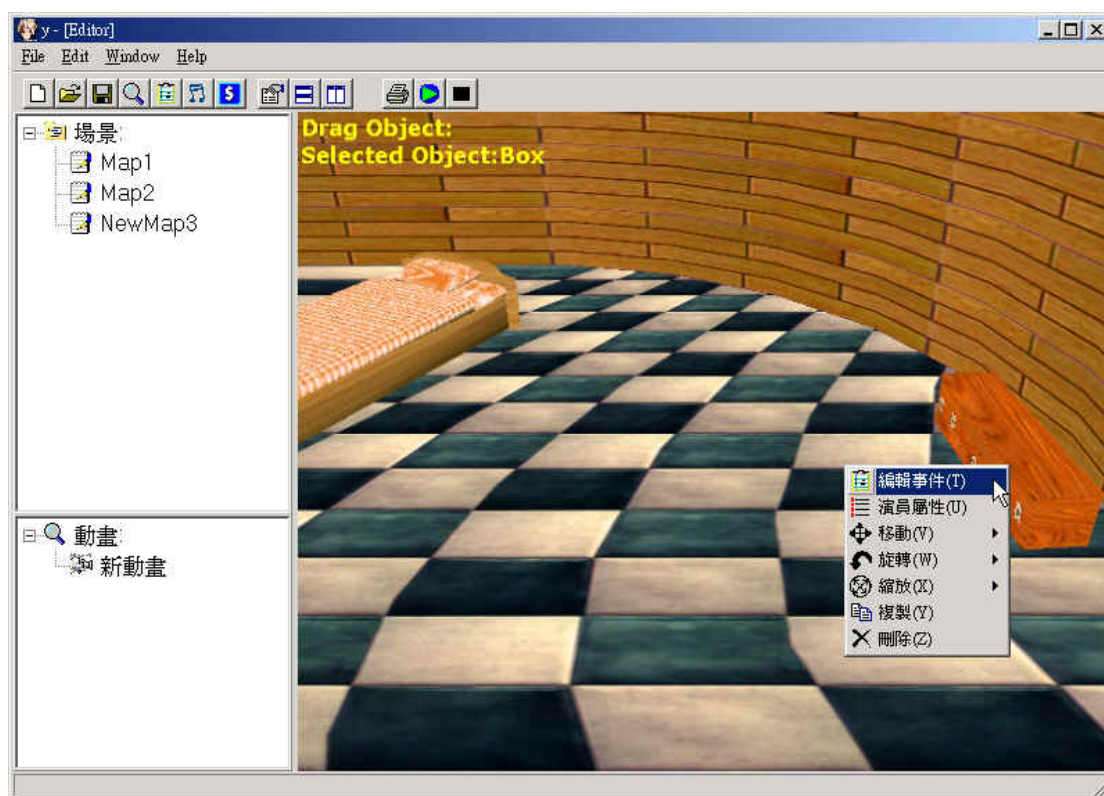
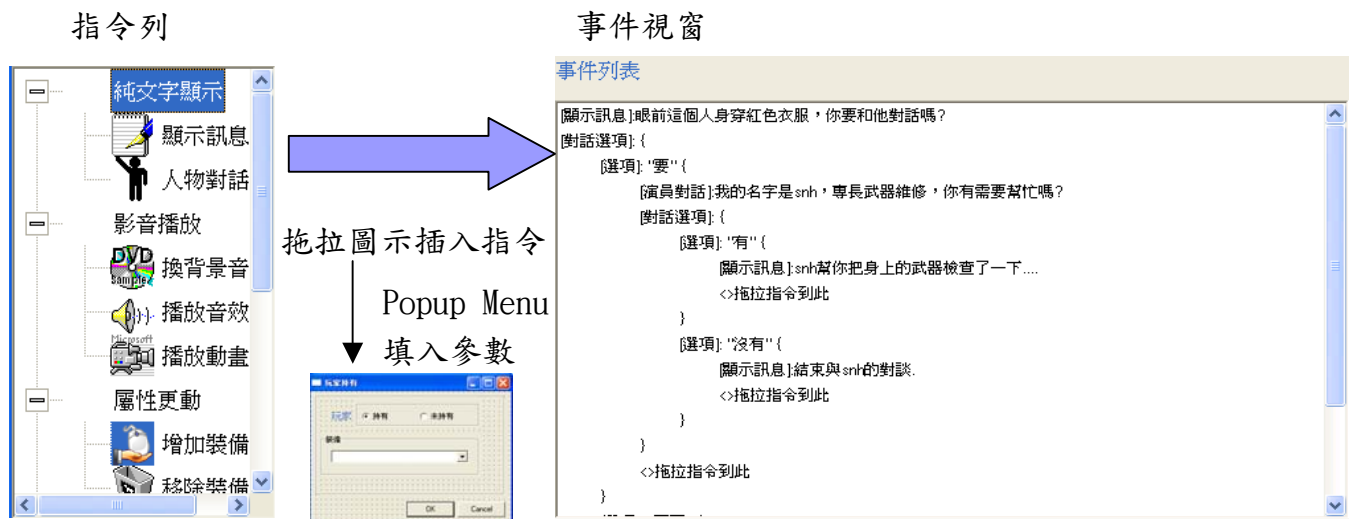


圖 18 事件編輯流程(1)

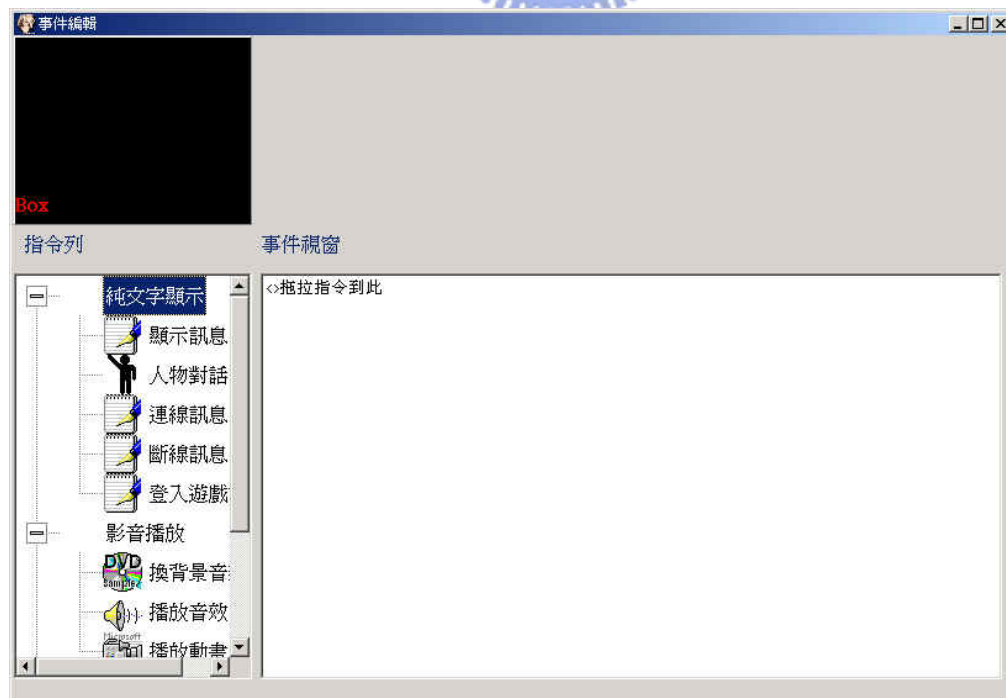
編輯演員事件畫面分為左邊的指令列與右邊的事件列表兩個視窗。事件列表為目前這個演員所擁有的事件。如果要新增一個指令，可以直接把左邊的指令圖示用滑鼠拖到事件列表中有『<>拖拉指令到此』的那一行，即可加入新的指令。若要刪除其中的一個指令，可在事件列表上按滑鼠右鍵選擇刪除指令。



下面我們舉出一個編輯事件的範例：

場景中有一個箱子，其演員名稱爲 **Box**。上面有紅藍兩個按鈕，藍色的按鈕可以打開這個箱子，而紅色的按鈕按了也無法打開。以下是編輯這個事件的流程展示。

(1) 在場景編輯畫面點選 **Box** 這個演員，按右鍵後在選單上選編輯事件，進入編輯事件的畫面。此時可以看到這個演員的事件列表中尚沒有任何指令。



(2) 這時候我們希望當事件發生前先顯示一個訊息“牆上有紅、藍兩個按鈕，你要按哪一個？”，來讓玩家知道發生了什麼事。要編輯這樣的事件我們可以將指令列的『顯示訊息』這個 Icon 拖拉到右邊事件視窗的“<>拖拉指令到此”。

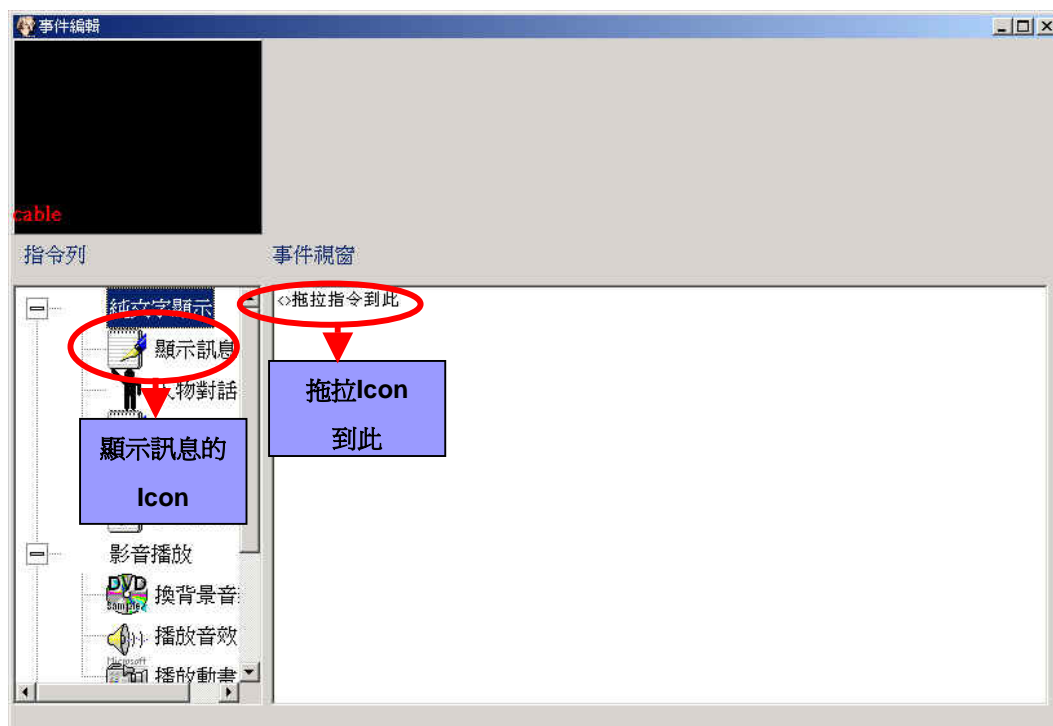


圖 21 編輯事件範例(2)

(3) 將 Icon 拖拉到該處後，這時畫面就會跳出一個輸入文字的選單，這時就可以把想要顯示的訊息輸入。按下 OK 的按鈕後，原本的事件視窗就會多出一個“[顯示訊息]”的指令，後面跟著這個指令的內容。

接下來我們希望這時候畫面跳出一個對話盒，上面有兩個選項，『紅』與『藍』。當玩家選『紅』時，則表示玩家要按下紅色的按鈕，若玩家選擇『藍』，則表示按下藍色的按鈕。

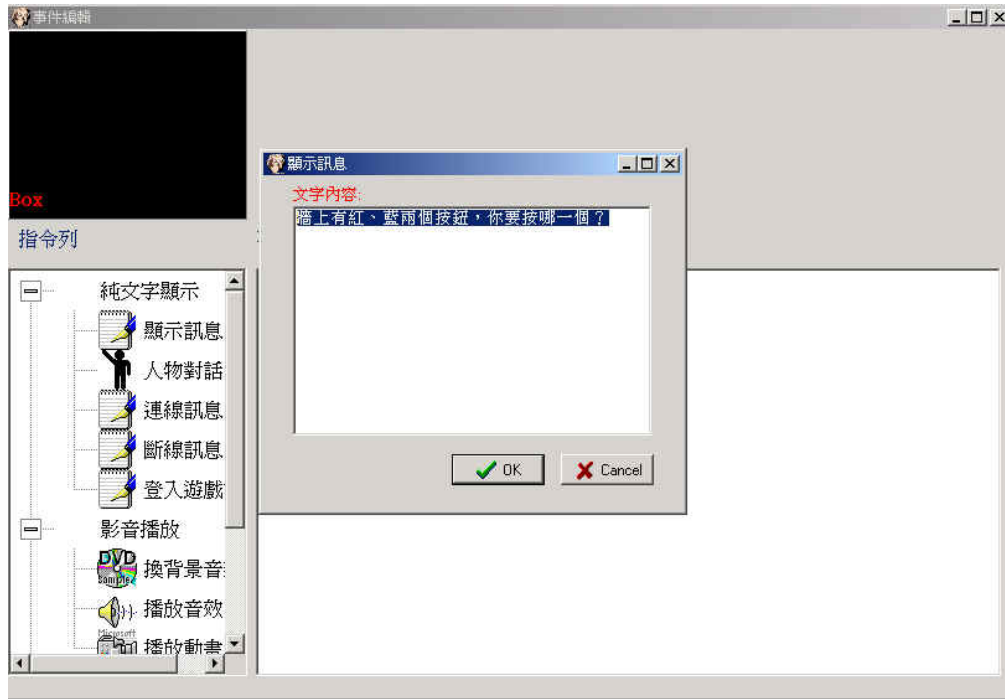


圖 22 編輯事件範例(3)

- (4) 這時我們要將『對話選單』這個 Icon 拖拉到事件視窗中，這時就會有輸入選項的選單跳出。由於我們需要『紅』與『藍』兩個選項，因此我們的選項個數選為 2，並且在選項 1 處輸入『紅』，選項 2 處輸入『藍』。

