

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

RPG遊戲角色人物圖形產生器 暨資料庫



學 生 : 林佳慧(四丁)
藍卞鴻(四丁)

指 導 教 授 : 林國貴老師

中 華 民 國 九 十 三 年 四 月

目 錄

摘要.....	4
第一章：導論	
1.1 動機與目的.....	5
1.2 討論遊戲業界.....	5
1.3 遊戲美工的地位.....	9
1.4 美工製作過程.....	10
1.5 遊戲類別簡介.....	12
第二章：系統架構	
2.1 系統簡介.....	17
2.2 人物自動產生器.....	17
2.3 資料表設計.....	18
第三章：系統實作	
3.1 發展環境.....	23
3.2 使用 JDBC 連線.....	23
3.3 連結資料庫.....	26
3.4 存取資料庫.....	27
3.5 介面製作.....	29
3.6 使用者介面.....	29
第四章：系統分析與評估	
4.1 使用評估.....	47
4.2 資料流程圖.....	47
4.3 未來發展.....	51
第五章：結論與心得	
5.1 結論與心得.....	57
5.2 參考資料.....	58

圖 片 目 錄

圖 1.1	STREET FIGHTER ZERO3.....	12
圖 1.2	DEAD OR ALIVE.....	12
圖 1.3	惡魔城年代記.....	13
圖 1.4	屠龍勇士.....	13
圖 1.5	DRAGON QUEST VII.....	13
圖 1.6	STAR OCEAN 2.....	13
圖 1.7	BIO HAZARD 3.....	13
圖 1.8	GetBackers!奪還屋.....	13
圖 1.9	FINAL FANTASY TACTICS.....	14
圖 1.10	德貝賽馬 99.....	14
圖 1.11	SISTER PRINCESS.....	14
圖 1.12	育嬰問答 0.....	14
圖 1.13	骰子方塊 XI.....	14
圖 1.14	魔法氣泡.....	14
圖 1.15	Warcraft III:Reign of Chaos.....	15
圖 1.16	Medieval: Total War.....	15
圖 1.17	Ridge Recer4.....	15
圖 1.18	爆走貨車傳說.....	15
圖 1.19	實況野球 2001.....	15
圖 1.20	WINNING ELEVEN 2002.....	15
圖 1.21	空戰奇兵 3.....	16
圖 1.22	RAYCRISIS.....	16
圖 1.23	我的料理.....	16
圖 1.24	柏青哥王國 2.....	16
圖 2.1	人物面向的方位.....	18
圖 2.2	資料關聯圖.....	22
圖 3.1	JDBC 連線方式.....	24
圖 3.2	Type1 驅動程式連線方式.....	25
圖 3.3	ODBC 聯結資料庫流程.....	27
圖 3.4	存取資料庫的流程.....	28
圖 3.5	登入畫面.....	29
圖 3.6	密碼錯誤.....	30
圖 3.7	功能選擇畫面.....	30
圖 3.8	角色資料表.....	31
圖 3.9	人物圖形自動產生器(未選取).....	32
圖 3.10	選擇模型圖片.....	33

圖 3.11	選擇衣服圖片.....	33
圖 3.12	選擇髮型圖片.....	33
圖 3.13	人物圖形自動產生器(已選取).....	34
圖 3.14	人物圖形自動產生器(已刪除).....	35
圖 3.15	儲存完成圖片.....	35
圖 3.16	新增角色資料.....	36
圖 3.17	確定離開.....	36
圖 3.18	更新角色資料.....	37
圖 3.19	組合結果.....	37
圖 3.20	模型資料表.....	38
圖 3.21	增加模型資料.....	39
圖 3.22	選擇硬碟中的圖檔.....	40
圖 3.23	存檔成功對話窗.....	40
圖 3.24	請輸入 ID 欄位的值.....	41
圖 3.25	是否刪除對話窗.....	41
圖 3.26	成功刪除對話窗.....	41
圖 3.27	更新模型資料.....	42
圖 3.28	ID 已存在對話窗.....	43
圖 3.29	資料更新成功對話窗.....	43
圖 3.30	儲存資料庫圖片.....	43
圖 3.31	服裝資料表(800*600).....	44
圖 3.32	增加服裝資料.....	45
圖 3.33	髮型資料表.....	46
圖 3.34	增加髮型資料 (750*650).....	46
圖 3.35	更新髮型資料.....	47
圖 3.36	群組資料表.....	48
圖 3.37	新增群組資料.....	48
圖 3.38	更新群組資料.....	49
圖 3.39	樂、嵐主畫面選單.....	49
圖 3.40	樂、嵐遊戲畫面.....	50
DFD 資料流程圖		
圖 4.1	Context Diagram.....	52
圖 4.2	Diagram 0.....	53
圖 4.3	Diagram 1.....	54
圖 4.4	Diagram 2.....	54
圖 4.5	Diagram 3.....	55
圖 4.6	Diagram 4.....	55
圖 4.7	Diagram 5.....	56

摘要

本系統---RPG 遊戲角色人物圖形產生器暨資料庫，是為了遊戲業界中，佔有重要地位的美工人員所設計的，將美工所畫過的圖片分門別類，以資料庫的方式管理，以便於圖片的再利用，不會使畫過的圖片就這樣浪費，也方便對於各式各樣的圖片，做比較以及對照。

在台灣美工人員缺乏的情況下，美工品質也會跟著下降，鑑於此，我們開發了針對於 RPG 遊戲的人物圖形，以電腦化的自動產生方式，使美工人員的工作量減輕，可以多點心思放在主要的設定畫上；將遊戲人物圖形分成三個部份，利用上述所說的資料庫中的各部分圖形，自動化的將所有組合呈現在使用者面前，讓使用者依照遊戲需要，選出適合的圖形。

以下為各章節的要點簡介：

第一章 導論

研究的緣由，說明選擇此專題题目的原因，工作的分配與進度，專題的目標與進度，還有對於遊戲業界與美工的簡介。談論現在全世界的電玩業界動向，與美工工作簡介，美工在遊戲中的地位，還有簡單的遊戲分類介紹。

第二章 系統架構

探討研究專題的製作方法、大致架構以及可行性，以利於往後的實作。

第三章 系統實作

系統的實作方法，說明 JAVA 與資料庫之間的關係，與連接的方式、資料庫的建立，資料庫的應用，以及人物圖形自動產生器的實作說明。

第四章 系統分析與評估

系統需求的說明，以及資料庫介面、人物產生器介面的使用解說。對於此專題的未來發展做討論，可以由此系統發展出更多的功能或者是更多樣化的應用。

附錄 小遊戲[樂·嵐]

第一章 導論

1.1 動機與目的

目前全世界的遊戲產業正在蓬勃發展，而台灣方面雖然步調較慢，可是在遊戲業者的努力之下，也在近幾年來，漸漸使台灣的遊戲能夠嶄露頭角，甚至將對象擴展至大陸、日本、以至於全世界，不過在這樣的情況下，遊戲人才的缺乏就是現在台灣遊戲產業面臨的最大問題，常常發現，許多的遊戲開發公司或是小組，在企劃、美工人才的缺乏下，常常是一位美工當作十位用，企劃也是如此，更糟的情況，就是刪除企劃這個職位，讓大家一起兼任。

為了解決現在台灣遊戲的美工人員極度缺乏的問題，我們試著找出方法，來幫助美工減少一些具有規律性，且重複性高的工作，例如，RPG 遊戲中，除了主要的遊戲設定畫外，在人物行走的圖方面，常常是相似性很高的，尤其是配角，或是村民、路人等非主要角色，很多相似的圖片，但是美工卻得一個個慢慢畫，不只降低了製作的速度，也使品質降低，而且這些繁瑣的工作，造成美工人員在主要的遊戲畫面，人物的設定等水準的下降；也因為需要畫的角色數量太多，如果不以一個有效率、有條理的方法來管理，常常會發生人物的畫像重複的問題。

開發 RPG 遊戲人物角色時，需要反覆組合、更換圖片，這樣既不方便又沒有效率，美工人員需以手動的方式反覆比較各組合結果，容易因為一次比較、觀看太多的組合，而到最後忘記之前已看過的組合，無法作出正確的判斷。另外，美工人員開發好人物圖片後，儲存的方式缺乏有系統的管理，常常會找不到需要的圖片。綜合以上要素，我們希望作出一個能更方便美工人員開發角色人物的系統。

1.2 討論遊戲業界

遊戲業在近幾年來的發展是全世界有目共睹的，遊戲在一開始，只不過數百萬的市場；但誰也沒有想到，在二十年後會發展成上百億美元的市場。

遊戲不但取代許多人娛樂性向，也使人們視電玩遊戲為休閒娛樂之一種選擇，更進而促使遊戲市場成為一固定市場。電玩遊戲其實範圍極廣，但從最主要的獲利市場來看，它可包括：「大型電玩」、「電視遊樂器」、「電腦遊戲」、「掌上型電玩」等四個主要的市場；不過其中，以「電視遊樂器」的市場為最大。以美國來看，遊樂器的市場為電腦遊戲的"四倍"大。(指銷售金額)

目前的遊戲業界主力國家為日本，再來是美國與韓國，這些國家在對於遊戲業界的投資與遊戲人才的培育方面，都下了相當程度的努力。其中電玩大國首推日本，日本方面，早在二十年前，任天堂剛發售的時候，日本就開始慢慢注意電玩遊戲這個產業，由於任天堂的轟動，電玩市場的高獲利，使得各家廠商都投下了心力，去開發電玩的硬體以及軟體。

主要發售的主機依硬體廠商來分有：

Nintendo(任天堂)：

FAMICOM (1983/07/15)
GameBoy (1989/04/21)
SUPER FAMICOM (1990/11/21)
VIRTUAL BOY (1994/11/21)
Nintendo (1996/06/23)
GameBoyAdvance
GameCube

SEGA：

MEGA-DRIVE (1988/10/29)
GAME GEAR (1990/08/08)
MEGA-CD (1991/05/08)
SEGA-SATURN (1994/11/21)
DREAMCAST

NEC：

PC-ENGINE (1987/10/30)
PC-GT (1990/12/01)
DUO (1991/09/21)
PC-FX (1994/12/23)

SNK：

NEO-GEO (1990/3/27)
NEO-GEO-CD (1994/09/09)
NEO-GEO-PARKET-COLOR

SONY :
PLAY-SATION(1994/12/03)
PLAY-SATION2

Microsoft :
X-BOX

主要的軟體廠商有：

Atlus(亞特拉斯)
Bandai(萬代)
BANPRESTO(帕布雷斯特)
Capcom(卡普空)
SQUARES · ENIX(史克威爾 · 艾尼克斯)
KOEI(光榮)
KONAMI(科拿米)
namco(拿姆科)
Nintendo(任天堂)
NEOGEO
SONY
TECMO
HUDSON

而歐美方面，因為遊戲的風格難以使亞洲人接受，所以在亞洲市場發售的情況並不是很好，除了主要的幾款強作之外，就很難見到歐美的遊戲出現在台灣市場了。

韓國則是這幾年，以線上遊戲異軍突起，以數款強力的線上遊戲，吸引了數以萬計的玩家加入，不只是韓國，連日本跟台灣，都有許多玩家加入，而單機遊戲則很少見。

在遊戲的人才培養方面，日本方面早在幾年前，就有許多家遊戲廠商，成立多媒體遊戲學院，以培養遊戲開發人才，由遊戲公司投資的電玩學院，學生們畢業後就可以直接進入其公司工作，也有很多獨立的電玩學院，沒有遊戲公司投資，讓學員們可以自己決定畢業後的