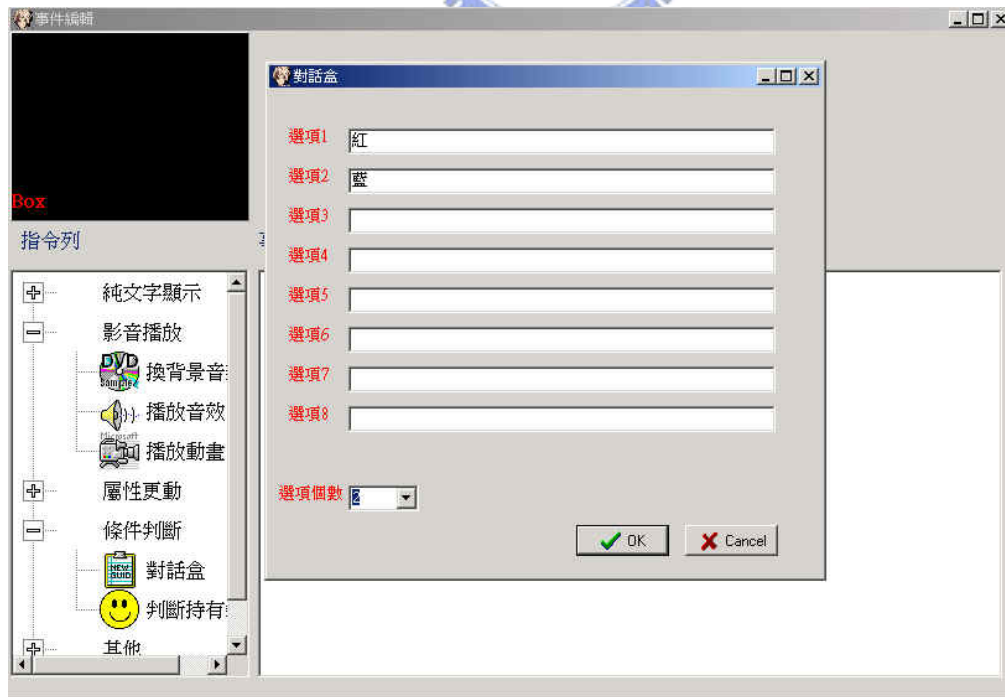


圖 23 編輯事件範例(4)

(5) 按下 OK 按鈕後，事件視窗會多出一個『對話選單』的指令以及多出兩個讓我們拖拉指令的區塊，分別代表玩家選擇『紅』與『藍』的敘述(statement)。



圖 24 編輯事件範例(5)

(5) 接下來我們就要來編輯當玩家選擇『紅』的選項後遊戲的反應。由於紅色的按鈕並無法打開箱子，因此我們將在這裡再插入一段顯示訊息，對玩家顯示此箱子無法打開。



圖 25 編輯事件範例(6)

- (7) 若玩家按下藍色的按鈕，則玩家可以打開箱子，並從裡面拿到一把自動步槍。為此我們將增加裝備的指令拉到事件視窗，畫面就會出現一個選擇裝備的選項列表，我們從列表中選擇自動步槍，然後按下 **OK** 按鈕。

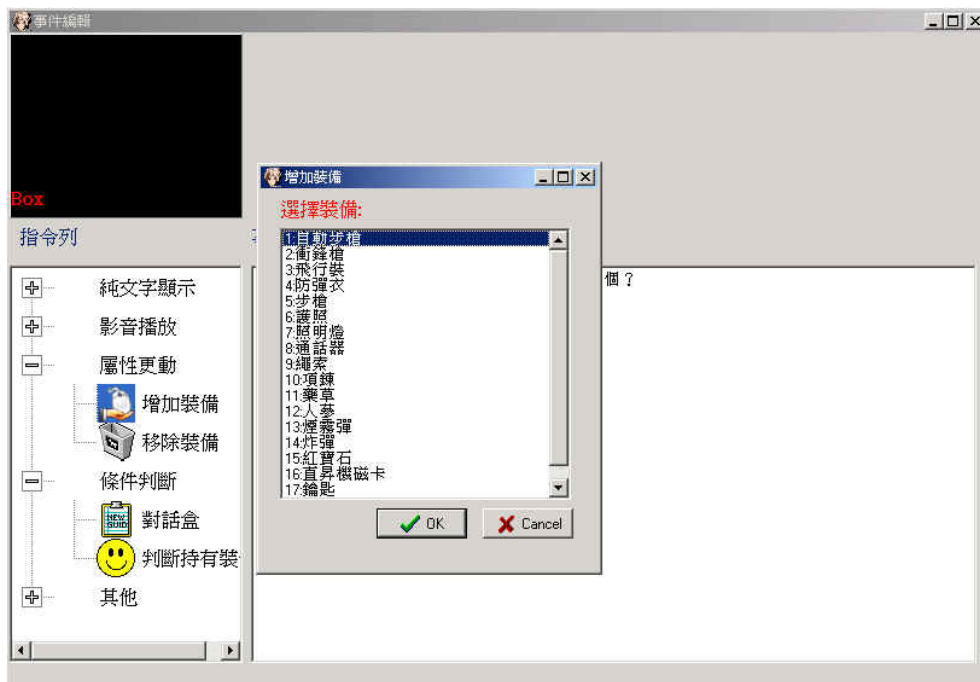


圖 26 編輯事件範例(7)

- (8) 最後，我們還希望在玩家成功打開箱子並拿到自動步槍時播放一段音效，以提醒玩家得到了一個新的物品。我們同樣將指令列表中的播放音效指令拖拉到事件視窗，即可設定完成。

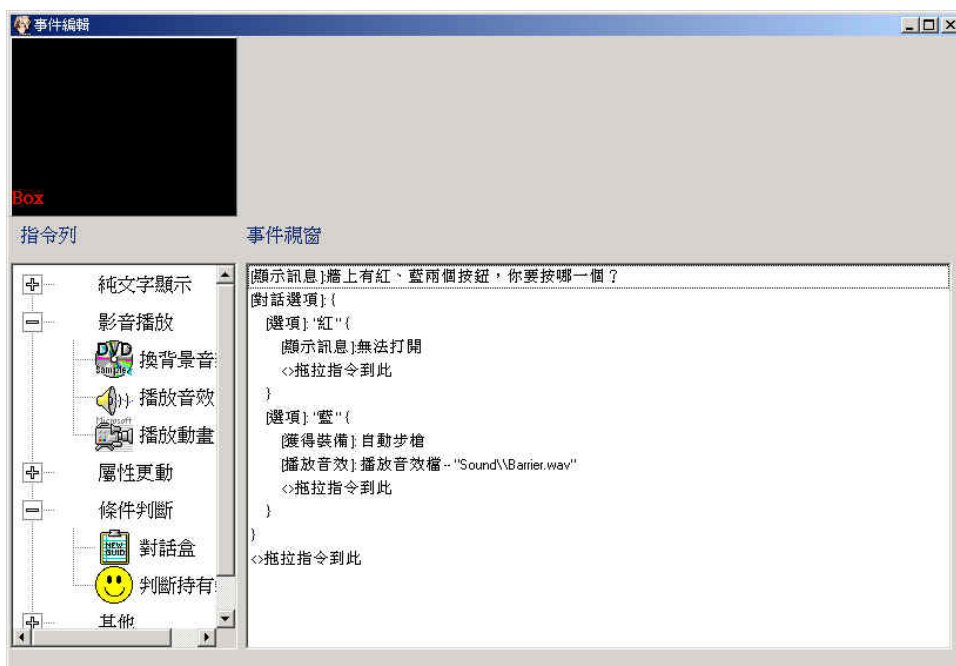


圖 27 編輯事件範例(8)

4.1.3 演員事件描述檔轉換

當編輯完成後，轉檔器就會自動產生以下的描述檔並存到此場景的描述檔記錄中：

```
Actor Box : {  
    ShowMessage("牆上有紅、藍兩個按鈕，你要按哪一個？");  
    ShowChoice {  
        "紅" : {  
            ShowMessage("無法打開");  
        }  
        "藍" : {  
            AddItem(1);  
            PlaySound("Sound\\Barrier.wav");  
        }  
    }  
}
```

透過這個視覺化的事件編輯器，遊戲設計者就可以利用拖拉圖示及利用表單來編輯遊戲的事件，而不需要撰寫描述語言就可以編輯出想要編的劇情內容，而且這些劇情內容都是以自然語言來描述，可以讓遊戲設計者對編輯後的內容一目了然。

4.2 在3D角色扮演遊戲編輯系統之遊戲

產生器中所待解決的問題

根據上節遊戲產生器的編輯與操作流程，可以歸納出用來製作網路連線遊戲時有下列幾個問題：

1. 缺乏網路連線功能

現有的遊戲產生器未提供網路連線模組，僅能產生單機版遊戲。玩家無法享受與朋友互相合作與對戰的樂趣。

2. 缺乏對多重玩家同時進行的支援

網路遊戲的特點就是同時提供多位玩家同時參與遊戲的樂趣。但現有的資料格式只能紀錄一個主角，即使有多個玩家要一起玩，也只能共有一個主角資料，無法達到多人遊戲彼此互動的能力。

3. 缺乏對玩家所扮演的主角(Leading Role)之屬性與事件的編輯

現有的事件編輯能力一定要連接到演員(Actor)上，因此僅能編輯遊戲內非玩家角色(NPC)與物品(Object)的相關事件，無法編輯玩家所扮演的主角專用事件，譬如當主角生命值小於一半時自動喝藥水，或當有其他玩家登入遊戲時讓線上玩家知道有誰進來了之類。

因此，要讓遊戲產生器可以產生網路連線遊戲，就需要加入網路連線支援的模組，以及提供多重主角同時進行的資料結構，並要可以編輯主角的個人屬性資料(如上述的生命力)與專用劇情事件，而製作出進階版的遊戲產生器。

4.3 加入網路連線支援

製作可產生網路連線遊戲的進階版遊戲產生器之第一步，就是要為系統的遊戲播放器加入網路連線模組，使我們編輯好的遊戲可以互相連線來玩。首先我們先分析目前常見的連線架構方式，並選擇一種最適合的加入原有的系統中。

4.3.1 網路連線遊戲的系統架構

系統架構的設計對於網路多人互動系統的整體表現有著直接的影響，不同的遊戲方式與需求就會使用不同的架構設計。目前主要分為 **Peer-to-Peer** 與 **Server-Client** 兩大架構

4.3.1.1 Peer-to-Peer 架構

Peer-to-Peer 是最簡單的一種架構。在此網路架構下，所有電腦沒有 **Server** 與 **Client** 之分，全都是一個 **Peer**，彼此以點對點的方式直接連到其他電腦，傳送訊息以直接連線的方式傳到對方手上，不需經過中介點層層轉送造成的延遲時間，因而連線速度較快。但每個 **Peer** 都必須對所有其他 **Peer** 建立連線處理，對頻寬的負擔較大，一般適合用於區域間的連線使用。**Peer-to-Peer** 最大的優點就是製作簡單，所有電腦用的都是同一套系統加上連線模組就可互相連接，系統也沒有專門處理資料用的訊息交換中心，由舊有的單機版遊戲直接升級就可成為支援網路連線架構的遊戲。

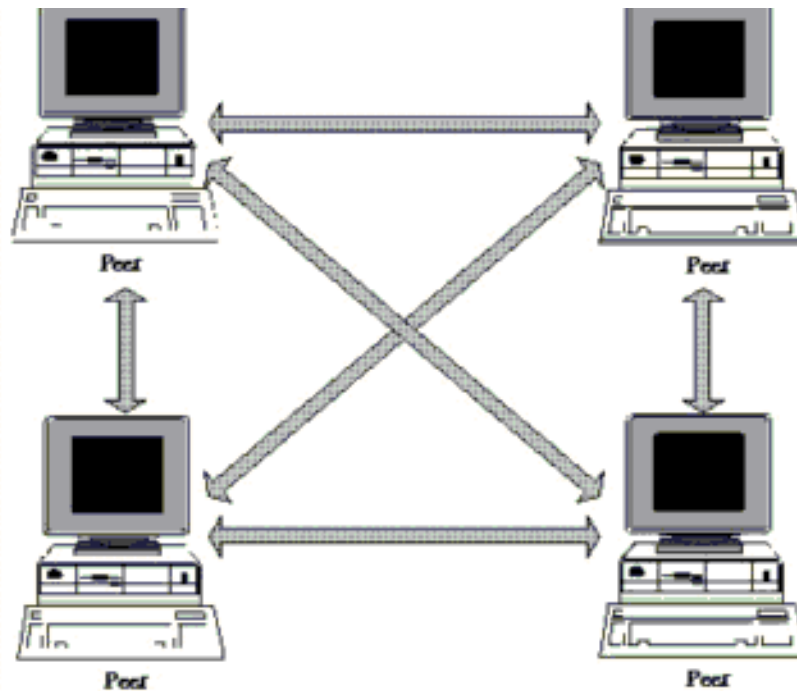


圖 28 Peer-to-Peer 架構

4.3.1.2 Server-Client 架構

Server-Client 架構由一或多台機器擔任 Server 作為連線中心，其他機器則作為 Client。在此架構下，所有資料的傳送都必須透過 Server 轉發，所有遊戲的資料運算也都在 Server 進行，再將結果送給其他 Client。此結構的優點在於對頻寬負荷較輕，且資料運算集中處理，對於遊戲執行的正確性與可靠度也較高。在 Server-Client 的結構下，玩家的電腦一般都作為 Client，而 Server 則放於遊戲發行業者的廠房內，兩邊系統所使用的程式往往完全不同。而玩家電腦的 Client 僅包含連線模組與畫面音效等介面而無資料處理，Server 的系統則主要著重於遊戲的資料運算與監控而不提供畫面音效等介面。玩家 Client 的遊戲無法單機執行，一定要連上 Server 才能進行遊戲，因此多用於大型的線上遊戲中。

此架構對遊戲產生器來說並不容易設計。因為視覺化遊戲產生器的原則就是在一套工具中以直覺化的方式直接製作遊戲地圖、加入演員、撰寫劇情等。但面臨了 Server 與 Client 兩端執行模組與資料運算方式的差異，製作出的地圖與劇本檔將難以同時符合兩端的需求，結果往往必須為 Server 與 Client 各自設計不同的資料檔，對於遊戲產生器的設計和更新上極為不便。

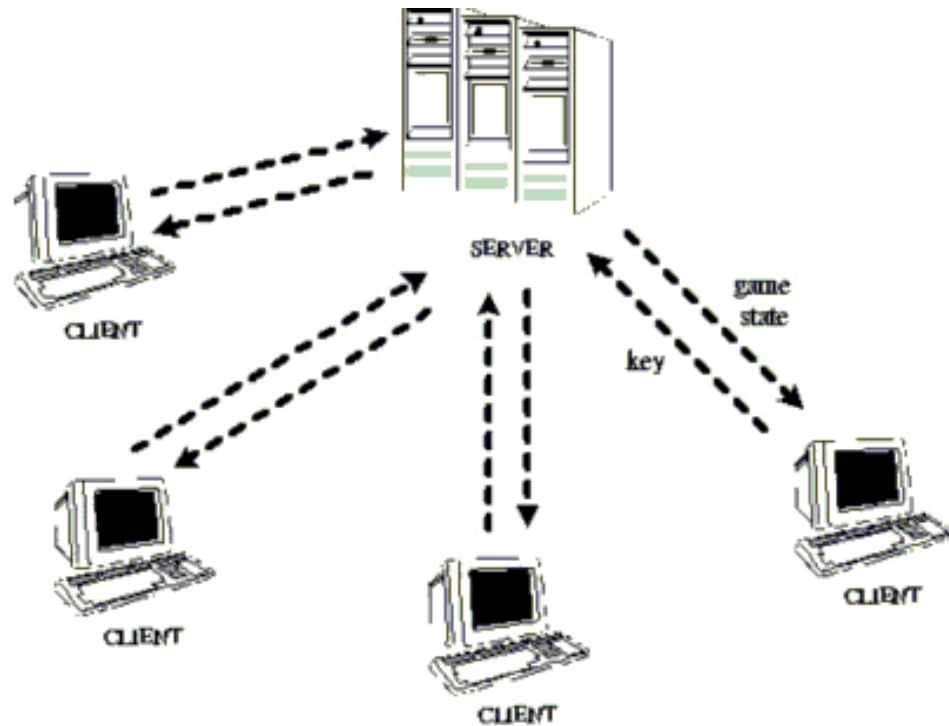


圖 29 Server-Client 架構

4.3.2 選擇的連線架構

在上述的兩種連線架構中，我們選擇使用 Peer-to-Peer 架構來製作進階版遊戲產生器。使用 Peer-to-Peer 架構的優點有：

1. 系統可直接由原本的單機版遊戲產生器擴充連線模組後升級而成，與原系統的相容性大且容易製作。
2. 製作的遊戲不用分 Server 版與 Client 版的分別，所有機器皆使用同一份版本即可，減輕遊戲設計者的負擔。
3. 可以產生網路連線遊戲或原本的單機版遊戲，增加遊戲開發時的選擇。同時，玩家玩遊戲時也可以選擇使用連線功能對打或不使用在單機上自己玩，不必受限於一定要連上 Server 才能進行遊戲。

4.4 加入多重玩家支援

加入網路連線模組後，再來就要將原本單一主角的資料結構，改為可提供多位玩家所操縱的主角同時進行遊戲的多重主角資料結構，才能正確反應出網路連

線遊戲提供多人同時進行的特性。首先我們來看在單機遊戲與連線遊戲中，所需要的玩家資料結構的差異性，再來看在網路連性遊戲中要如何正確處理多重玩家主角的資料結構，才可以讓所有玩家可以同時使用自己所操縱的主角進行遊戲，又不會影響到其他玩家所操縱的角色。

4.4.1 單機與連線環境中角色資料結構的差異

原本在遊戲產生器中，我們用演員(Actor)來表達場景內非玩家角色與物品，而主角則使用自己的資料結構。但在連線環境中，原有的資料結構已不足以表達連線遊戲中各玩家的統計狀態：

- ◇ 在連線遊戲中，有代表玩家本身所使用角色的主角(Leading Role)與其他玩家所使用的配角(Role)兩種使用者資料之分。兩者合起來為目前線上人數(Online User)。
- ◇ 線上人物資料隨玩家的登入(Login)與登出(Logout)而自動改變和更新的動態資料。與演員物件是在遊戲啟動時即產生的靜態資料類型不同。

基於上述原因，因此有必要增加一種新的角色(Role)資料結構來放置包含玩家自己與其他玩家等所有人物角色資料，以正確表達 Multi-User 的環境。

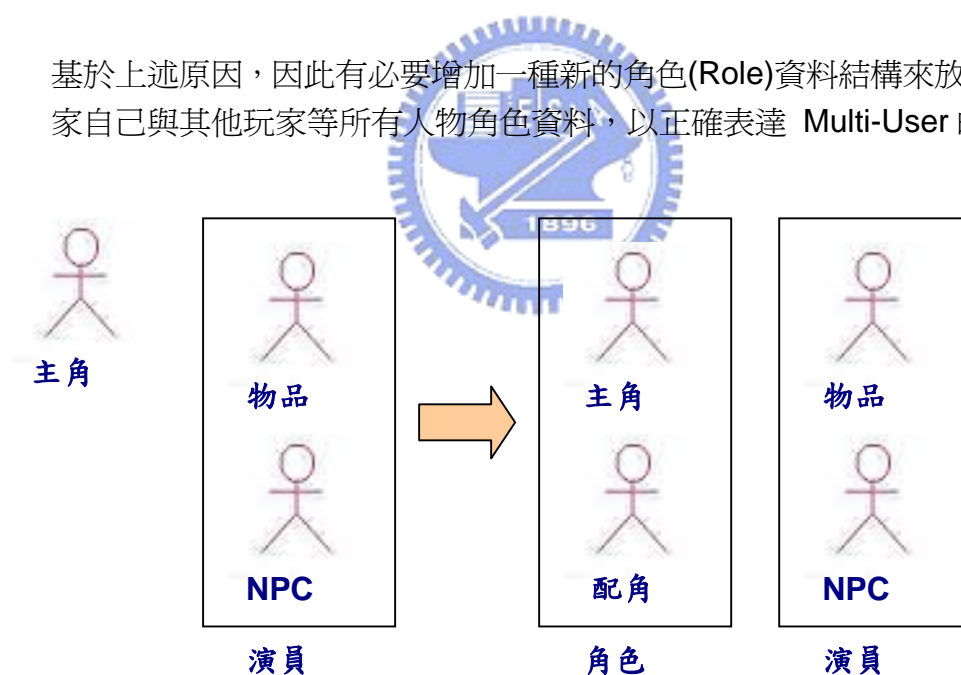


圖 30 網路遊戲處理玩家資料結構的改變

4.4.2 角色資料在連線遊戲中的特性

此角色資料應有以下特性：

1. 當玩家未連上網路時，所能看到的只有玩家所使用的主角。
2. 對於網路中每一台 Peer 機器來說，所保有的角色資料列表皆為一致且同

步更新。

3. 列表中的角色資料根據該角色的操縱玩家是否在線上(On-Line)而動態產生與刪除。

4. 在玩家所使用的 Peer 電腦中，玩家所控制的角色自動為主角接受玩家命令。其他玩家所控制的角色則為配角，玩家只能看到該角色的存在，但無法操縱。

5. 玩家可以對自己所控制的主角下指令控制，主角在接受玩家命令行動後，會將動作更新後的狀態資料送到其他所有 Peer 上更新該處的配角資料。同樣的，每部 Peer 也會接收其他 Peer 送來的配角資料並用以更新自己玩家列表內的配角資料。

圖 31 是在不同 Peer 中角色列表內主角與配角的變化。對 Peer A 這部電腦來說，玩家自己所操縱的角色即為主角，其他兩部電腦的角色自動成為配角 A 與配角 B。但同樣的角色列表在 Peer B 的機器中就變成了配角 A，而原本的配角 B 則變成該 Peer 上的主角。

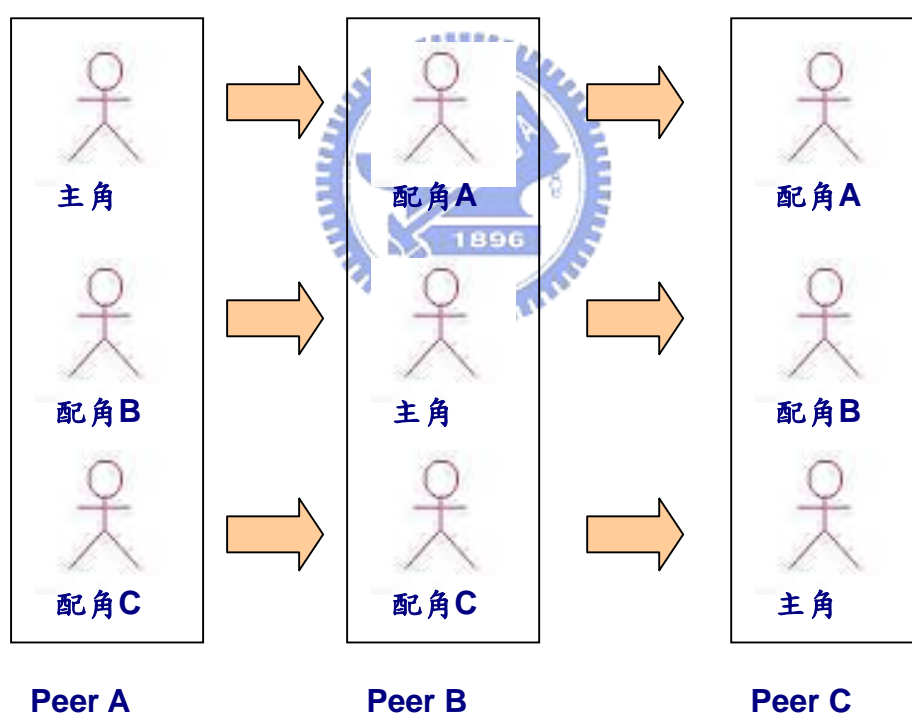


圖 31 在不同 Peer 中玩家所使用角色的關係

4.5 加入主角屬性與劇情編輯支援

在前兩節我們為系統加入了網路連線模組與多重玩家角色的支援，再來就是

要提供對主角專用的屬性資料與劇情事件編輯，將角色資料與連線功能結合起來，使角色資料在連線時自動透過網路傳送到其他 Peer 上，以互相交換線上玩家資料，成爲一個真正的網路連線遊戲。

4.5.1 編輯主角屬性資料

主角的屬性資料包含了此角色的一些資訊如 ID、座標、生命力、造型等，可以用來讓遊戲判斷每個角色目前的連線狀態是否正常。主角屬性資料可以分類 Private Data 與 Public Data 兩種：

Private Data：爲網路環境支援所必須之隱藏性資料。遊戲中的玩家無法看到這些資料，或雖能看到但無法修改其內容，如角色的名稱、連線 IP、角色造型等。這些資料主要用來讓系統辨識每個連線玩家的唯一性。這些資料主要由遊戲設計者在設計遊戲時定義。

Public Data：爲遊戲進行時所必須之公開性資料。遊戲中玩家可看到這些資料，並能依遊戲所提供的方式修改其內容，如：生命力(HP)、法力(MP)、經驗值(Exp)、等級(Level)等，玩家可以透過遊戲的進行而影響這些資料，如戰鬥後玩家可以獲得經驗值並提升等級。



4.5.2 編輯主角劇情資料

主角的劇情資料可以用來處理與主角相關的專用事件，以達到各玩家所控制的角色之間的互動。舉例來說，可以設定一個主角專用的事件，在有玩家登入或登出遊戲時，就可以收到訊息，以便隨時掌握線上人數的最新變化。

主角劇情描述資料格式與原有的演員格式完全相同，僅以一個名爲主角 (Role) 的特殊演員來代表。其組成方式爲使用如 OOP 方式的物件結構 [22][23][24]，使主角可同時具有屬性 (Attribute) 與行爲 (Method) 的特性。而對於行爲則使用事件觸發(Event Trigger)式處理方式，其由以下元件組成：

1. **Role**：代表玩家所扮演的主角(Leading Char)，以地圖中一個名爲 Role 的特殊演員來代表。主角可以擁有 0 ~ 多組專屬的事件。
2. **Trigger**：觸發事件的方式

3. **Condition Recognition**：判斷事件是否符合觸發條件的判斷處理。包括 if、else、case 等判斷語法
4. **Statement**：滿足條件後所進行的敘述動作。一個動作(statement)由 1 ~ 多個命令(command)所組成

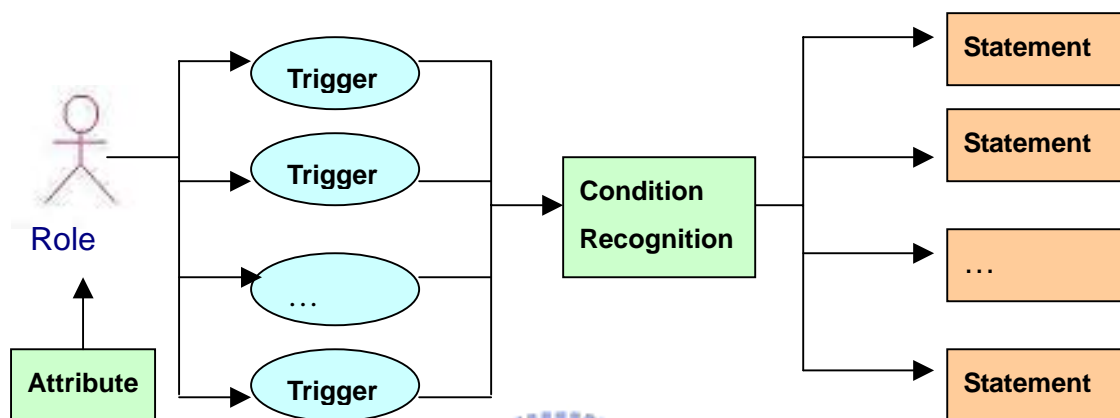


圖 32 主角劇情事件的組成結構

新增的主角專用事件類型有以下幾種：

1. 玩家登入網路：LoginMsg

用來描述當收到某個網路位址上的用戶連上了遊戲的網路時，要送給本地玩家的提醒訊息，訊息中會包含對方來自哪個網路位址。此時對方玩家已經完成網路連線設定，但尚未進入遊戲內建立自己的角色。

2. 玩家登出網路：LogoutMsg

用來描述當收到遊戲內某位其他遠端玩家所控制的角色切斷網路並退出遊戲時要送給本地玩家的提醒訊息，訊息內包含對方從哪個網路位址要退出遊戲。此時對方玩家已退出遊戲並切斷網路。

3. 玩家連入遊戲：ConnectMsg

用來描述當收到某位其他遠端玩家進入遊戲時要送給本地玩家的提醒訊

息，訊息中會包含對方來自哪個網路位址以及對方在遊戲中所使用的角色帳號。此時對方玩家已經完成網路連線設定，並已進入了遊戲內開始控制他的角色。

下面是一段應用範例：

```
Actor Role : {
```

```
LoginMsg("來自網路位址 %s 的玩家登入遊戲了");
```

```
LogoutMsg("來自網路位址 %s 的玩家登出遊戲了");
```

```
}
```

4.6 改善3D角色扮演遊戲編輯系統後之

遊戲產生器



綜合 4.3、4.4、4.5 節所提出遊戲產生器在產生網路版角色扮演遊戲時所需增加的功能，我們提出改善後的遊戲產生器架構，在遊戲編輯器中加入主角屬性資料與劇情事件的編輯能力，並在遊戲播放器中加入網路連線與多重玩家角色的支援。圖 33 是加入主角編輯功能後的新版遊戲產生器的模組關係與資料流程圖(Data Flow)：