出路，所成立的科系，也依照遊戲的製作過程，設立了很多細節專業科目科系，例如：遊戲企劃學系、音樂與音效學系、遊戲程式學系、遊戲情報與宣傳學系、遊戲劇情編劇學系等。

美國方面，也有很多遊戲學校，剛開始是由日本 Nintendo(任天堂)的美國分公司在美國的任天堂公司本部裡面成立了學校，之後就有愈來愈多的遊戲學校建立。

韓國則是最近開始很多有關遊戲的相關事業出現，像是職業的遊戲玩家，遊戲的學校也愈來愈多，但是主要還是把重點放在線上遊戲居多，對於單機遊戲的接觸還是很少。

在遊戲風格方面，也有很大的差異，歐美遊戲的風格大多比較灰暗，血腥、暴力的成分也佔了絕大部分，而且歐美的 RPG 遊戲大多以任務方式進行，對話也不多。

相較於歐美，日本的遊戲則非常的多樣化，傳統的遊戲類型數量繁多已經不在話下，而新的遊戲類型也不斷的推成出新，遊戲的遊玩方式令人驚奇，任何在身邊發生的事，或是天馬行空的幻想，都可能成為遊戲。

韓國則是有別於上述二國，各式各樣的線上遊戲層出不窮，在線上遊戲的開發與經營，經驗還大過於日本，但是在單機遊戲，韓國所推出的，都會在其遊戲中都會發現日本遊戲的影子。

至於台灣，在產業部份，因為一直以來，電玩被視為罪惡的根源，說到電玩就好像是壞事，一直到現在，其實還是有很多家長這樣的認為，但是事實真的是這樣嗎？在其他國家的電玩業發展中，就可以知道答案。

台灣的電玩業，一直到這幾年，才慢慢的有起色，在 1995 年 12 月 25 號，聯華電子曾經發售過一台完全中文的遊戲主機 A'CAN，雖然主机的性能，比起同時期的日本遊戲主機，有過之而無不及，但是遊戲內容卻是差了一大截，很明顯的抄襲，而且畫面奇差無比，導致在 1996 年 4 月，A'CAN 慘遭停產，發售的情況慘不忍睹。

這幾年台灣因為沒有電玩主機，所以遊戲公司的開發主力，全部都在個人電腦上，在眾多公司的努力之下，能稱為名作的遊戲也出現

了幾部，但是也只有這幾部名作，展現出台灣公司的功力，很多台灣遊戲，可能畫面很好，但是常常是創意缺缺，音效草率，常常可以找出一個一模一樣的日本遊戲，明顯的抄襲，讓人擔心台灣遊戲業界的未來。

而現在台灣遊戲公司的主要利益，還是來自於由國外代理的線上遊戲，至於單機遊戲，由於台灣的盜版猖獗，所以單機遊戲的販賣就更加困難了。

在 X-BOX 的發行之後，台灣的遊戲公司，開始有機會將遊戲推到電玩主機上去，因為 Microsoft 所發行的 X-BOX 跟個人電腦一樣，都是 WINDOWS 系統，而且因為 Microsoft 以重金砸出性能優越的硬體主機，賠錢的銷售於日本國內，使得日本國內的遊戲業界面臨很大的挑戰，突然的強敵，但是其實 Microsoft 在電視遊樂器業界，沒有經驗，所以在遊戲軟體的開發方面，則是用重金挖角各家軟體廠商，但是大部分廠商還是不願意跳槽，也因此日本人大半都不接受 X-BOX，也因此 X-BOX 在日本的銷售量，比起預期的相差甚多，也使得 X-BOX 的軟體開發廠商少，Microsoft 極力的想請各大軟體廠加入，而台灣的遊戲公司也被邀請，去為 X-BOX 開發遊戲，熟悉的 WINDOWS 系統也讓台灣廠商開發簡單了許多，雖然抄襲的習慣還是不改但是最近有幾家新的遊戲公司卻是讓人感覺可以期待。

1.3 遊戲美工的地位

在一個遊戲中，美工可以說是整個遊戲的外貌，在我們看到一個遊戲的時候，第一眼就是看到美工的展現，遊戲的畫面是不是好看，遊戲的宣傳畫面是不是特別，人物畫像是不是順眼，這些由美工努力的成果，往往決定一個遊戲的生死。

很多一般玩家說，遊戲，畫面一定要漂亮，要炫，至於遊戲的內容，是不是有深度，是不是有劇情，操作性是不是好，這些都無所謂了；但是資深玩家說，遊戲最重要的是內容，是不是有劇情、有深度，遊戲故事的背後，是不是讓我們想起了什麼事，發人省思，操作性是不是容易上手，遊戲性是不是很耐玩，至於畫面，雖然不是完全無所謂，但是並不是太重要。

就我來說，我覺得，遊戲的每一部份都重要，就劇情來說，並不

是任天堂遊戲，那短短幾天就可以玩透的遊戲劇情就是不好，現在的次世代遊戲，玩上幾個月還不一定玩的透，才叫做好劇情；就畫面來說，並不是任天堂時代的方格人，如此粗糙的畫面就是不好，也不是現在的遊戲，已經分不清楚是真人還是CG動畫，這樣精緻的畫面就叫做好。

好的劇情與畫面，是依照當時的時代背景，可以將主機的性能發揮到淋漓盡致的遊戲，就可能可以是好的畫面與劇情，當然還要就主機的極限來說，算是漂亮的畫面，算是有深度，而且劇本長度也剛好的，才算是好。

所以美工在遊戲中的地位，就像是遊戲的化妝師，對於一個遊戲的存亡，掌握著決定性的關鍵，台灣現在很多的遊戲就是缺乏美工，所以遊戲不討喜，賣相也不好，畫不出好的人物畫，就用很一般的3D人物圖濫竽充數，實在是不可取。

其實台灣美術方面的科系也不少，只是遊戲業界的美工人數還是十分稀有，希望在現在台灣開始有遊戲科系的同時，美工的人才也可以出頭。

1.4 美工製作過程

現今市面上有各式各樣的繪圖軟體，種類繁多，可供有各種需求的消費者挑選、購買。Adobe公司所開發的photoshop是其中最受歡迎的繪圖軟體之一，我們製作美工圖片所使用的，就是這個擁有廣大支持者的繪圖軟體，以及Adobe Photoshop所附，製作動畫的軟體ImageReady。

製作任何美工圖片都是從設計草稿開始。設計時，先在紙上勾勒出適合主題的草稿，最好多畫幾種不同類型的設計稿，在作畫時可以多想些創新的點子，並融合到畫作之中。不要有“一定必須跟XXX一樣才行”的念頭，這樣常會拘泥在舊形式之中，就算畫的再好，明眼人一看就知道“這是學XXX的”。要根據主題設計出有其獨特個性的人物，人物的特徵、個性、動作和色彩，無一不影響這個人物的形成。舉例來說，精靈的造型就會和矮人相差甚多。在外型上賦予人物不同的靈魂，是美工人員重要的任務。

另外，考慮實際製作出人物時，可能會遭遇的困難也是很重要的。

有時，某種長度的頭髮或者衣服的造型，會在角色人物在行走，或作某些動作時會有意外的情況出現。例如：當頭髮的長度超過肩膀時，在人物側面的行走圖中，因行走時雙手的擺動，就會發生有時手臂被頭髮遮住，有時頭髮又被手臂遮住的現象(因疊放髮型的圖層是在衣服和模型的圖層之上)。為避免不該出現的現象發生，在設計草稿時就必須審慎的思考，並再三考慮，才能使發生的機率降到最低。在畫好自己滿意的草稿後，可以試著將這些草稿給其他的組員看看，聽看看其他人的看法，這樣比較容易製作出較讓大眾接受的人物，而不會流於美工人員個人的偏好。而且跟大家一起討論，集合眾人的意見，也許還會激出更好的 idea。

在決定好要用的草稿後，就可以開始製作實際要用的角色圖片。一般美工在電腦上畫圖時，都是使用專業的繪圖板。繪圖板的規格愈大，畫出來的線條就越精細，但是相對的繪圖板的價格也就愈高。一開始先是製作人物的模型圖，也就是沒穿衣服，沒頭髮的人物圖。因為要畫出人物行走時的圖片，除了平常要多觀察人走路的样子之外，選出最佳的關鍵動作也是很重要的。分解動作的圖片，合成人物行走的動作，以同樣長度的動畫來說，有愈多張影格達成的效果愈好，一般來說每秒 24 張影格，就可以組成十分順暢的動畫了，不過在製作遊戲時因考慮硬體大小、運算速度…，可能會以減少影格的數目來達到其他的目的。

人物動作流暢固然很重要，但若不符合自然界的物理原理，就會變的奇怪而不自然。我們可以利用運動學針對像人這樣有關節的結構，對所有關節和以關節連結的部位，例如膝蓋彎曲、臀部放鬆、肩膀擺動與頭部上下運動等，計算出他們的位置、轉動速度與加速度，並致力於呈現這些相對的位置與運動。要記得雖然美工畫出的圖形是平面 2D 的，但所描寫的物體卻是在 3D 空間運動的。

在製作這些圖片期間，可以使用 ImageReady 來跑跑看製作的動畫畫面，藉此挑出不夠完善的缺點和瑕疵，使組成的動畫達到預期的標準。而且一邊製作一邊測試，比較容易避開大的問題，在開發初期就可以解決較大的缺失，較能夠避免在開發後期時，發現難以挽回錯誤，而造成太多時間和人力的浪費。

製作完成人物模型後，我們可以就此模型來製作人物的造型了。電腦繪圖軟體有個好處，就是可以將圖分成多個圖層來作畫，這樣我們就不會因製作或修改新的圖層，而破壞其他已完成的圖層。接著我

們就用這種方法，新增圖層來繼續製作服裝和髮型圖片。製作髮型時，除了頭髮的造型之外，顏色的搭配也是不可輕忽的，這點在設計服裝時也是一樣的。配色的原則有均衡、漸層、強調、律動與支配等五項，主要的目的是要使色彩有整體的和諧美。而且使用的顏色特性必須符合角色的印象，我們不能將一個須給人陰沉、冷漠印象的人，配上粉紅色的洋裝。此外，人物服裝的圖跟模型一樣，走路時衣衫的晃動、拉扯、扭曲或型變，都必須符合物理定律。

在製作完所有的圖形後，不要忘記進行測試，除了開發的美工人員之外，最好也請其他多一點人，最好是有經驗的老手來幫忙測試。有些小缺點可會因為開發者本身的經驗不足、先入為主的觀念或種種因素而被忽略。這時其他的測試者，往往可以提供較客觀的意見和測試方法，來找出這些缺點。例如美工人員在作畫時，使用的是透明圖層，如果空白處不小心點上一個白點，而透明層的預設底色又正好是淺色系，這個白點就很容易被忽略。但有經驗的人就會知道，在測試時將底色設成黑色，或其他深色就可以很容易找出這個白點來了。測試完修改，修改完再測試，反覆這個過程到圖片沒有錯誤後，我們就完成了美工製作。最後所得到的結果，就是我們所要的成品了，接下來，就可以使用這些圖片進行各式各樣的組合，來產生我們所需的人物角色，而美工的工作就到此告一段落。