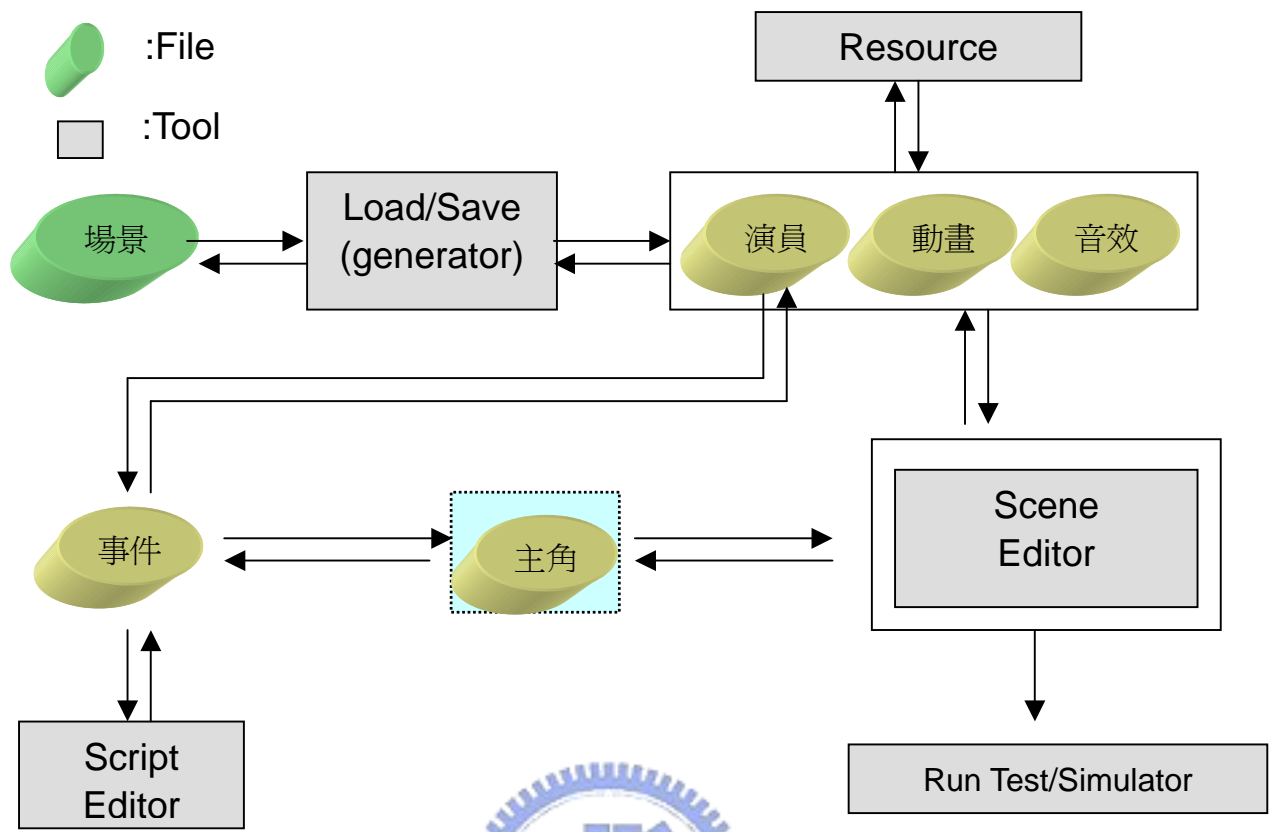


圖 33 改善後的遊戲產生器內模組的資料流程圖

遊戲設計者除了可在原有的場景中編輯演員、動畫與音效等屬性內容與劇情事件外，還可以編輯場景中主角的個人屬性資料與專用劇情事件，編輯好的資料同樣會透過遊戲播放器自動轉換成所要使用的描述語法儲存。同時，遊戲設計者可以透過遊戲播放器連上網路來測試編輯好的結果。當啟動遊戲播放器連上網路後，便會加入存在網路上的遊戲虛擬世界與其他玩家產生互動，玩家所控制的角色也會自動成為遊戲中存在的線上玩家之一。

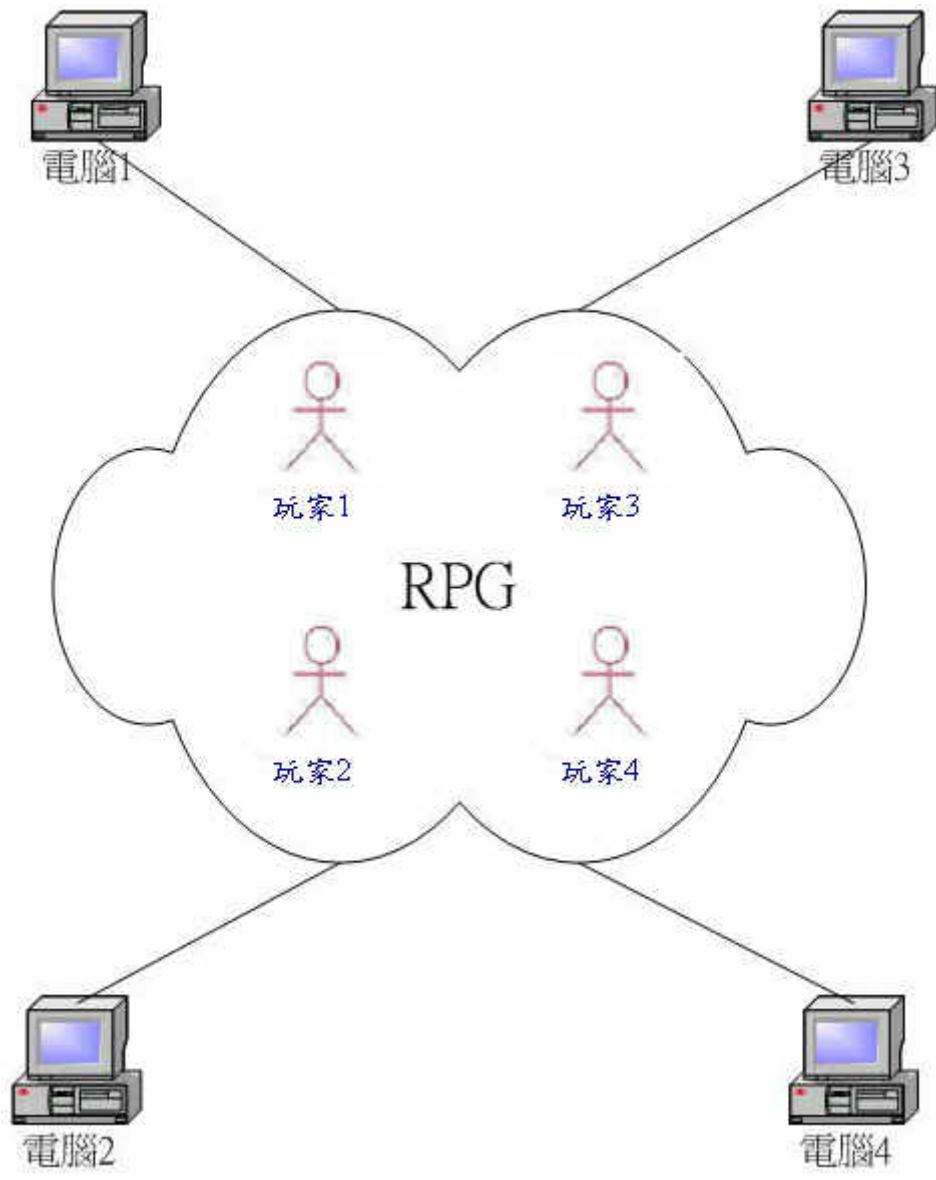


圖 34 改善後的遊戲播放器的網路連線架構圖

五、網路連線角色扮演遊戲產生器之 實作

本章將介紹如何設計及實作前面所述遊戲產生器要加入的三個功能，以達到可產生網路連線遊戲的目標。

5.1 遊戲播放器之網路連線模組實作

首先我們將實作一個網路連線用的模組，讓遊戲設計者使用遊戲播放器時可以測試網路連線的功能。根據第四章的分析結果，我們將採用 Peer-to-Peer 的連線方式，以減小加入連線功能時系統的變更幅度。常見的網路連線模組有 WinSock 與 DirectX 兩種製作方式，在此我們將採用 DirectX 中的 Direct Play 來實作此連線模組[25][26][27][28]。

使用 Direct Play 有下列優點：

1. Direct Play 直接就支援 Peer-to-Peer 架構的 Function。
2. Direct Play 可包裝傳輸媒介(Serial Connection、IPX/SPX、TCP/IP、Modem)的實作細部，簡化程式設計師與硬體的溝通。讓程式設計師不用特別去考慮玩家將用什麼方式連線，也不用考慮玩家使用的網路卡是那一塊，所有與底層傳輸媒介硬體的溝通全都會被 Direct Play 包裝起來。舉例來說，如果玩家選擇使用 TCP/IP 方式連線，則 Direct Play 啟動時會自動詢問玩家要連上哪個 IP 位址；當增加新的通訊媒介時，使用 Direct Play 的遊戲也會自動運用，程式設計師不用一一去考慮每個細節，只要考慮上層的 Function 如何撰寫即可，大大減輕了設計上的負擔。

5.1.1 連線管理

5.1.1.1 Session 概述

Direct Play 活動的組合稱為 Session，即 Communication Session 的簡稱。依所使用連線方式的不同，玩家可以在一個網路硬體上同時執行不同遊戲的許多 Session。Direct Play 同時提供了列舉 Session、建立 Session、加入現有的 Session 等連線方式。一旦 Session 被建立，玩家就可以互相建立連線。連

線建立後，Direct Play 再提供了建立、刪除、與列舉 Session 內玩家物件資料的功能。

5.1.1.2 Session 的辨識

Direct Play 對於 Session 的辨識，使用一個稱為「通用唯一識別碼」GUID(Globally Unique Identifiers)的方式判別。GUID 是一個 32 位數的十六進位序列，而且其內容值必定為唯一，下面是一個有效 GUID 的例子：

6B29FC40-CA47-1067-B31D-00D010662DA

Direct Play 使用此序列，在通訊網路上確認 Direct Play 的物件、Sessions、資料區塊與應用程式。因此啟動 Direct Play 的第一步，就是建立一個有效的 GUID，以便網路上的應用程式可以互相辨別誰要與誰產生連線。

5.1.1.3 Session 的開啓

當玩家成功啟動連線後，接著就是要加入一個遊戲 Session 與來自其他網路位址的玩家一起玩。Direct Play 首先會收集一份在目前連上的網域中所能找到的 Session 列表並列出供玩家選擇，玩家可以選擇要加入一個既有的 Session 還是要建立一個新的 Session 以供其他玩家加入遊戲，這兩個動作都稱為 Session 的開啓。當 Session 完成開啓後，即表示玩家已經成功加入了這個遊戲的網路連線協定中，接著就可以開始傳送與接收遊戲內的資料封包並進行遊戲。

5.1.2 連線流程

圖 35 是 Direct Play 的連線流程：

1. 建立 GUID 並初始化 Direct Play
2. 列舉可使用的連線方式
3. 選擇要使用的連線方式並初始化網路連線模組
4. 列舉網路上可找到的現存 Session
5. 選擇自行建立一個新的 Session，或加入一個現有的 Session 中
6. 開始傳送與接收資料封包
7. 切斷連線

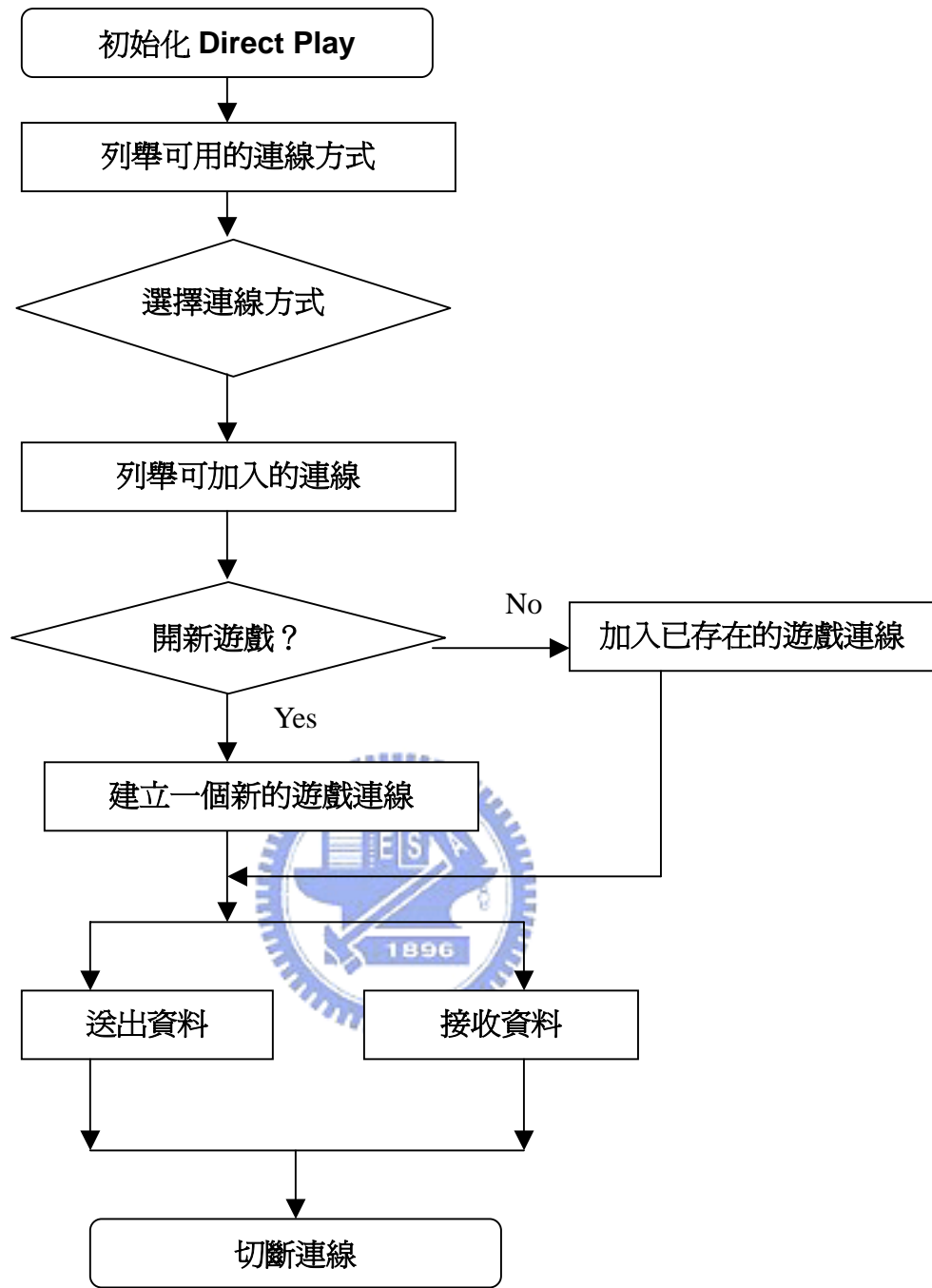


圖 35 DirectPlay 連線流程

5.1.3 訊息處理

5.1.4.1 訊息的完整性

Direct Play 會定義遊戲中會使用的訊息類型。當 Direct Play 物件收到訊息時，首先會執行完整性檢查以便別訊息是否在傳送過程中受損。同時，

不屬於遊戲所定義的類型也會被丟棄不放入訊息佇列裡。只有內容完整且類型正確的訊息才會保留在訊息佇列裡，因此從佇列種取得的訊息可以保證沒有在傳輸過程中受損。

5.1.4.1 傳送訊息

遊戲播放器在傳送訊息時，可選擇使用保證傳到或不保證傳到的方式來傳送資料封包。使用保證傳到的方式會多花一點時間來確認資料封包是否正確傳達，但一般來說，由於遊戲進行時所要傳送的資料封包非常多且非常頻繁，中間漏掉一些封包對遊戲的進行並不會有太大影響，反而會因頻繁的檢查而拖慢了遊戲速度。因此在此採用不保證傳到的方法，以保障遊戲速度的執行。

5.1.4.2 接收訊息

當收到一段其他連線機器送來的訊息時，Direct Play 會先檢查訊息的完整性與正確性，確認無誤後再放進訊息佇列內等待遊戲播放器去讀取該段資料訊息並使用。遊戲播放器對於訊息的接收方式採用多重執行緒(Multi Thread)及狀態池(Status Pool)的作法。

由於遊戲進行時無法判斷何時會有資料封包進來，因此讓系統不斷去檢查訊息佇列會造成主程式運作中大量的「空轉」接收動作，是非常浪費時間及 CPU 資源的方法。因此使用多重執行緒的方法，可以讓原有的程式主執行緒(Main Thread)無須考慮何時會有資料封包進來，將一切的檢查動作都交給另一條接收執行緒(Receive Thread)處理。Status Pool 則是一個用於讓接收執行緒與主執行緒互相溝通的共享資料區段(Share Data Block)，負責存放所有要用在網路間傳輸的資料。接收執行緒會不斷檢查 Direct Play 的狀態佇列，並在有新訊息封包進來的時候解析封包的資料內容更新至 Status Pool。主執行緒要使用其他網路端主機的資料時直接從 Status Pool 讀取即可取得最新的狀態，同時主執行緒在每次更動 Status Pool 資料時皆要更新 Status Pool 的內容並送出給其他 Session 的機器負責更新該機器上的遠端的 Status Pool。如此即可達到傳送與接收資料的即時更新性與正確性。

5.2 多重角色資料結構實作

5.2.1 多重玩家角色的資料結構

遊戲播放器會將同一地圖內的所有玩家透過繪圖引擎及場景控制模組來顯示在畫面上。圖 36 是多重玩家系統下的遊戲播放器的類別圖(Class Diagram)：

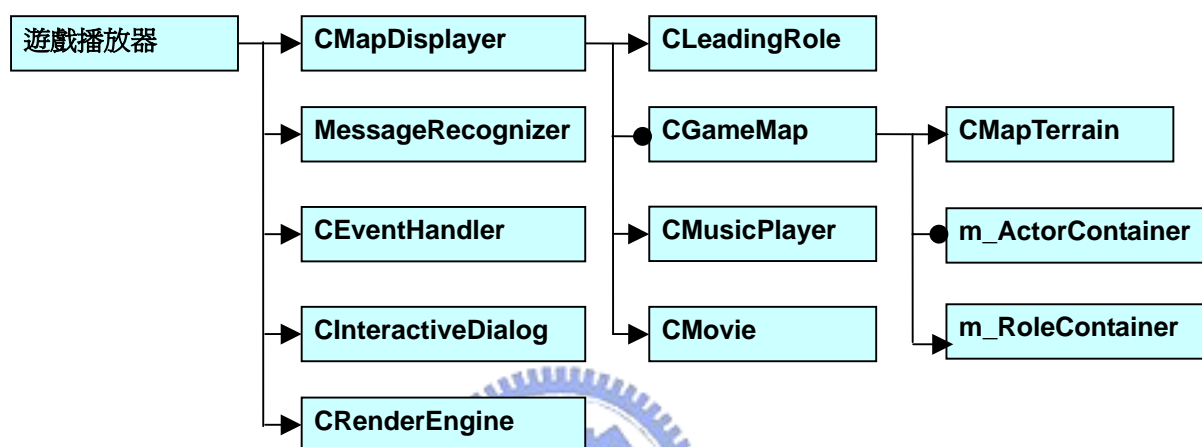


圖 36 遊戲進行中多重角色的類別圖

以下是這些 Class 的功能簡介：

- (1) CMapDisplayer：所有遊戲場景的容器(Container)，負責控制及顯示場景中的所有物件。
- (2) CInteractiveDialog：遊戲的對話盒物件，當播放器執行到對話盒的指令時，對話盒物件就會展現在畫面上，讓使用者可以點選上面的選項。這時播放器再根據使用者所點選的選向來執行相對的指令。
- (3) CRenderEngine：3D 的繪圖引擎，以微軟的 Direct3D 所包裝而成。包含所有的 3D 繪圖函式及碰撞處理偵測。
- (4) CLeadingRole：代表遊戲主角的物件。包含了目前主角的屬性、身上的裝備等資訊。
- (5) CGameMap：遊戲的場景的物件，也是場景中所有演員與角色的容器(Container) 負責顯示及控制所有的演員與玩家角色。
- (6) CMovie：負責動畫播放的物件。當遊戲播放器執行到播放動畫的指令時，播放器就會呼叫這個物件來進行動畫播放。

(7) `m_ActorContainer`：記錄場景上有哪些演員的容器。此資料在遊戲啓動時便已決定有多少個演員。

(8) `m_RoleContainer`：記錄場景上有哪些主角以外的玩家角色的容器。此資料在遊戲啓動時爲空集合，在有其他玩家登入與登出遊戲時，透過 `Direct Play` 將玩家資料傳送到本端機器上接收後才會視更新狀態自動產生與刪除玩家資料。

5.2.2 連線中的多重玩家角色管理

連線環境中各 `Peer` 之間玩家狀態資料的交換，採用一塊 `Status Pool` 的共享資料區段透過 `Direct Play` 來傳輸。`Status Pool` 內記錄了本端玩家所控制的主角與其他遠端玩家所使用的配角等所有角色的狀態資料，其中包括角色的帳號、姓名、座標與屬性等。`Status Pool` 會使用 `Direct Play` 的多重執行緒(`Multi Thread`)方式不斷監聽(`Listen`)網路上傳來的玩家狀態封包並更新到 `Status Pool` 內。系統在執行玩家所下的指令前，會先到 `Status Pool` 內取出所有角色的資料後才執行命令，命令執行完後，若玩家的狀態產生改變（如玩家下移動命令改變了主角的座標），則會把新的角色狀態寫進 `Status Pool` 並立刻使用 `Direct Play` 將更新狀態發送到所有遠端機器，以保證網路連線內所有的 `Peer` 機器上的玩家資料都是最新的。

由於玩家下的指令中會更新角色狀態的情形非常多且非常頻繁，而且所有更新的資料都一律會廣播(`Broadcast`)發送到網路內所有機器上，因此其中有一些封包即使未正確送達，也能迅速由之後的封包得到最新的狀態。故其封包的傳送並不需太著重於保證正確傳送到；因爲 3D 與圖形運算必須暫用大量的 CPU 效能，因此可以犧牲部分封包傳送的正確性來保障遊戲執行的速度。圖 37 是遊戲播放器執行時玩家狀態資料封包的傳遞方式：

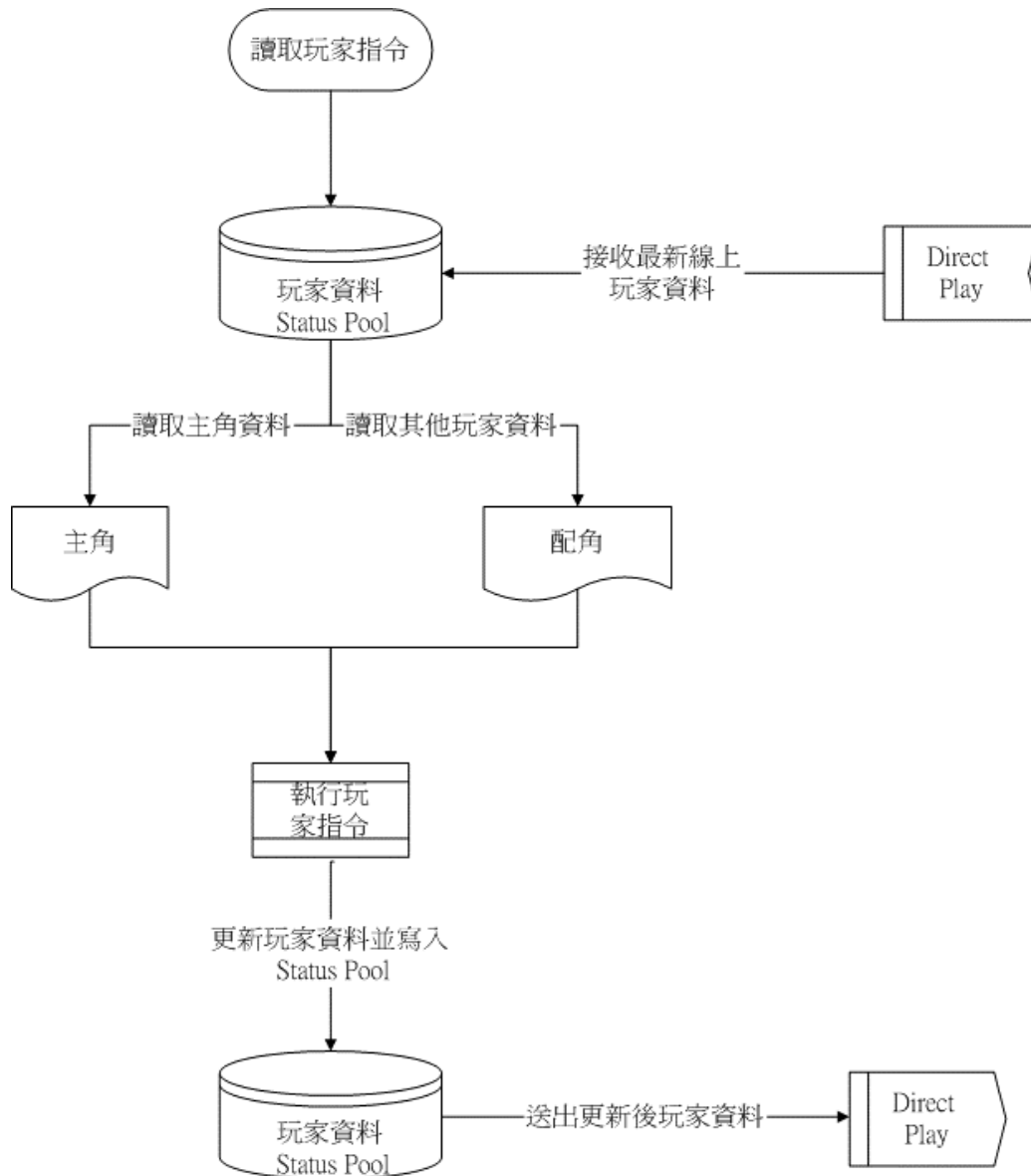


圖 37 遊戲進行中資料封包的交換方式

5.3 主角屬性與事件編輯功能實作

5.3.1 主角屬性資料編輯

主角屬性(LeadingRole)資料即為在遊戲播放器中 Status Pool 內所記錄的玩家資料，其格式定義如下：

[Registry_Begin]

[Role]

Name = "Ken"

ModelType = "MD2MODEL"

ModelFile = "graphic\models\Tris\tris.md2"

TextureFile = "graphic\models\Tris\solo.bmp"

IniFile = "graphic\models\Tris\tris.ini"

2DPicturePath = "GRAPHIC\character\photo2.bmp"

[Registry_End]

(1) [ROLE]、Name、ModelType、ModelFile、TextureFile、IniFile、2DpicturePath 皆為關鍵字(Key Word)

(2) Name = "Ken"

定義此角色在遊戲中所使用的名字(Name)。所有玩家看到此角色時的名字都為此值。此外，不同玩家所使用的角色，也可以取同樣的名字。

(3) ModelType = <ModelType>

定義此角色 3D 模型檔的類型，目前支援.X 檔及.MD2 檔兩種。

(4) ModelFile = <FilePath>

定義此角色的 3D 模型檔的檔案路徑。

(5) TextureFile = <FilePath>

定義此角色所使用的 3D 模型貼圖檔案路徑。

(6) IniFile = <FilePath>

定義此角色行走時所使用的動畫檔案路徑。

5.3.2 主角劇情事件編輯

最後在遊戲編輯器裡的事件編輯器中，增加對於主角專用劇情事件的編輯。事件編輯器主要是由指令列(CCommandList)和事件視窗(CEventView)兩個元件所組成的，指令列裡裝了各個指令所代表的圖示(Icon)。事件視窗顯示了目前所編輯的事件內容，並且接受由指令列所拖進來的圖示，來新增一個指令。