1.5 遊戲類別簡介

遊戲基本大概可以分成以下幾種類型：

FTG(FIGHT GAME) 對戰格鬥：以兩人對打格鬥來進行遊戲



圖 1.1 STREET FIGHTER ZERO3



圖 1.2 DEAD OR ALIVE

ACT(ACTION GAME) 動作過關：由一人或是兩人以過關對抗敵人進行



圖 1.3 惡魔城年代記



圖 1.4 屠龍勇士

RPG(ROLE PLAY GAME) 角色扮演：以一人為主，玩家以主角身分，體驗各式中的世界



圖 1.5 DRAGON QUEST VII



圖 1.6 STAR OCEAN 2

A V G (ADVENTURE GAME) 冒險動作：以第三人稱視點進行 3D 的動作遊戲，或是文字遊戲



圖 1.7 BIO HAZARD 3



圖 1.8 GetBackers! 奪還屋

S L G (SIMULANT GAME) 戰略模擬：以棋盤式戰鬥或是模擬設施建設



圖 1.9 FINAL FANTASY TACTICS



圖 1.10 德貝賽馬 99

S L G (SIMULANT GAME) 戀愛養成：以第一人稱，進行養成或是戀愛遊戲



圖 1.11 SISTER PRINCESS



圖 1.12 育嬰問答

T A B (TABLE GAME) 桌上遊戲：以方塊，大富翁為主



圖 1.13 骰子方塊 XI



圖 1.14 魔法氣泡

R T G (REAL TIME GAME) 即時戰略：多人即時線上模擬戰略遊戲



圖 1.15
Warcraft III:Reign of Chaos



圖 1.16
Medieval: Total War

R C G (RACE GAME) 賽車競速：以車子或是其他物體(人物)競速



圖 1.17 Ridge Racer 4



圖 1.18 爆走貨車傳說

S P G (SPORT GAME) 運動競賽：各類運動比賽的遊戲



圖 1.19 實況野球 2001



圖 1.20 WINNING ELEVEN 2002

S T G (SHOOT GAME) 空戰射擊：以飛機或其他物體(人物)飛行射擊



圖 1.21 空戰奇兵 3



圖 1.22 RAYCRISIS

E T C (ETC GAME) 其他：其他無法歸類的類型



圖 1.23 我的料理



圖 1.24 柏青哥王國 2

第二章 系統架構

2.1 系統簡介

這個系統可以將硬碟中的圖片檔案存入資料庫中，再將存放在資料庫中的圖片分類，加以組合，用人物產生器產生各種組合而成的角色，讓使用者可以方便的比較這些組合，以找出最適合的組合方式，再將此組合存入資料庫中，方便以後的管理及使用。

2.2 人物自動產生器

對於人物自動產生器，我們所設計的方式，是針對 2D 的 RPG 遊戲來製作，大部分 2D 的 RPG 遊戲的人物行走圖形，都是以 2~3 頭身的圖，在地圖上走動，而方向則有幾種方式，有斜 45 度角的四方向或是八方向走動，也有水平的左右兩方向走動，而最常見的就是正面的四方面或是八方面的走動。

也因此大部分的圖，尤其是同一個遊戲公司或是小組，更是有許多重複的圖，所以這樣的人物產生器，就可以減輕許多美工的工作；而我們將人物圖形分成三個部份，人體的模型、衣服身上配件、頭髮與頭上配件，因為在 RPG 的行走圖中，2~3 頭身的人物，如果分的太細，反而會讓圖形難以管理與組合，所以將人物分成三個部份，再用自動化加以組合展示。

在人物產生器的製作方面，美工在繪圖時，則是將角色分成這三個部份來繪製，將除了原件本身的其餘部分透明化，以方便圖形的組合；組合時依照順序(模型→衣服→髮型)，將三張圖疊在一起，以完成組合，再以 9 張圖一頁的方式展現給使用者看，方便加以比較、選擇。

2.3 資料表設計

人物面向的方向：

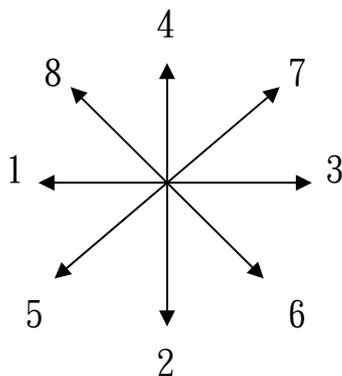


圖 2.1 人物面向的方位

- 1：面向左方
- 2：面向前方
- 3：面向右方
- 4：面向後方
- 5：面向左前方
- 6：面向右前方
- 7：面向右後方
- 8：面向左後方

我們所建立的資料庫名稱為 CharacterImage，其中一共包括五個資料表：

- 1) Model(模型資料表)：標準 ID 欄位的編碼方法是 M000000001。一共有 36 個欄位。
- 2) Clothes(服裝資料表)：標準 ID 欄位的編碼方法是 CL00000001。一共有 36 個欄位。
- 3) Hair(髮型資料表)：標準 ID 欄位的編碼方法是 HA00000001。一共有 12 個欄位。
- 4) Character(角色資料表)：標準 ID 欄位的編碼方法是 CH00000001。一共有 6 個欄位。
- 5) Group(群組資料表)：標準 ID 欄位的編碼方法是 GR00000001。一共有 2 個欄位。

在所有資料表中的 ID 欄位皆為索引鍵，所以其欄位的值皆不可為 null，也不可以出現相同重複的值。Mode、Clothes 與 Hair 資料表中

的 group 欄位是用來區分此資料列所屬的群組，系統只允許相同群組的資料列中的圖可以互相組合，這樣可以避免不同樣式的圖片混合在一起，造成組合出不能使用的圖片。另外 Mode 和 Clothes 資料表中的 image 欄位，例如：image1-1、image1-2、image1-3 和 image1-4 是表示面向（參照圖 1）方向 1（“1”就表示面向左）的四張分解圖（每個方向的行走分解圖都分別有四張）。

Model--模型資料表

欄位名稱	資料類型	資料長度	說明
ID	char	10	索引鍵，此資料列的編號
name	char	30	此資料列的名稱
sex	bit	1	此模型適用的性別
group	char	10	此資料列所屬群組的 ID
image1-1	image	16	面向左方的圖
image1-2	image	16	面向左方的圖
image1-3	image	16	面向左方的圖
image1-4	image	16	面向左方的圖
image2-1	image	16	面向前方的圖
image2-2	image	16	面向前方的圖
image2-3	image	16	面向前方的圖
image2-4	image	16	面向前方的圖
image3-1	image	16	面向右方的圖
image3-2	image	16	面向右方的圖
image3-3	image	16	面向右方的圖
image3-4	image	16	面向右方的圖
image4-1	image	16	面向後方的圖
image4-2	image	16	面向後方的圖
image4-3	image	16	面向後方的圖
image4-4	image	16	面向後方的圖
image5-1	image	16	面向左前方的圖
image5-2	image	16	面向左前方的圖
image5-3	image	16	面向左前方的圖
image5-4	image	16	面向左前方的圖
image6-1	image	16	面向右前方的圖
image6-2	image	16	面向右前方的圖
image6-3	image	16	面向右前方的圖
image6-4	image	16	面向右前方的圖

image7-1	image	16	面向右後方的圖
image7-2	image	16	面向右後方的圖
image7-3	image	16	面向右後方的圖
image7-4	image	16	面向右後方的圖
image8-1	image	16	面向左後方的圖
image8-2	image	16	面向左後方的圖
image8-3	image	16	面向左後方的圖
image8-4	image	16	面向左後方的圖

Clothes—服裝資料表

欄位名稱	資料類型	資料長度	說明
ID	char	10	索引鍵，此資料列的編號
name	char	30	此資料列的名稱
sex	bit	1	此服裝適用的性別
group	char	10	此資料列所屬群組的 ID
image1-1	image	16	面向左方的圖
image1-2	image	16	面向左方的圖
image1-3	image	16	面向左方的圖
image1-4	image	16	面向左方的圖
image2-1	image	16	面向前方的圖
image2-2	image	16	面向前方的圖
image2-3	image	16	面向前方的圖
image2-4	image	16	面向前方的圖
image3-1	image	16	面向右方的圖
image3-2	image	16	面向右方的圖
image3-3	image	16	面向右方的圖
image3-4	image	16	面向右方的圖
image4-1	image	16	面向後方的圖
image4-2	image	16	面向後方的圖
image4-3	image	16	面向後方的圖
image4-4	image	16	面向後方的圖
image5-1	image	16	面向左前方的圖
image5-2	image	16	面向左前方的圖
image5-3	image	16	面向左前方的圖
image5-4	image	16	面向左前方的圖
image6-1	image	16	面向右前方的圖
image6-2	image	16	面向右前方的圖

image6-3	image	16	面向右前方的圖
image6-4	image	16	面向右前方的圖
image7-1	image	16	面向右後方的圖
image7-2	image	16	面向右後方的圖
image7-3	image	16	面向右後方的圖
image7-4	image	16	面向右後方的圖
image8-1	image	16	面向左後方的圖
image8-2	image	16	面向左後方的圖
image8-3	image	16	面向左後方的圖
image8-4	image	16	面向左後方的圖

Hair—髮型資料表

欄位名稱	資料類型	資料長度	說明
ID	char	10	索引鍵，此資料列的編號
name	char	30	此資料列的名稱
sex	bit	1	此髮型適用的性別
group	char	10	此資料列所屬群組的 ID
image1	image	16	面向左方的圖
image2	image	16	面向前方的圖
image3	image	16	面向右方的圖
image4	image	16	面向後方的圖
image5	image	16	面向左前方的圖
image6	image	16	面向右前方的圖
image7	image	16	面向右後方的圖
image8	image	16	面向左後方的圖

Character—角色資料表

欄位名稱	資料類型	資料長度	說明
ID	char	10	索引鍵，此資料列的編號
name	char	30	此角色的名稱
sex	bit	1	此角色的性別
Model	char	10	此角色使用模型的 ID
Clothes	image	16	此角色使用服裝的 ID
Hair	image	16	此角色使用髮型的 ID

Group—群組資料表

欄位名稱	資料類型	資料長度	說明
------	------	------	----

ID	char	10	索引鍵，此資料列的編號
name	char	30	此群組的名稱

資料關聯圖

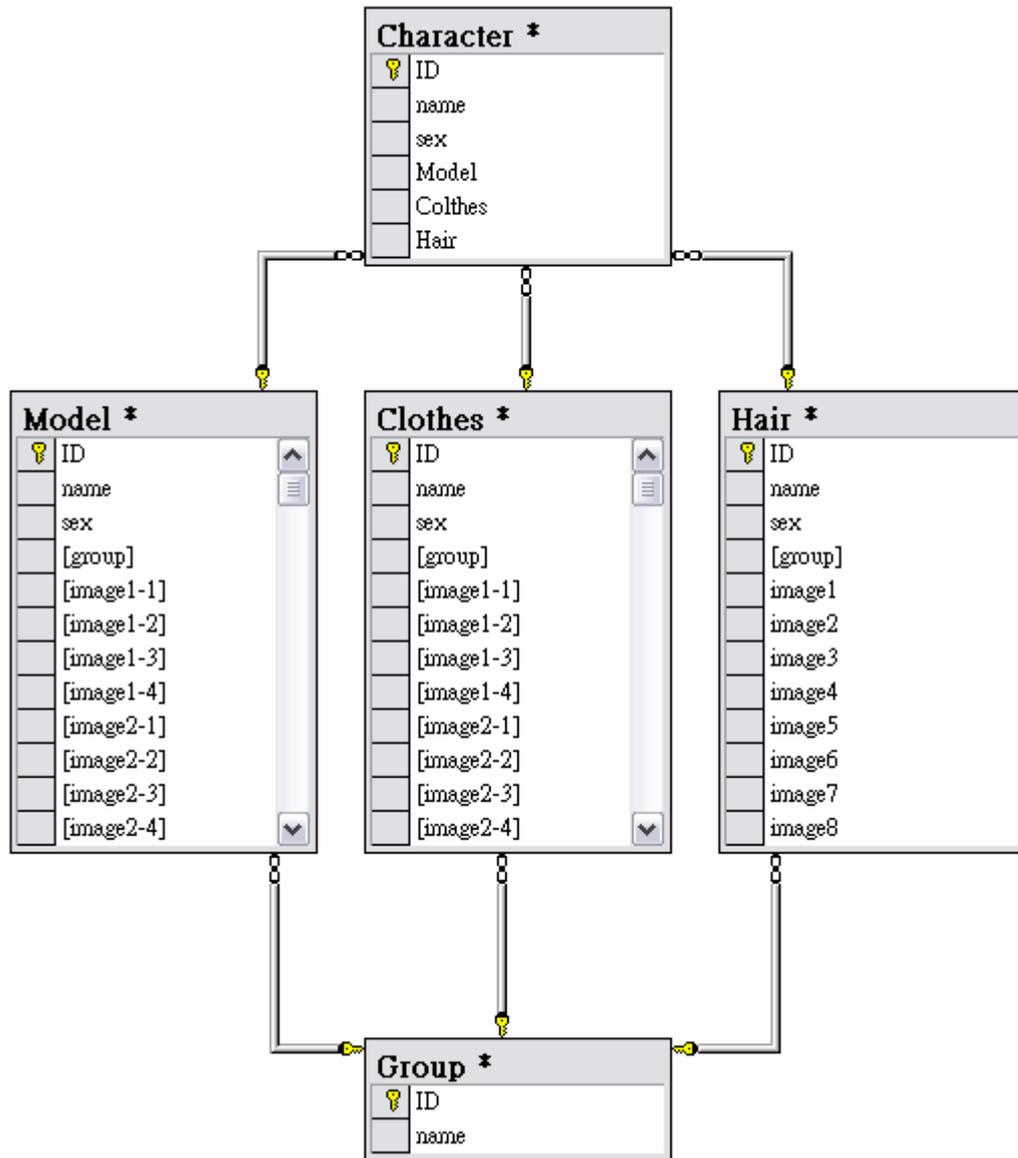


圖 2.2 資料關聯圖

第三章 系統實作

3.1 發展環境

軟體方面，我們開發系統所使用的平台是現今最多人使用的作業系統之一 Microsoft 公司的 Windows 中的 Windows XP，而資料庫的建立是使用 Microsoft 公司的 SQL Sever 2000，使用的電腦語言是 SUN 公司開發的 JAVA 語言，撰寫程式所用的編輯器是 Borland 公司的 JBuilder 9。硬體方面，CPU 速度為 2.4G，硬碟大小為 40G，RAM 的大小是 256MB。

我們之所以選擇這些工具是因為我們認為，這些工具在開發我們的專案上，是市面上所有工具中，可以幫助我們的專案達到最佳成果的工具。

3.2 使用 JDBC 連線

在 Java 的函式庫中有一組專門用來處理和資料庫相關的 API，稱之為 JDBC(Java Database Connection)，JDBC 提供的功能是讓 Java 語言與資料庫進行溝通。JDBC 並且提供一個讓程式設計師及資料供應商，依循同樣的標準來做資料存取的介面。JDBC 主要的目的有：

- 1) 讓 Java 程式設計師，發展單一的資料庫存取方式。這樣 Java 的程式設計師就可以利用 JDBC，方便、有效地存取資料庫中的資料。所以 Java 的程式設計師只要知道使用 JDBC 的方法，並不需要了解資料供應者是如何實做資料存取以及資料庫連結，就可以利用 JDBC API 來與資料庫溝通，進行資料庫中資料的存取及資料庫連結。
- 2) 不同的資料供應廠者可以利用 JDBC 的標準，來實作同樣資料存取的方式。換句話說，雖然資料來源不相同的，我們卻可以利用相同的方式來存取資料。資料供應商也可以根據 JDBC 的規格專心於實做的部分，提供 JDBC 的驅動程式，不用擔心用戶端 (client) 接收的部分。

有了 JDBC API 統一的介面，這 Java 程式設計師和資料供應廠者之間就能夠輕易的進行溝通。JDBC 的基本概念如圖 3.1 所示。

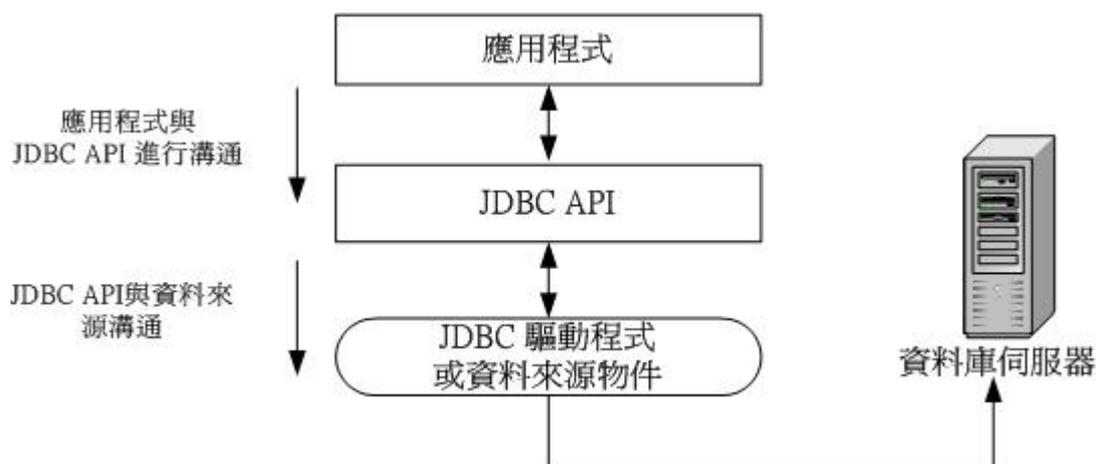


圖 3.1 JDBC 連線方式

JDBC API 可分為兩個部分：

- 1) 用戶端(client)：連結資料庫及存取資料庫。程式中必須宣告 `java.sql`。
- 2) 伺服器端 (server)：增加伺服器的功能。程式中必須宣告 `java.sql`。

JDBC 主要的功能：

- 1) 與資料庫連線。
- 2) 傳送 SQL 指令給資料庫。
- 3) 接收及處理資料庫所執行的結果。

JDBC 完成上述事件相對的元件類別：

- 1) 利用 `Connection` 類別來建立與資料庫的連線。
- 2) 利用 `Statement` 類別傳送 SQL 指令給資料庫，以存取資料庫中的資料。
- 3) 利用 `ResultSet` 類別來接收與處理執行結果。

目前 JDBC 根據資料庫連接的方式，將驅動程式 (Driver) 分成四種類型：

- 1) Type1：JDBC-ODBC Bridge plus ODBC Driver
- 2) Type2：A native-API partly Java technology-enabled Driver
- 3) Type3：Pure Java Driver for Database Middleware
- 4) Type4：Direct-to-Database Pure Java Driver

其中Type1與Type2的驅動程式是屬於同一類型，這兩類都是透過 JDBC Bridge的方法，利用現有與資料庫連線的函式，來達成和資料庫連結的目的。另外Type3及Type4的驅動程式是利用Java語言寫的資料

庫連線函式，也就是說這兩類是屬於Pure Java的JDBC驅動程式。一個Java的程式設計師必須要清楚了解這四種類型驅動程式的運作方式，和使用這些驅動程式的方法，以及哪一種驅動程式適合用來連結哪一種資料庫。由於使用者常常對資料庫有不同的需求，所以必須使用各種不同的資料庫，來配合使用者的需要。而且在不同場合也必須配合使用不同類型的驅動程式來連接資料庫，例如：Applet就要選擇Type3的驅動程式。

在建立我們的資料庫連線時，我們所使用的是Type1的驅動程式。首先要先建立與資料庫ODBC API的連線，ODBC API與JDBC API類似，也是一種連結資料庫的方法，但若要與Java進行溝通就必須再經由JDBC-ODBC Bridge的驅動程式，這樣就可以透過ODBC連線連結資料庫。Type1的驅動程式是透過與ODBC的連接來和資料庫進行溝通(JDBC-ODBC Bridge)，也就是JDBC會將所有與資料庫溝通的SQL指令經由ODBC處理，由ODBC來負責與資料庫溝通，Type1驅動程式的概念如圖3.2所示。

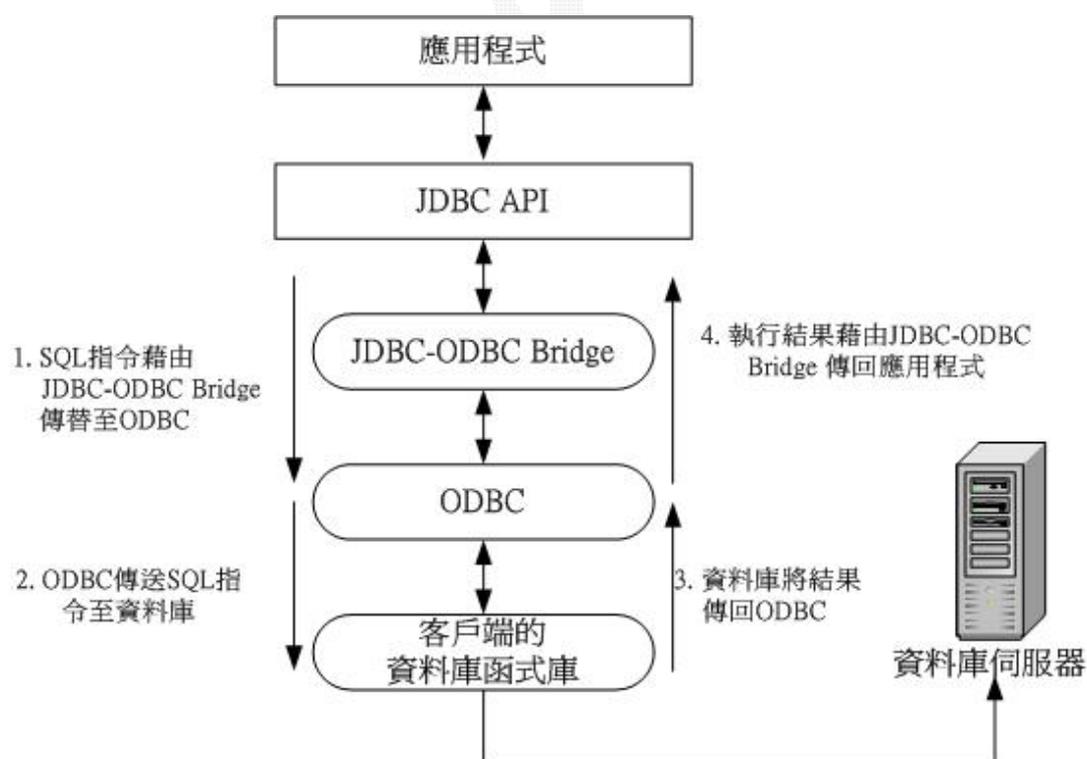


圖 3.2 Type1 驅動程式連線方式

圖3.2說明了JDBC-ODBC Bridge plus ODBC Driver的連線方式。應用程式要與資料庫溝通的方式是，首先透過JDBC API，接著JDBC API會藉由JDBC-ODBC Bridge來與ODBC API進行溝通，然後最後再由ODBC來溝通資料庫。由於JDBC API內部就已經有支援Type1驅動程式的方