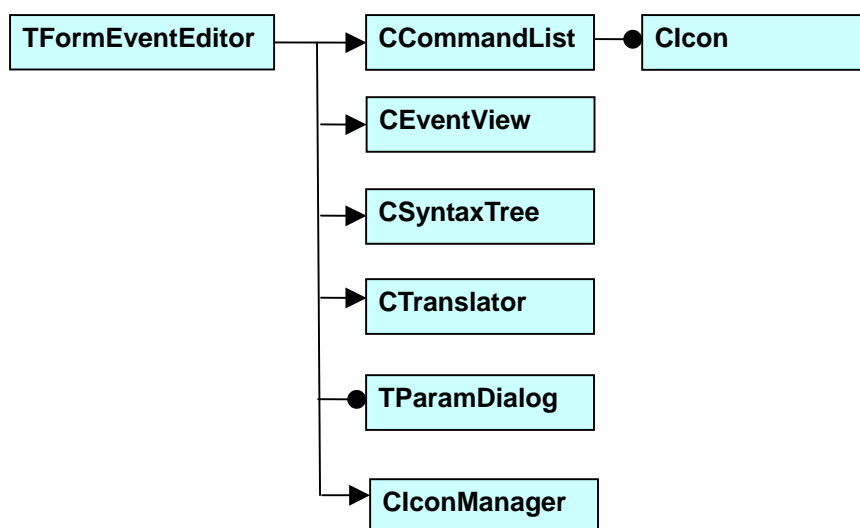


圖 38 事件編輯器的類別圖

- (1) **CCommandList**：事件編輯器的指令列，將遊戲中的指令以圖示表示，將所有的圖示列出，每個圖示對應到一個對話表單。每個圖示(**Icon**)都可以讓使用者拖拉至事件視窗來進行編輯。
- (2) **CEventView**：顯示演員事件的視窗，將指令圖示拖拉到此進行編輯，並呼叫相對應的對話表單來讓使用者輸入指令的參數。在增一個指令後以自然語言顯示目前的事件內容。
- (3) **CSyntaxTree**：紀錄目前編輯的結果的樹狀資料結構(**Parse Tree**)，儲存了整個 事件的文法結構。
- (4) **CTranslator**：描述檔轉換器的類別，描述檔與編輯結果(**CSyntaxTree**)間的轉換。
- (5) **TParamForm**：輸入指令參數的對話表單，讓使用者輸入該指令的參數，每個指令都有一個相對應的對話表單來輸入該指令的參數。
- (6) **CIconManager**：對遊戲指令作分類的模組，我們依照指令的類型分成純文字顯示、影音播放、屬性更動、條件判斷等類型。

事件編輯器的主要 **method** 及運作方法如下：

- (1) **DragDropEvent**：當使用者將指令列的圖示(**Icon**)拖拉到事件視窗時所發生的事件。在指令視窗我們用一個資料結構來記錄視窗上的哪一行是對應到那

個指令，當 Icon 被拖拉到上面時，我們會判斷這個指令是否被合法地插入。如果是正確的插入指令，我們會叫出該指令的對話表單來讓使用者來填該指令所用到的參數。

- (2) FillEvent：將目前的事件的資料結構轉換成自然語言，顯示在事件視窗上。
- (3) TraslateToScript：將事件的資料結構轉成描述語言存入描述檔中。

5.4 系統模組架構

新版的遊戲產生器系統架構，在遊戲播放器中加入了 Direct Play 連線模組、多重角色資料結構與資料封包的訊息交換中心。在遊戲編輯器的部分則加入了主角的屬性資料與劇情事件的編輯功能。圖 39 是加入新模組後的遊戲產生器之系統架構與類別圖，其中深色部分代表新加入的模組。

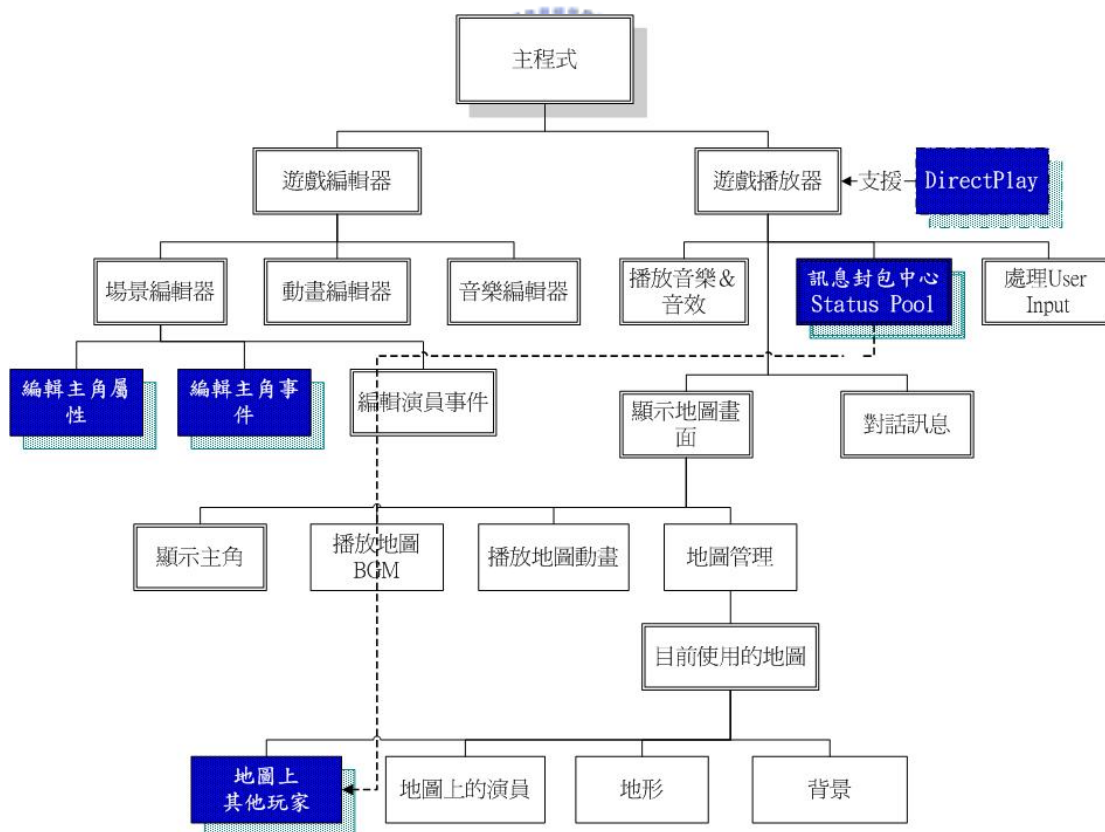


圖 39 新版遊戲產生器的系統類別圖

六、系統展示

在本章中將會展示以遊戲產生器來產生與進行一個網路版角色扮演遊戲的流程。首先由使用遊戲產生器的視覺化編輯介面開始設定遊戲中主角的屬性資料，再由劇情事件編輯畫面中設定主角的專用事件，產生可用於遊戲播放器中執行的劇情腳本檔案。然後以遊戲播放器展示遊戲的網路連線功能，並使用設定好的主角事件來驗證遊戲資料在網路中傳遞的正確性。展示流程共分爲：

1. 視覺化編輯階段
 - (1) 編輯主角屬性資料的操作流程
 - (2) 編輯主角劇情事件的操作流程
2. 遊戲執行階段
 - (1) 網路連線流程
 - (2) 使用主角的劇情事件

6.1 視覺化編輯階段

編輯一個角色扮演遊戲的編輯流程一共包括編輯場景、放置演員、設定演員屬性事件、編輯主角屬性事件等。其中場景與演員的編輯部分在 4.1 節已有介紹，本節將專注於主角的屬性與事件的編輯上。

6.1.2 編輯主角屬性資料的操作流程

依 4.1 節所述的方法建好一個新的地圖，以及其中一個演員名稱爲 **Box** 的爲箱子所屬的劇情事件。我們希望由一位玩家甲扮演一個名叫 **Ken** 的冒險者角色進入遊戲中的地圖來開啓箱子並取得裡面的物品，然後在另一位玩家乙使用另一個不同造型與名字的角色進入與離開遊戲時，都可以收到對方連線與斷線的遊戲訊息。並且在對方進入遊戲時，雙方玩家都可以看到彼此的角色與在遊戲中的動作。

首先我們點選遊戲編輯器工具列上的主角屬性編輯按鈕，進入主角資料的編輯視窗畫面。進入後，可選擇編輯在遊戲中要扮演的主角姓名，與所要使用的人物造型。若要編輯主角的專用劇情事件，則按下『編輯事件』的按鈕即可進入事件編輯畫面：

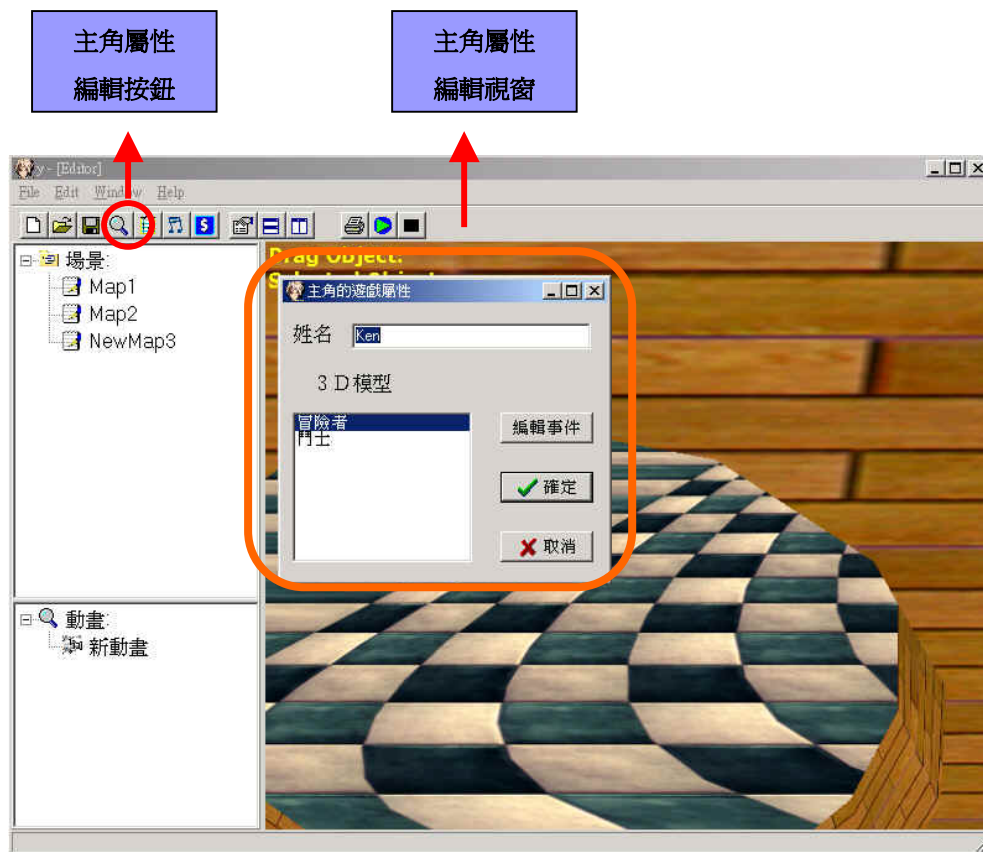


圖 40 編輯主角遊戲屬性

在 3 D 模型的選項中選擇「冒險者」的造型並在姓名欄中輸入姓名後按下確定，所設定的主角姓名與造型即會自動轉成屬性資料檔的格式，並在玩家進入遊戲時自動套用：

```
[Registry_Begin]
[Role]
2DPicturePath = "GRAPHIC\character\photo2.bmp"
ID = "IMYME"
IniFile = "graphic\models\Tris\tris.ini"
ModelFile = "graphic\models\Tris\tris.md2"
ModelType = "MD2MODEL"
Name = "Ken"
Position = "(50.000000 0.000000 -20.000000)"
Quaternion = "(0.000000 0.000000 0.000000 1.000000)"
Size = "(2.600074 2.600074 2.600074)"
TextureFile = "graphic\models\Tris\solo.bmp"
[Registry_End]
```

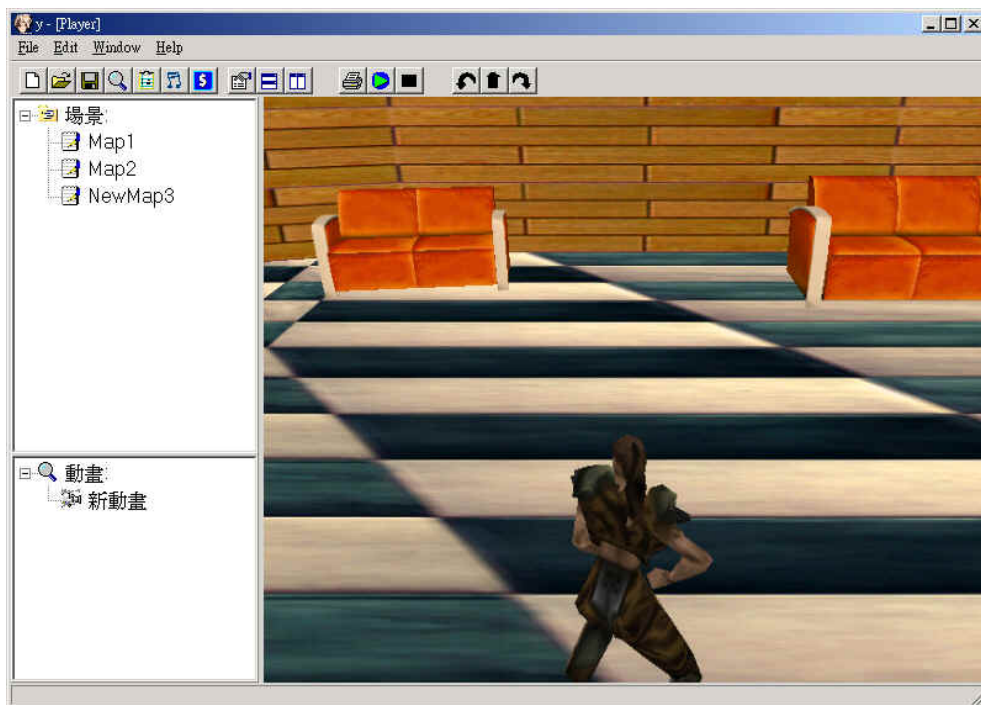


圖 41 套用主角遊戲屬性(1)

接著用同樣的方式，設定玩家乙所用的主角造型與姓名等資料：

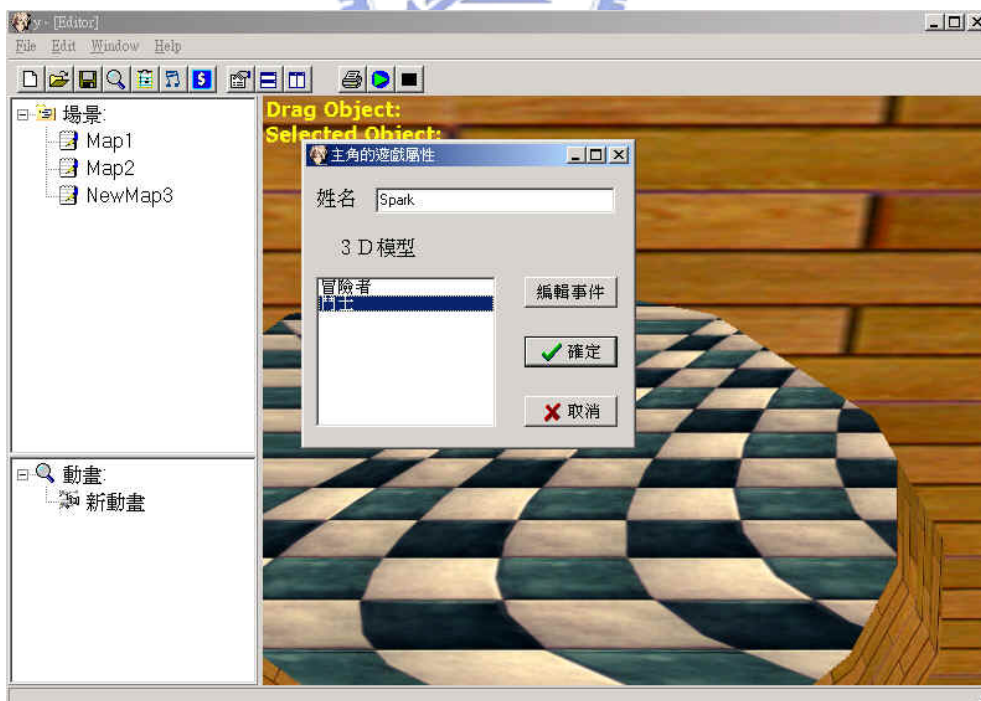


圖 42 套用主角遊戲屬性(2)