法，所以我們可以不用考慮太多驅動程式連結資料庫的問題，只要完成 ODBC 的設定就可以了。

其優點為：這種類型由於設定很方便的關係，是一開始使用 JDBC 連線時不錯的方式。其他的好處還有，一些使用者他們所使用的電腦上都已經設定 ODBC 的連線，我們就可以直接使用 ODBC 來連結資料庫，不用再做連線的設定。另外可能有些舊有的資料庫或應用程式，沒有支援其他的連結方式，必須要用這些 ODBC 的方法來連結資料庫，來能存取資料庫和運作程式。

缺點為：如果有多數的使用者要來使用這個應用程式，或這個資料庫必需讓多個使用者存取時，每一個使用者的電腦就必須都要設定 ODBC 連接。由於維護這些 ODBC 的設定，可能需要花許多時間及人力來處理，當專案很大或是使用者很多，例如線上開放的資料庫，此時維護的 ODBC 方式的連線是一件辛苦的事情。

我們建立資料庫連線是用，JDBC-ODBC Bridge plus ODBC Driver 的方法來連結資料庫。因為微軟的 Windows 現今最常被使用的作業系統，在 Windows 的系統內都有提供使用者建立資料來源方式的介面，可讓使用者簡單、快速的建立連線，而且介面操作起來十分友善。反觀其他驅動程式的連接方式，都沒有像這樣方便的介面，若不熟悉設定連結的方法，可能會只因為一些小錯誤，而造成整個連線癱瘓。另外，因為我們使用資料庫的人員不可能達到千人之多，所以在維護 ODBC 連線方面，應該不會太過困難，或是麻煩的事。因此在綜合比較後，還是決定使用 JDBC-ODBC Bridge plus ODBC Driver 的方法來連結資料庫。

3.3 連結資料庫

連結資料庫所用的類別：

- 1) Class
- 2) DriverManager
- 3) Connection

流程說明：

- 1) 首先利用 Class 類別的方法 `forName`，來載入資料庫使用的 Driver。例如：
`class.forName("sun. jdbc. odbc. JdbcOdbcDriver")。`

- 2) 成功載入 Driver 的類別之後，Class.forName 會向 DriverManager 註冊這個類別，接著我們就可以透過 DriverManager 中的靜態方法 getConnection 與資料庫建立連接。例如：

```
Connection con = DriverManager.getConnection( "jdbc:odbc:servername" , "username" , "password" );
```
- 3) getConnection 會回傳一個 Connection 物件(如上述的 "con")，接下來就可以利用這個 Connection 物件來與資料庫進行溝通。

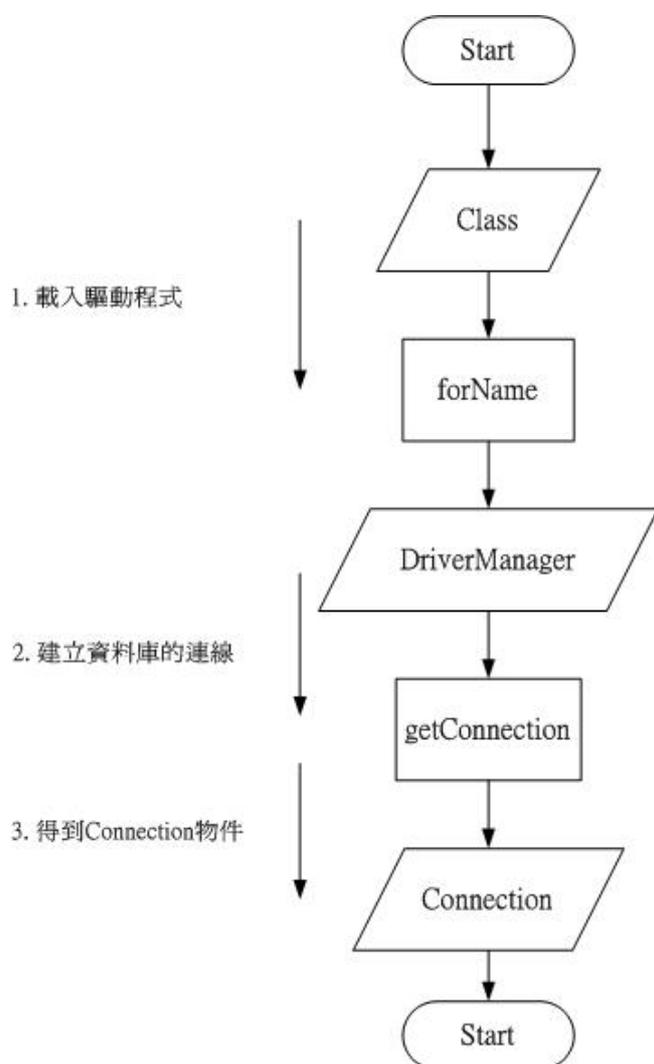


圖 3.3 ODBC 聯結資料庫流程

3.4 存取資料庫

存取資料庫所用的類別：

- 1) Statement

2) ResultSet

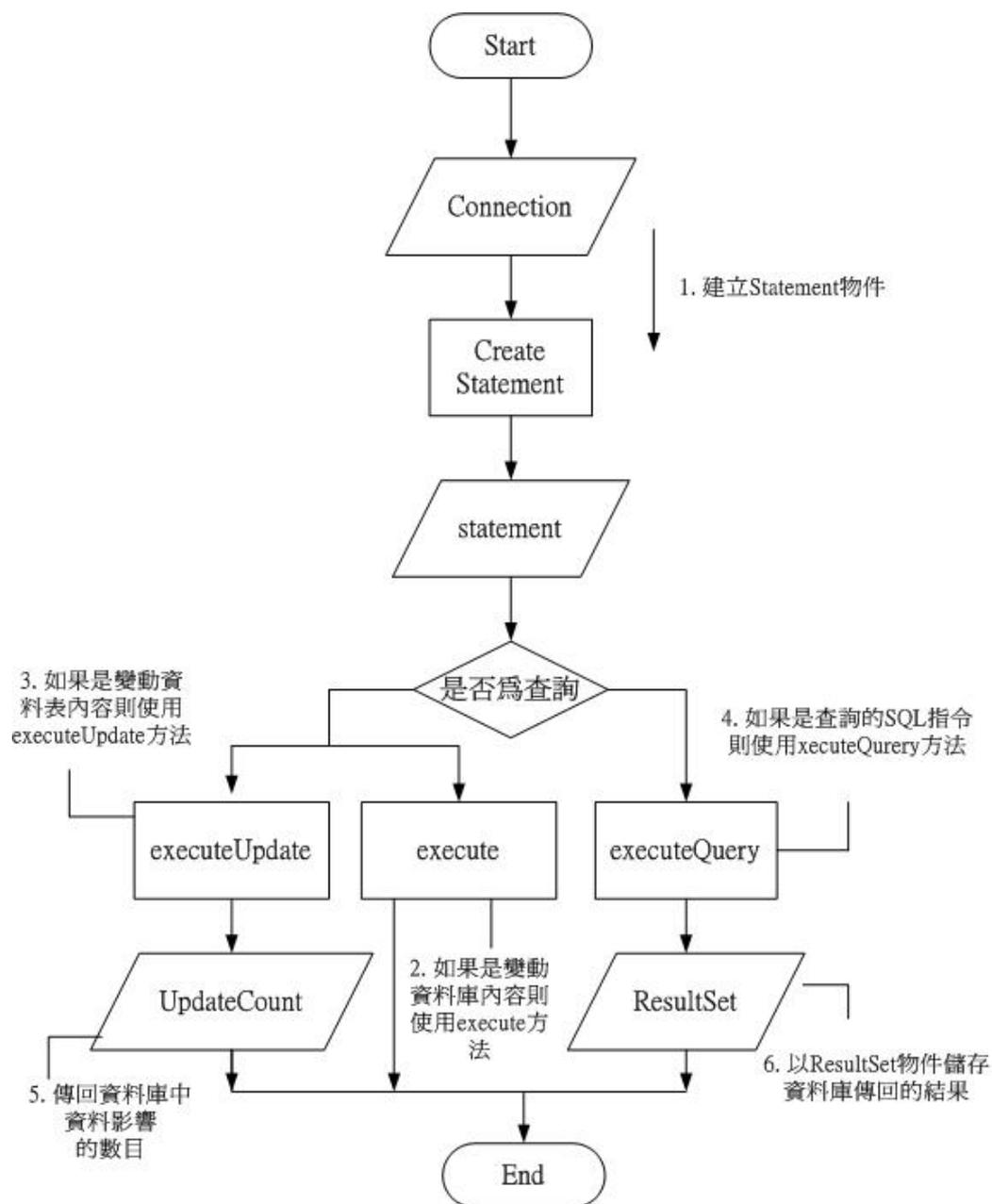


圖 3.4 存取資料庫的流程

流程說明：

- 1) 首先將利用 Connection 物件的 createStatement 方法，建立 Statement 物件。例如：`Statement sta = con.createStatement();`
- 2) 利用 Statement 物件的 execute 方法、executeUpdate 方法、或是 executeQuery 方法來執行 SQL 指令。如果是新增、修改或是刪除資料表或資料庫內容的 SQL 指令，使用 execute 方法。如果是新增、修改或是刪除資料列內容的 SQL 指令，使用

executeUpdate 方法。而屬於查詢的 SQL 指令，則是使用 executeQuery 的方法。

- 3) execute 方法會回傳執行 SQL 指令所影響的資料表內容。
- 4) executeUpdate 方法會回傳執行 SQL 指令所影響的資料筆數。如更新的資料數目或是刪除的資料數目。
- 5) executeQuery 方法會回傳 SQL 指令產生的 ResultSet 物件。
- 6) 最後利用 ResultSet 物件取得查詢結果。

3.5 介面製作

介面製作的方面，在更換式的視窗以 JPanel 來製作，跳出式的視窗以 JDialog 來製作，而在版面管理方面，為求精確，則是採 XYLayout，將所有元件都一個一個以 XY 座標定位；在主程式中，以一個 CardLayout 來切換在其中的切換視窗，也就是 JPanel。