6.1.2 編輯主角劇情事件的操作流程

從屬性資料的畫面中按下『編輯事件』按鈕即可進入主角的事件編輯畫面。在這裡，我們希望玩家甲所控制的角色可以在有玩家乙做登入(Login)與登出(Logout)時收到一個提醒訊息，以了解目前遊戲中線上人數(Online User)的變化。因此我們從指令列中將『連線訊息』以滑鼠拖拉到事件視窗以加入此一新事件。畫面上接著會出現一個輸入連線訊息用的對話窗，其中「網路位址」為訊息中的保留字，在遊戲進行時會自動替換為該玩家連線時電腦在網路環境中的網路位址。如玩家乙是從一部在網路中名稱爲 SPS702 的電腦與遊戲連線，則玩家甲就會在對方連接網路連線的同時立刻收到一句「來自網路 SPS702 的玩家連上遊戲了」的訊息。

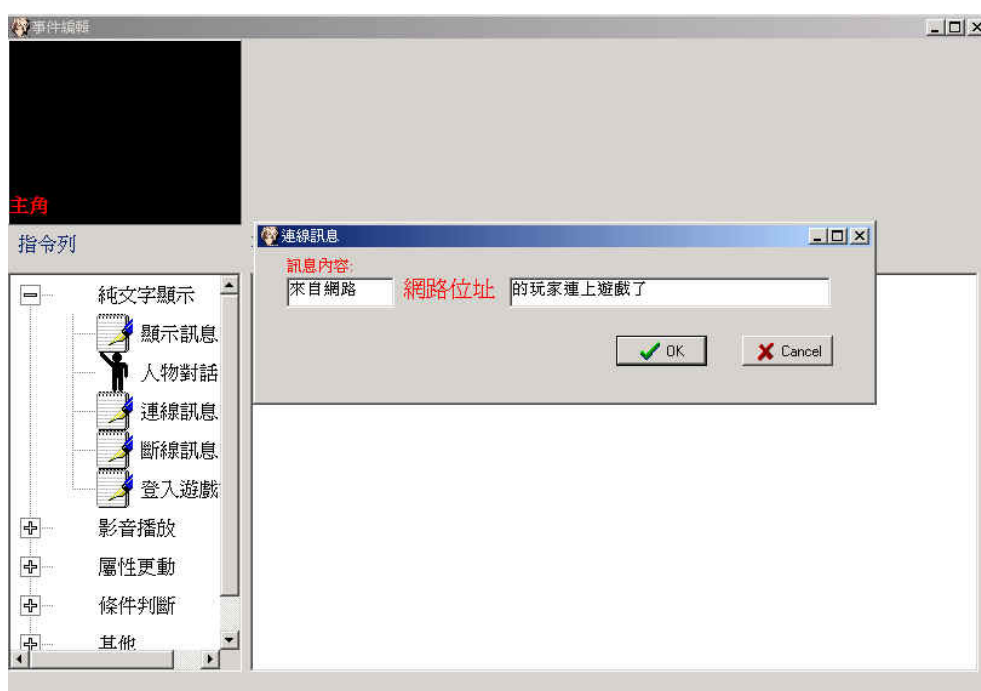


圖 43 編輯主角劇情事件(1)

按下 OK 的按鈕後，在事件視窗就會出現一個“[連線訊息]”的指令，後面跟著這個指令的內容。接著我們以同樣的方式將『斷線訊息』和『登入遊戲訊息』拖拉到事件視窗並加入這兩個指令。前者用在對方玩家切斷網路連線退出遊戲時收到的訊息，後者用在玩家建立網路連線後進入遊戲時收到對方登入遊戲的訊息：

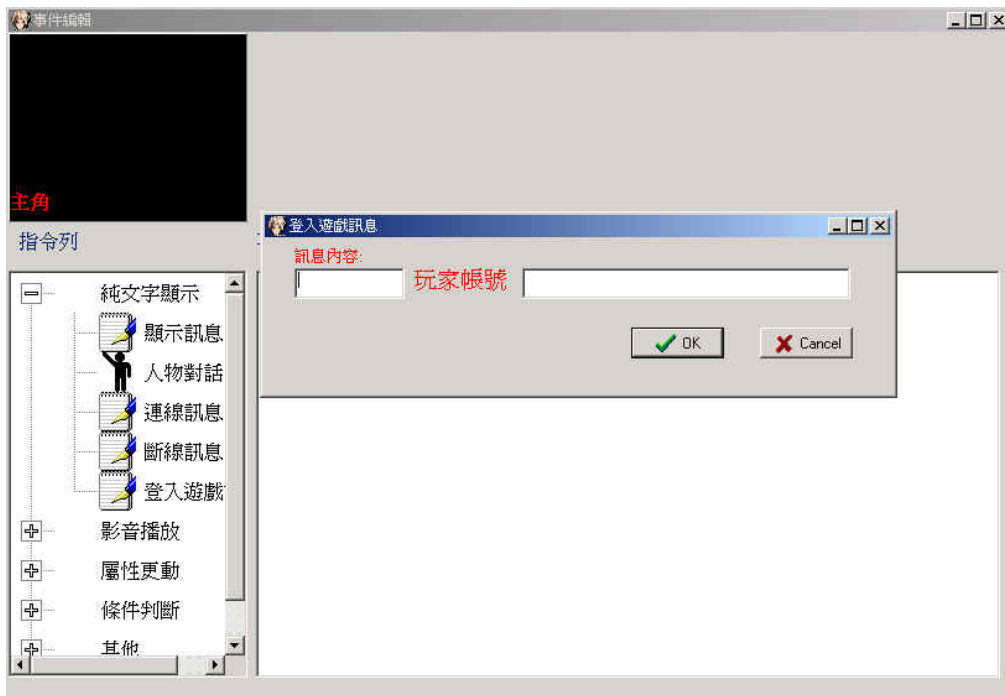


圖 44 編輯主角劇情事件(2)

最後，在玩家乙的主角劇情事件中同樣將這三組事件訊息加入：



圖 45 編輯主角劇情事件(3)

6.2 遊戲播放階段

完成兩部電腦上個別的主角屬性資料與劇情事件編輯後，接下來我們將透過遊戲播放器中的網路連線功能使兩部電腦互相連線，來測試多人遊戲環境的效果。

要讓遊戲可以正確互相連線，必須在進入遊戲之前，先啟動網路連線功能連上網路。首先按下網路連線的按鈕叫出 **Direct Play** 的網路連線視窗，從中選擇連線的類型。

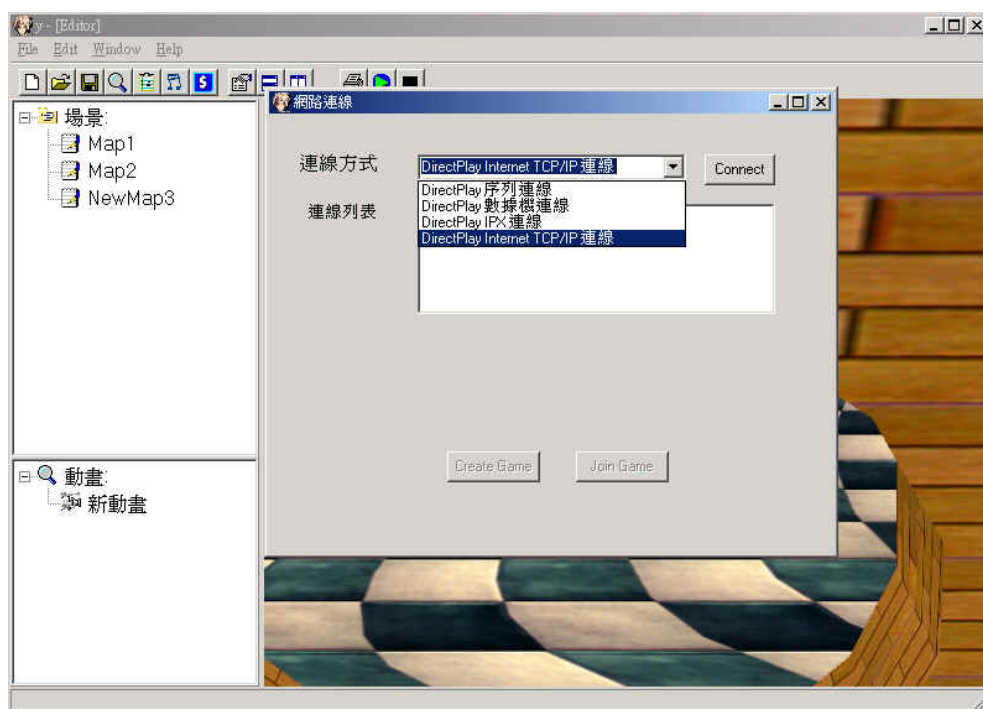


圖 46 網路連線流程(1)

如同 5.1 節所述，**Direct Play** 已經為我們提供了序列、數據機、IPX 和 TCP/IP 四種連線方式，我們從這裡讓玩家甲選擇 TCP/IP 方式，並按下 **Connect** 按鈕開始連線流程。

接下來要決定是要自己開一個新遊戲，或是加入網路上已經存在的別人開好的遊戲。要加入一個已存在的遊戲，**Direct Play** 必須先知道對方電腦所在的網路位址，才能搜尋到遊戲的連線資料，因此 **Direct Play** 會先跳出一個視窗要求輸入要搜尋的對象在網路中的名稱。若對方在 **Internet** 上，則在此必須輸入對方的電腦名稱或 IP 位址；若對方在區域網路(**Localnet**)內，則輸入可以留空白直接按下確定，**Direct Play** 可以自動搜尋到區域網路內已存在的遊戲連線。在這

裡我們讓玩家甲來建立一個新的遊戲連線，然後由玩家乙來連上這個遊戲。因此不必特別輸入網路位址去搜尋連線，直接按下確定之後執行 **Create Game** 表示要建立一個新的遊戲連線，即完成 **Direct Play** 的連線設定。

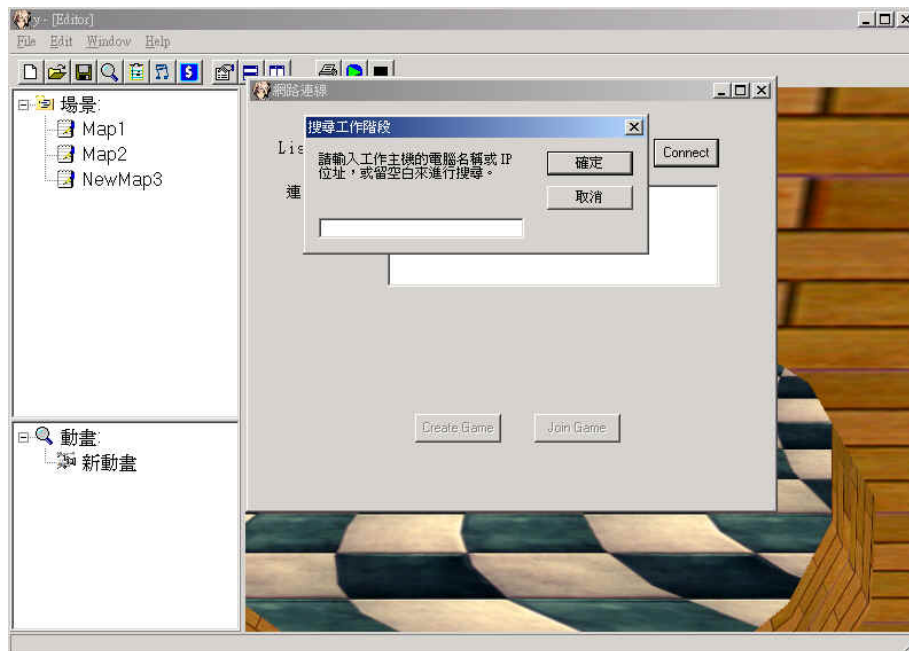


圖 47 網路連線流程(2)

接著按下工具列上的 **Start Play** 按鈕進入遊戲。點選之前設定好事件的箱子，就會接著跳出我們設定好的劇情選項：

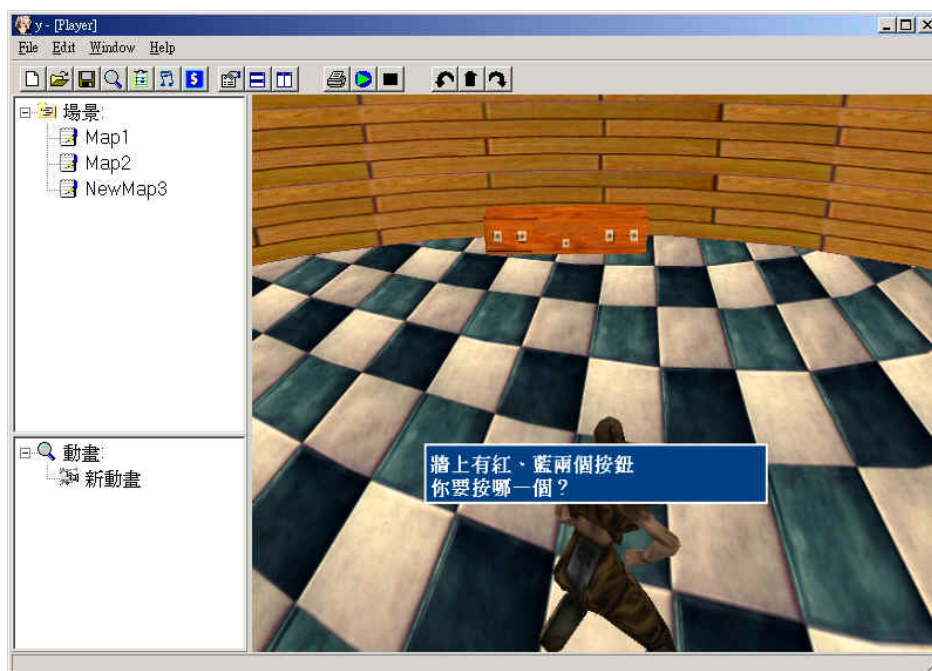


圖 48 演員事件播放流程(1)