3.6 使用者介面

[RPG 遊戲角色人物圖形產生器暨資料庫] 登入畫面：

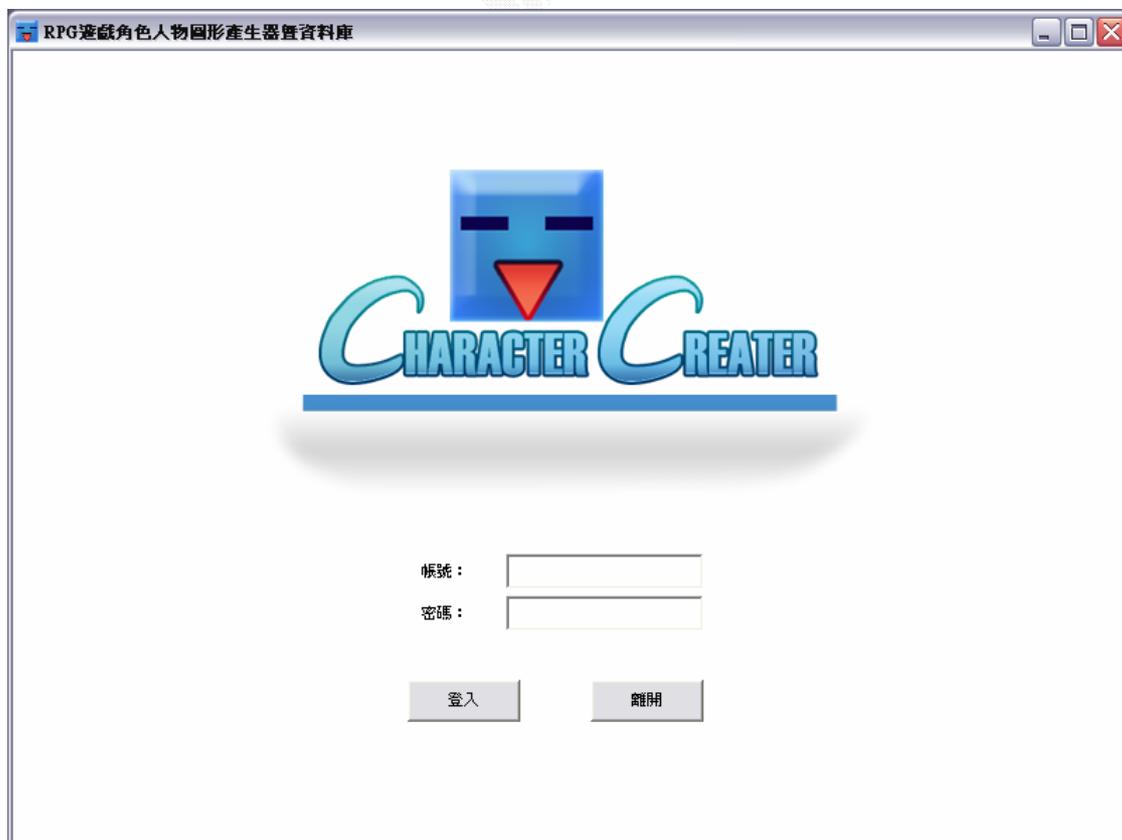


圖 3.5 登入畫面

登入畫面大小為 800*600，開啟此系統實的第一個畫面，可以防止非相關人士來濫用其中的資料，當美工人員依照帳號密碼登入後，即可到功能選擇畫面。



圖 3.6 密碼錯誤

當輸入的密碼錯誤時，即會跳出此視窗，按下確定，跳回登入畫面。

主選單：



圖 3.7 主選單畫面

登入畫面大小為 800*600，從登入畫面登入後，就到此畫面，這個畫面可以選擇四種類圖形的資料表，[角色資料表]、[模型資料表]、[服裝資料表]、[髮型資料表]、[群組資料表]、以及[樂、嵐]。

[角色資料表]就是可以將人物的模型、服裝、髮型排列組合，將所有的組合全部都展示出來，讓美工人員可以仔細挑選，選好之後可以存在資料庫裡，方便管理。

[模型資料表]人物的基本模組的資料表，模型也就是角色本體，未加上服裝、髮型的身體。(見 [模型資料表])

[服裝資料表]人物服裝的資料表，即是加在模型上的衣物、配件。(見 [服裝資料表])

[髮型資料表]人物髮形的資料表，加在模型頭上的頭髮，或是配件，帽子、髮夾等。(見 [髮型資料表])

[群組資料表]人物角色圖形的資料表，角色圖形群組，也就是已經組合好的人物角色圖形。(見 [群組資料表])

[樂、嵐]是可以就組合好的圖來操縱的小遊戲。

角色資料表：



圖 3.8 角色資料表

角色資料表的大小為 800*600。開啟角色資料表後，系統會在左邊顯示出所有的資料列。若想重新排列欄位順序，可用滑鼠按一下欄位名稱，例如：ID，這樣系統就會依 ID 的名稱順序重新排列資料列，可選擇來排列的欄位有：ID、name、sex 及 group。右邊的是功能選項，一共有 8 個功能：

- 1) 搜尋：可搜尋此資料表的資料。先選擇欲搜尋欄位的名稱(可提供搜尋的欄位有：ID、name、sex)，再於下方的空白欄位中輸入欲查詢的值，然後按下“搜尋”鍵，就會在左邊的 table 顯示出所有符合的資料列。若查無資料，左邊的 table 就不會顯示出任何資料。
- 2) 顯示全部：按下“顯示全部”左邊的 table 就會顯示出模型資料表中所有的資料列。
- 3) 新增角色：按下“新增角色”就會切換到“角色人物產生器”視窗。

角色人物產生器：

從角色資料表畫面選取“新增角色”，就到此人物圖形自動產生器畫面，畫面大小為 800*600，從右邊的“選擇模型圖片”、“選擇服裝圖片”、和“選擇髮型圖片”，就可以到各部位圖片選擇視窗。



圖 3.9 人物圖形產生器(未選取)

各部位圖片選擇：



圖 3.10 選擇模型圖片



圖 3.11 選擇衣服圖片



圖 3.12 選擇髮型圖片

點選“選擇模型圖片”、“選擇服裝圖片”、和“選擇髮型圖片”，就可以選擇想組合的各部位圖片，選好各部位的圖片之後，按下“組合”，就會產生所有的圖形組合。



圖 3.13 人物圖形產生器(已選取)

依照選取出來的圖，產生出所有可能的組合，讓使用者可以一一比較挑選。如果組合的結果超過九張圖，可以利用“上一頁”、“下一頁”來切換頁面。如果在選擇的時候難以抉擇，想先刪去不合適的圖片，只要勾選圖片，按下“刪除選擇項目”，就可以將不需要的圖片刪除。若要取消所選擇的所有資料，只要按下“全部清空”就可以重新開始選擇、編輯角色資料。



圖 3.14 人物圖形產生器(已刪除)

若想要將組合好的圖片存入硬碟可先勾選欲儲存的圖片，再選擇“儲存完成圖片”按鈕，畫面上就會跳出如圖 3.15 的視窗，讓使用者將圖片存到硬碟中。



圖 3.15 儲存完成圖片

當決定想要儲存的圖片之後，只要勾選該圖片，按下“儲存角

色”，就會跳出“新增角色資料”視窗。

新增角色資料：

輸入“ID”圖片的ID，“名字”角色名字，和“性別”角色性別，然後按下“存檔”，就可以將人物新增至資料庫中了。至於“模型”、“服裝”、和“髮型”欄位，會由產生器自動產生數值，存入資料庫中，使用者不用在費心思去記所選擇欄位的各個ID數值，只要點選所要的圖片，系統就會自動產生出來。



圖 3.16 新增角色資料

選擇人物產生器畫面中的“回上一頁”可回到“角色資料表”的畫面，系統會跳出如圖 3.17 的對話框提醒使用者。但要注意的是，回到上一頁後使用者所選擇的資料都會被清除掉，需特別小心。然後角色資料表的“新增角色”功能介紹全部就到此結束。

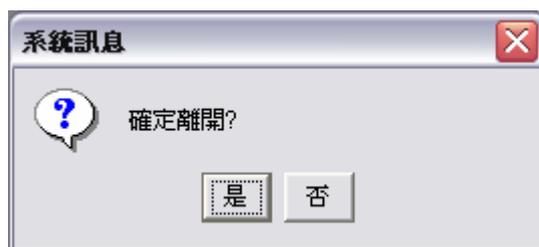


圖 3.17 確定離開

- 4) 刪除一行：跟“模型資料表”中的“刪除一行”大同小異，詳細請參閱“模型資料表”中的“刪除一行”功能說明。
- 5) 修改一行：可修改“ID”、“名字”、和“性別”等三個欄位。如圖 3.18 所示。



更新角色資料

ID: CH00000001

名字: 金髮祭司

性別: 男 女

模型ID: M000000001

服裝ID: CL000000001

髮型ID: HA000000003

回復 更新資料 離開

圖 3.18 更新角色資料



圖 3.19 組合結果

- 6) 顯示此組合圖片：可顯示預覽所選資料列中“model”、“clothes”、和“hair”欄位所組合的結果。另外，只要按下“儲存圖片”按鈕，使用者也可以從這裡將這個欄位的組合結果圖片，存入硬碟之中。
- 7) 存成樂、嵐用圖：可以將所選擇資料列組合成的圖，放到小遊戲“樂、嵐”中測試、使用。
- 8) 回主畫面：可回到“主畫面”的頁面中。在所有資料表畫面選擇“回主畫面”按鈕，皆會出現如圖 3.17 的對話框，確定使用者是否要離開這個畫面。

模型資料表：



圖 3.20 模型資料表

模型資料表的大小為 800*600。開啟模型資料表後，系統會在左邊顯示出所有的資料列。若想重新排列欄位順序，可用滑鼠按一下欄位名稱，例如：ID，這樣系統就會依 ID 的名稱順序重新排列資料列，可選擇來排列的欄位有：ID、name、sex 及 group。右邊的是功能選項，一共有 7 個功能：

- 1) 搜尋：可搜尋此資料表的資料。先選擇欲搜尋欄位的名稱(可提供搜尋的欄位有：ID、name、sex、group)，再於下方的空白欄

位中輸入欲查詢的值，然後按下“搜尋”鍵，就會在左邊的 table 顯示出所有符合的資料列。若查無資料，左邊的 table 就不會顯示出任何資料。

- 2) 顯示全部：按下“顯示全部”左邊的 table 就會顯示出模型資料表中所有的資料列。
- 3) 新增一列：按下“新增一列”後會跳出圖 3.21 的視窗，視窗大小為 800*600。使用者可在空白欄位中輸入欲建立的值。這些欄位除了 ID 欄位外，皆可為 null。ID 欄位旁的“尋找 ID”按鈕可自動幫助使用者找出資料表中最末資料列的遞增值。例如最末資料列的 ID 值為 MO00000009，按下此按鈕後系統就會把 ID 欄位填上 MO00000010。“性別”欄位有兩個選項，分別是“男”與“女”，這兩個選項一次者能選擇其一，系統並有機制可保持一次只有一個選項被選擇。例如先選擇“男”後再選擇“女”，“男”選項中被勾選的記號就會被自動取消，只顯示“女”選項被選擇的狀態。第四個欄位“群組”可用下拉式選單，來選擇現有的群組。這裡選單選項的資料來源是從資料庫中所擷取的，因此使用者只要一建立新的群組資料，就會立刻反映在此選單上，使用者就不用怕新建了群組卻找不到資料。