在畫面的選項中，主角的姓名與臉孔圖示也會顯示在上面。選擇按下藍色按鈕，就可以拿到箱子裡的自動機槍，完成演員事件的播放。

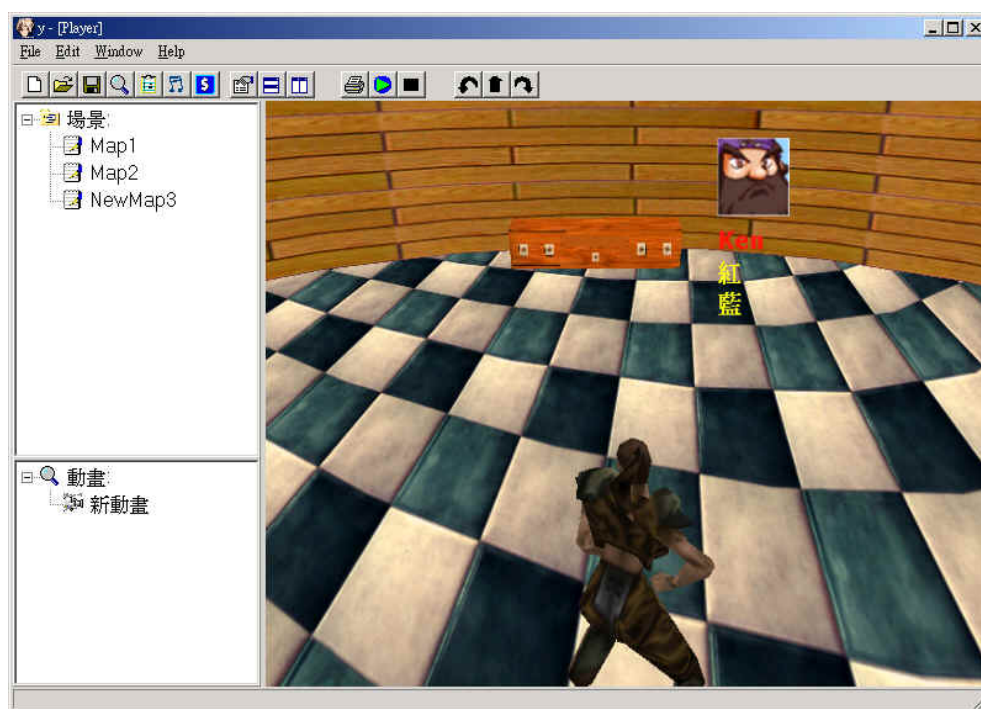


圖 49 演員事件播放流程(2)

接著我們讓另一部電腦的玩家乙也進入遊戲中，來觀察網路連線與主角事件的播放。我們同樣讓玩家乙叫出網路連線視窗後選擇 TCP/IP 方式連線。由於兩部電腦是在區域網路內，因此搜尋主機的地方同樣讓 Direct Play 自動搜尋。Direct Play 會將搜尋到的遊戲連線，自動填入連線列表中。選擇列表中要加入的遊戲連線，再按下 Join Game，即可進入這個遊戲連線中，完成網路連接的程序：

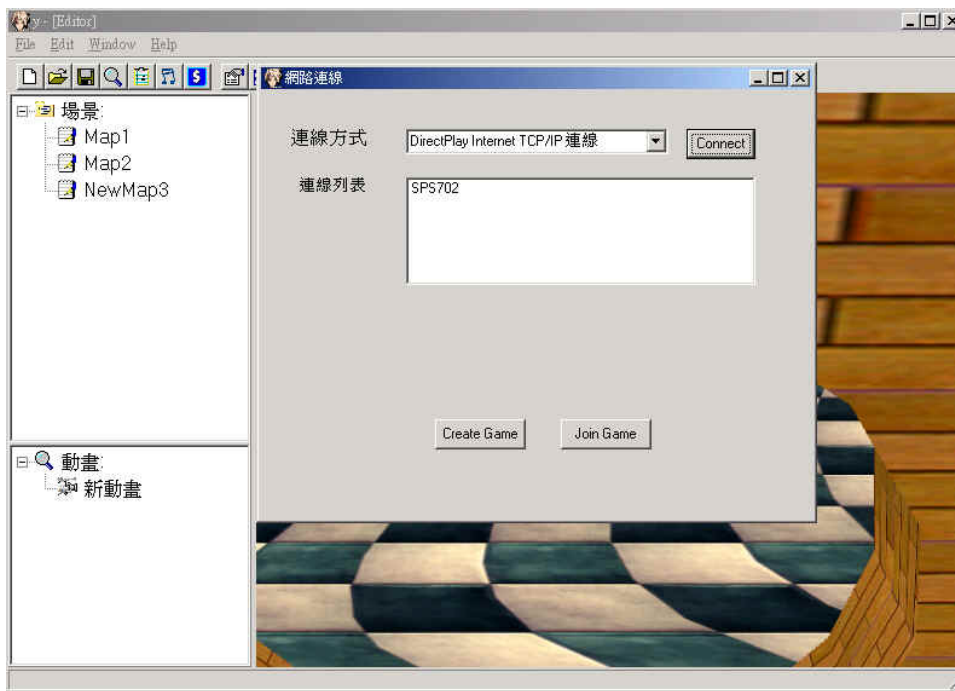


圖 50 網路連線流程(3)

當玩家乙按下 **Join Game** 加入原有的遊戲連線時，玩家甲就會立刻收到在玩家事件中所設定的連線提醒訊息，知道從某個網路位址上有別的玩家準備要連進這個遊戲內：



圖 51 使用主角劇情事件(1)

接著，當玩家乙按下 **Start Play** 進入遊戲後，雙方玩家的 **Direct Play** 就立刻開始自動傳送與接收彼此的角色資料，並在遊戲中立刻建立出對方玩家所使用的角色。同時玩家甲會收到一句對方新玩家進入遊戲的事件訊息。

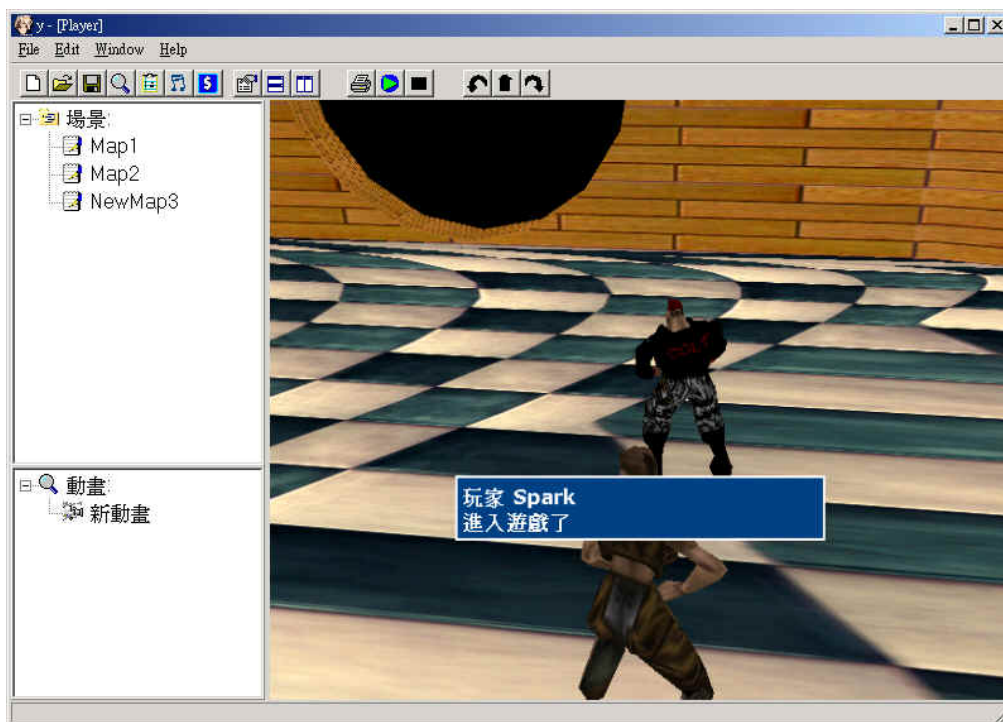


圖 52 使用主角劇情事件(2)

最後，當玩家乙切斷連線退出遊戲時，系統會自動刪除玩家乙所使用的角色資料，並傳送一句設定好的離線事件訊息給玩家甲。



圖 53 使用主角劇情事件(3)



七、結論與未來發展方向

7.1 結論

在本論文中，我們探討了開發遊戲產生器的原因，在於簡化遊戲開發過程中程式設計師與遊戲設計者之間的溝通流程，加快遊戲劇情的開發速度；並提供視覺化方式的編輯功能與描述檔轉換工具，讓遊戲設計者可以輕鬆直觀的方式設計遊戲劇情，不用撰寫文字式的描述語言。

進階版的遊戲產生器，則提供了網路連線能力，以及 **Multi-User** 的支援。並加入了對於和主角本身的屬性與事件的編輯能力，讓遊戲設計者可用於網路連線類型遊戲的劇情設計上。

7.2 未來發展方向

對於我們所設計的進階版遊戲編輯器，我們提出了以下的未來發展方向：

- (1) 增加 **Server-Client** 網路連線方式與專用遊戲伺服器(**Game Server**)的模組，來開發大型線上遊戲(**On-Line Game**)。**Direct Play** 的優點在於擴充方便與撰寫容易，但一般只用於支援小型連線遊戲，若能加入 **Server-Client** 方式連線遊戲的編輯開發支援，則可用以發展成線上角色扮演遊戲。
- (2) 網路連線遊戲提供非常龐大的玩家資訊而增加了操作上的樂趣，但玩家資料封包的傳輸中並無法保障其資料不會產生傳輸延遲、封包遺失、未依正常順序送達或資料內容有損壞等問題。其原因除了玩家使用的網路連線頻寬等硬體問題外，遊戲本身在資料交換運算與封包傳遞的安全性上也有極大的影響。因此如何再提高系統處理封包的速度與可靠度，乃是一個重要的課題。
- (3) 隨著下一代無線網路的來臨，可以將無線網路上的遊戲和目前的 **Internet** 上的多人連線遊戲整合，形成新一波網路遊戲的風潮。以後的玩家出門在外時，透過行動電話或是 **PDA** 上網方式來繼續他的虛擬冒險遊戲。如此參與遊戲的人數和時間都將比目前的網路遊戲要來的多與長。

參考文獻或資料

- [1] 資策會MIC, 1998-2004 年全球遊戲產值, January, 2002.
- [2] 電腦玩家雜誌, 「國產遊戲十年史」, 2001.08
- [3] IDC(國際數據資訊), <http://starlike.idv.tw/>
- [4] Jim Adams, "Programming Role Playing GAMES with DirextX", Traditional Chinese edition copyright 2003 by DrMaster Press Co., Ltd.
- [5] Mason McCuskey, "Special Effects Game Programming with DirectX", Traditional Chinese edition copyright 2002 by DrMaster Press Co., Ltd.
- [6] Chihiro SAKAMOTO, "Role Playing Game Programming", Traditional Chinese edition copyright 2002 by DrMaster Press Co., Ltd.
- [7] Ismodel, PC遊戲編程, <http://www.vchelp.net>, 2002-8-9.
- [8] 葉展, 《電子遊戲軟體》, 遊戲理論研究, 中國遊戲開發網.CN, 2001-09-11.
- [9] 新藝術遊戲學苑, http://www.bone.idv.tw/Non_Travel/MyGames/NuartCourse/
- [10] 郭靜怡、陳蕙瑛、王玲華、彭詩晴、陳姿薰, 「線上遊戲專業能力與人才培育之研究」, 國立彰化師範大學商業教育學系遊戲專題, 民國 92 年
- [11] 林美純、郭靜怡、王玲華、陳蕙瑛、彭詩晴、陳姿薰, 「線上遊戲企劃專業能力與人才培育之需求」, 國立彰化師範大學商業教育學系遊戲專題, 民國 93 年
- [12] Drew Sikora, "You Got Game", GameDev.net, March, 2001.
- [13] Francois Dominic Laramee, "The Game Design Process", GameDev.net, November, 1999.
- [14] 黃國洲, 「我國電腦遊戲產業之新產品開發策略考量」, 元智大學資訊傳播學系碩士論文, 民國 92 年
- [15] 周榮, 「電腦角色扮演遊戲中的神話原型之研究」, 交通大學資訊科學研究所碩士論文, 民國 87 年
- [16] 郭昕周, 建構取向的遊戲式 MUD 學習環境研究, 國立交通大學傳播研究所碩士論文, 民國 86 年
- [17] Chorng-Shiuh Koong, "A Component-based Visual Scenario Construction Environment for Non-Programming Users to Create Interactive Electronic Books", PHD Thesis, N.C.T.U. Taiwan, October, 2000.
- [18] Kang Zhang, Da-Qian Zhang, Yi Deng, "A Visual Approach to XML Document Design and Transformation", Proceedings IEEE Symposia, 2001.
- [19] Kang Zhang, Da-Qian Zhang, Jiannong Cao, "Design, Construction, and

- Application of a Generic Visual Language Generation Environment”,
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, APRIL 2001.
- [20] Sherif M. Yacoub, Hengyi Xue, Hany H. Ammar, “Automating the
Development of Pattern-Oriented Designs for Application Specific
Software Systems”, Proceedings IEEE Symposia, 2000.
- [21] 卓勇君, 「3D 角色扮演遊戲編輯系統的設計與製作」, 交通大學資訊科學研
究所碩士論文, 民國 92 年
- [22] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen,
“Object-Oriented Modeling and Design”, Prentice-Hall, New York, 1991.
- [23] Grady Booch, “Object-Oriented Analysis and Design With Applications”,
The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1994.
- [24] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, “Design
Patterns, Elements of Reusable Object-Oriented Software”,
Addison-Wesley Publishing Company, Inc., New York, 1996.
- [25] Xuefeng, 「Microsoft DirectX 8.1 VB Document中文翻譯」,
<http://blog.csdn.net/asklxf/archive/2004/09/20/110997.aspx>, 2004-9-20
- [26] 楊冰, 「DirectPlay SDK 中文翻譯 Understanding DirectPlay」, 2002-9-26
- [27] Jim Adams, 張世敏, 2D/3D RPG角色扮演遊戲程式設計：使用Direct X,
博碩出版社, 2003-02-25
- [28] 榮欽科技, Visual C++遊戲設計魔法寶典, 碁峰出版社, 2004-03-10