圖 3.21 增加模型資料

關於圖片路徑的部分，增加模型資料表的畫面上每一列共有四個欄位分別代表資料表中的 imageX-1、imageX-2、imageX-3 及 imageX-4(X 代表 1~8，是指圖片中人物面向的方向)，一共有八列。每一個欄位可利用“選擇檔案”按鈕從硬碟中讀取檔案資料如圖 3.22，可選擇的檔案類型只有.gif 檔，其他類型的資料我們的資料庫沒有支援。這是因為.gif 檔可支援圖層透明的方法，只有使用圖層透明的方法才能順利組合多張圖層。若使用沒有支援圖層透明的方法的檔案類型，例如.bmp 檔或.jpg 檔，背景將會被前景遮住，就無法顯示圖形組合的效果了。



圖 3.22 選擇硬碟中的圖檔

輸入欄位完成後按下“存檔”按鈕，若順利存檔系統會跳出“存檔成功”的對話框。如圖 3.23



圖 3.23 存檔成功對話窗

若使用者按下“存檔”按鈕時，ID欄位是沒有輸入的任何值的狀態，此時因為系統的ID欄位是被設為索引鍵不允許為null，因此系統會跳出要求使用者輸入ID欄位的對話框，如圖3.24。另外，若使用者不想做任何新增的話，可按下“離開”，此“新增模型資料”的對話框就會關閉，然後回到模型資料表介面。



圖 3.24 請輸入 ID 欄位的值

- 4) 刪除一行：使用者欲刪除某一資料列時，必須先在左邊的資料表內選擇欲刪除的資料列，被選擇的資料列其背景會變成灰色，此時按下“刪除一行”的按鈕，系統就會跳出提示的對話框，讓使用者確定將要刪除的資料列。例如選擇刪除的是ID為HA000044的資料列，對話框就會顯示“是否刪除ID：HA000044這一行？”，如圖3.25所示。

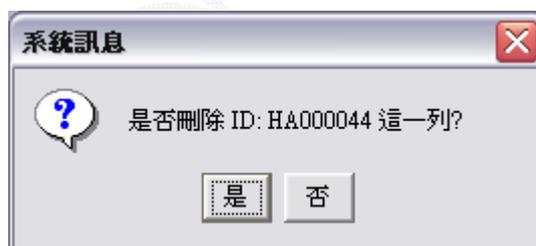


圖 3.25 是否刪除對話窗



圖 3.26 成功刪除對話窗

若使用者不刪除，可按下“否”的按鈕，這樣系統就會取消刪除的動作，回到模型資料表的介面。若確定刪除的話就按下“是”的按鈕，系統刪除成功就會再跳出圖 3.26 的對話框。

- 5) 修改一列：欲修改某一資料列，也是要先在左邊的資料表選擇資料列，被選擇的資料列其背景會變成灰色，此時按下“修改一列”的按鈕，系統就會跳出“更新模型資料”的對話框。“修改一列”的操作方式與“新增一列”大同小異，圖 3.27 是按下“修改一列”按鈕後跳出的視窗，大小為 750*650。與“新增模型資料”對話框比較不同的是，當對話窗一跳出來，所有欄位的資料會顯示在畫面上，使用者可直接在上面作編輯。若編輯錯誤，按下“回復”按鈕可回復已儲存的資料。



圖 3.27 更新模型資料

另外若使用者編輯後將 ID 欄位的值刪除，並按下“更新”按鈕，系統將會跳出如圖 3.24 的提示對話框。又若編輯後的 ID 值與原 ID 值不同，系統就會檢查此 ID 值在資料表中是否存在。假如存在，就不會作更新的動作，並要求使用者指定其他

的值，如圖 3.28。假如不存在，系統就會更新成功，如圖 3.29。

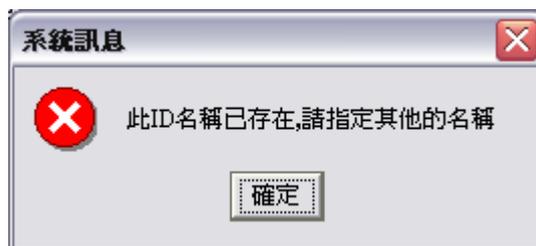


圖 3.28 ID 已存在對話窗



圖 3.29 資料更新成功對話窗

- 6) 儲存圖片：按下“儲存圖片”按鈕可選擇欲儲存欄位的圖片，然後按下“存檔”按鈕便可將選擇欄位的圖片資料存入硬碟中。如圖 3.30。



圖 3.30 儲存資料庫圖片

- 7) 回主畫面：按下“回主畫面”按鈕可結束模型資料表介面，回到主選單介面。

服裝資料表：



圖 3.31 服裝資料表(800*620)

服裝資料表的介面及操作方法跟模型資料表介面幾乎一樣，所以這裡不再多加贅述。只放上幾個代表性的介面圖片，以供比照。



圖 3.32 增加服裝資料

髮型資料表：

髮型資料表的大小為 800*600。髮型資料表的介面和操作方法也和服裝資料表介面及模型資料表介面很類似，但是服裝資料表介面及模型資料表介面存放圖片的欄位一共有 32 個(8 個方向*4 個分解動作)，而髮型資料表存放圖片的欄位只有 8 個。這是因為髮型圖片只有分八個方向，沒有再分分解動作的緣故。因此髮型資料表介面，較服裝資料表介面及模型資料表介面要小一些，不過這三者能作的功能是相同的。由於介面和操作方法相似的關係，我們在這裡也只放上代表性的圖，不再多加贅述。



圖 3.33 髮型資料表



圖 3.34 增加髮型資料 (750*650)



圖 3.35 更新髮型資料

群組資料表：

群組資料表跟之前提到的三個資料表差別較大，群組資料表的欄位只有 ID 和 name 兩個，因此規模要小的許多，不過資料表的功能選項還是跟之前提到的那三個資料表相同。

新增群組資料的對話框如圖 3.37，大小為 380*300，可供使用者輸入資料值。此新增功能的機制，例如：辨識 ID、尋找 ID 和 ID 欄位值不可為 null 等，都與之前提到的三個資料表相同。更新群組資料的對話框，如圖 3.38，與新增群組資料的對話框在使用方式上大致相同，更新功能的機制也與前三個資料表相同，之前已有詳細的解說，所以這裡就只放更新群組資料對話框的圖，僅供比照。

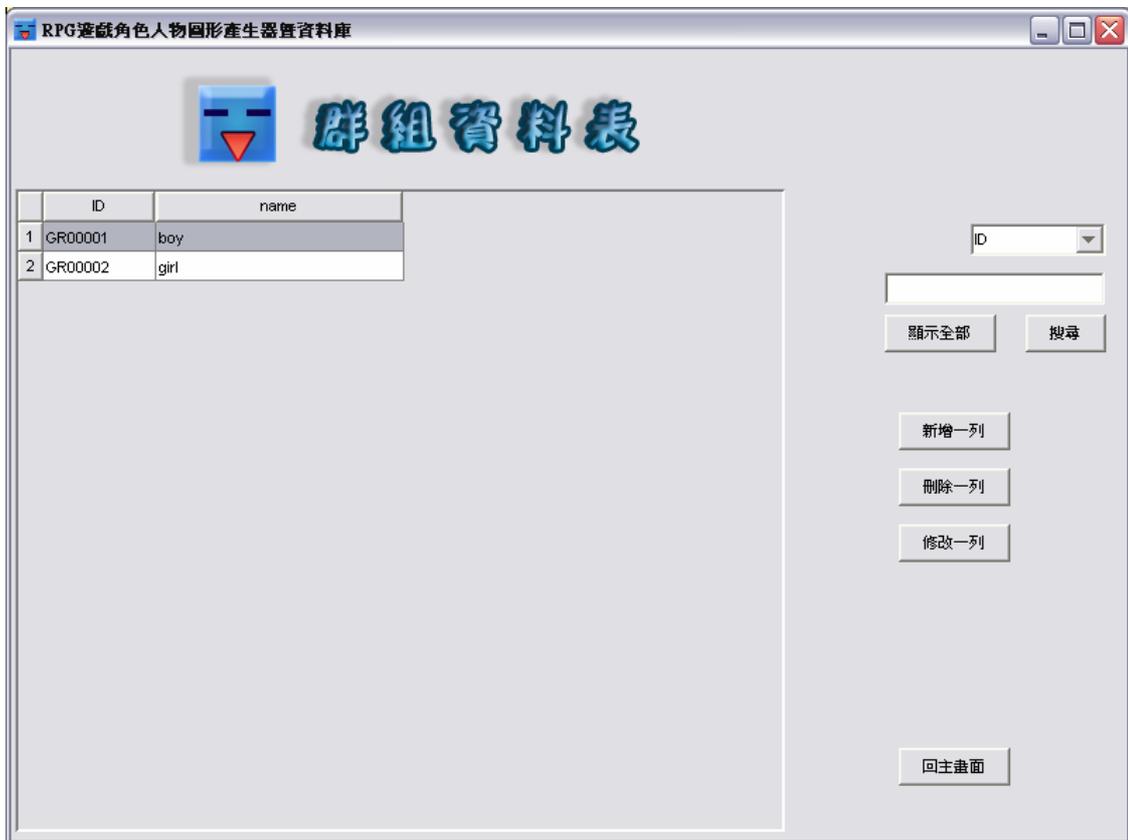


圖 3.36 群組資料表



圖 3.37 新增群組資料表



圖 3.38 更新群組資料

樂、嵐：

這是可以用來實現我們人物組合結果的實驗性小遊戲，執行的畫面如圖 3.39 和圖 3.40。



圖 3.39 樂、嵐主畫面選單



圖 3.40 樂、嵐遊戲畫面

甘特圖



介面設計	12/1~15	12/16~31	1/1~15	1/16~31	2/1~14	2/15~28	3/1~15	3/16~31	4/1~15	4/16~5/7
介面設計										
資料庫新增										
資料庫刪除										
資料庫更新										
資料庫查詢										
資料庫顯示資料表										
資料庫顯示圖形資料										
系統介面設計										
系統介面實作										
自動產生器設計										
自動產生器實作										
繪製圖形										
系統整合										
程式測試與修改										
報告資料收集										
報告撰寫										

第四章 系統分析與評估

4.1 使用評估

使用者需求：熟悉 WINDOWS 介面的操作方式。

系統需求：WINDOWS 95 以上的作業系統。

4.2 資料流程圖

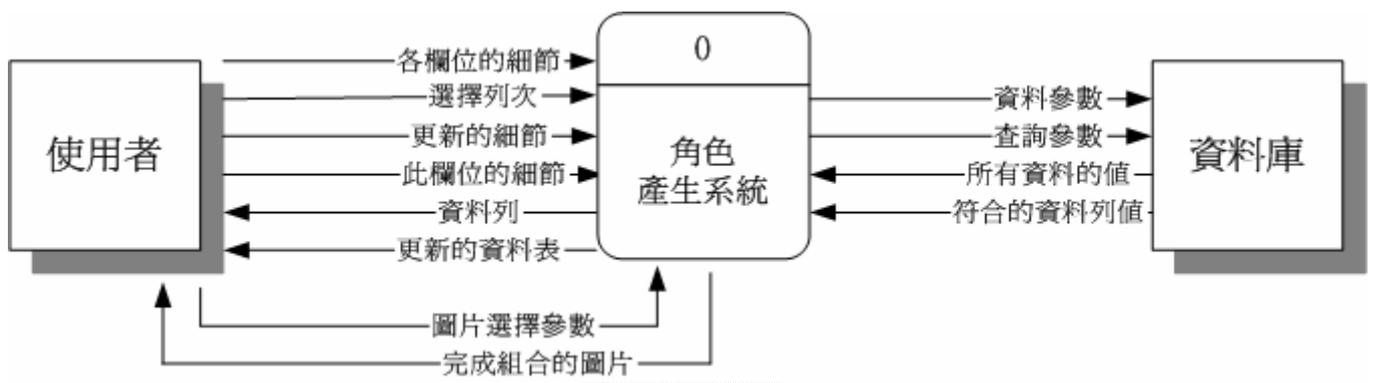


圖 4.1 Context Diagram

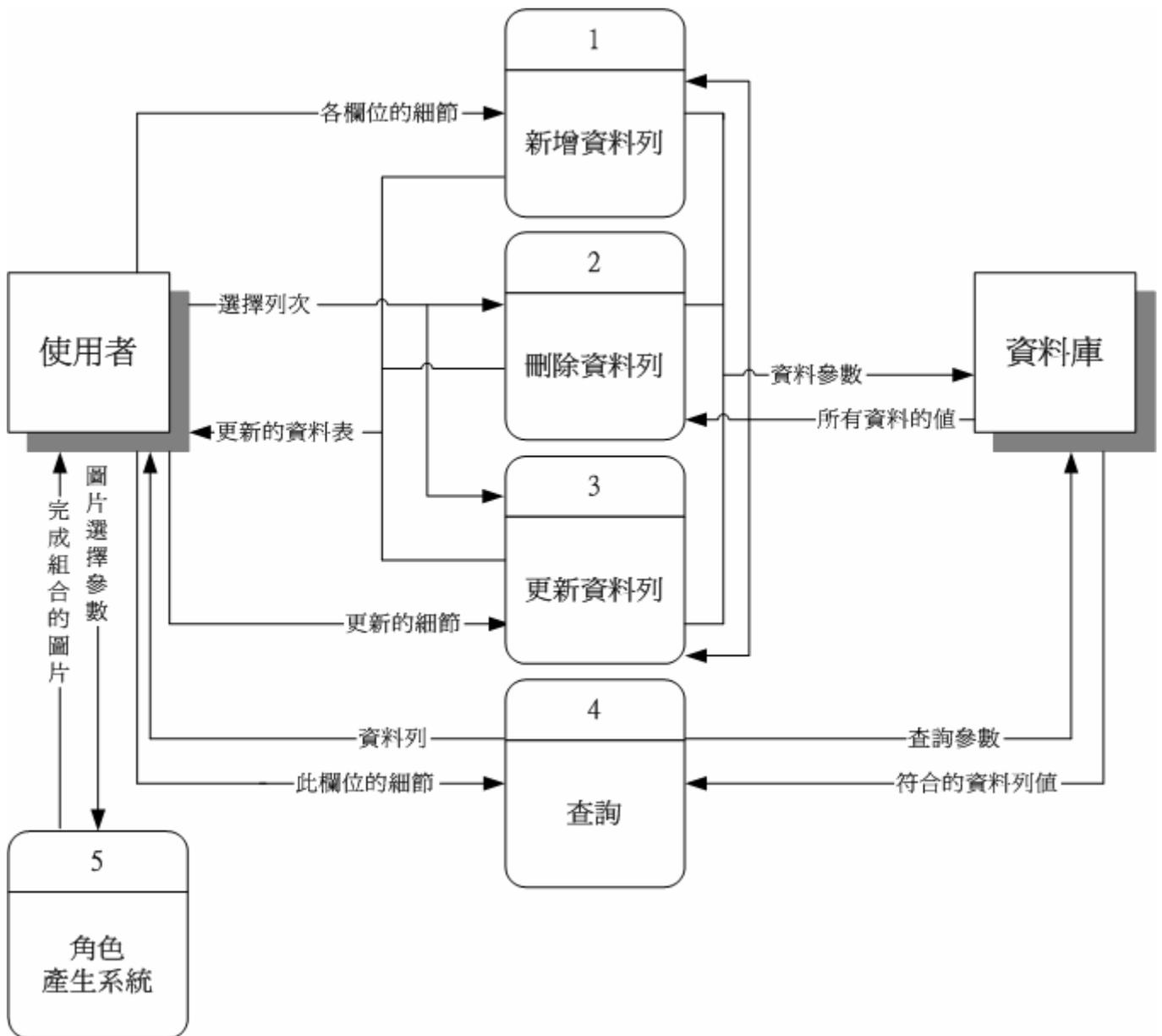


圖 4.2 Diagram 0

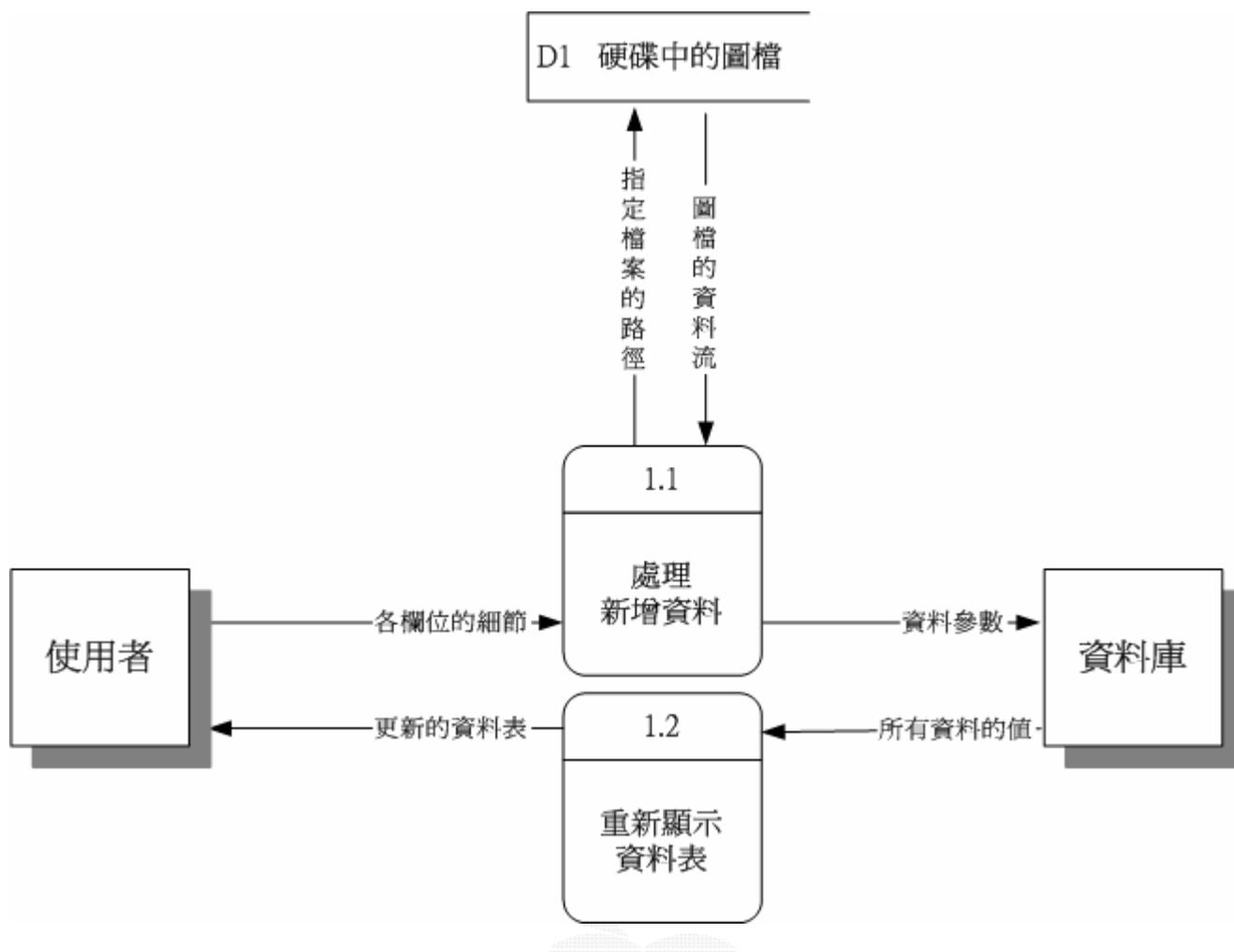


圖 4.3 Diagram 1

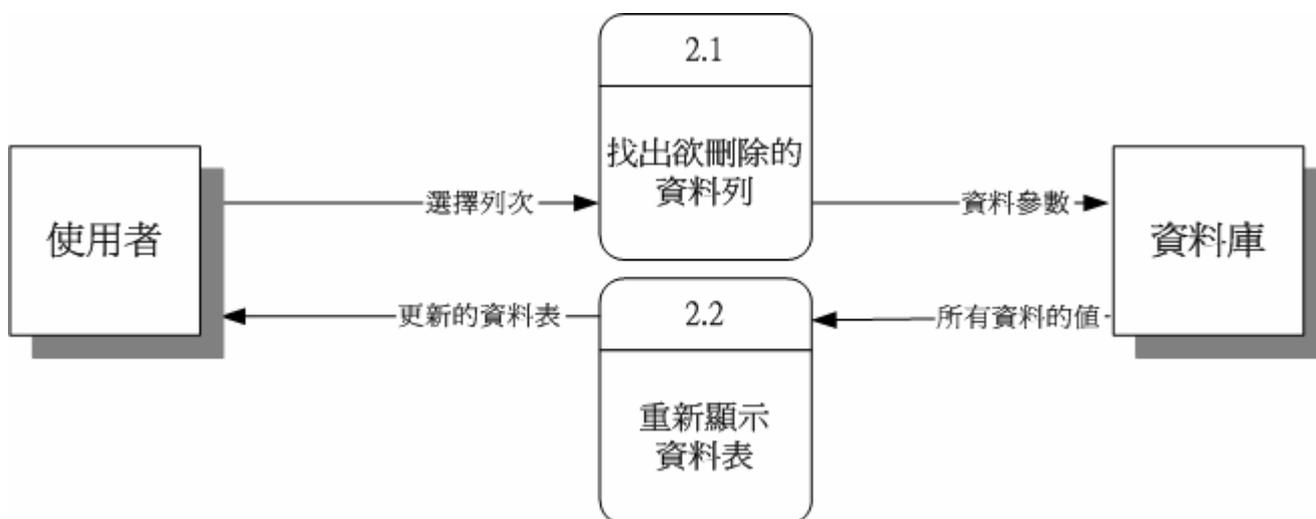


圖 4.4 Diagram 2

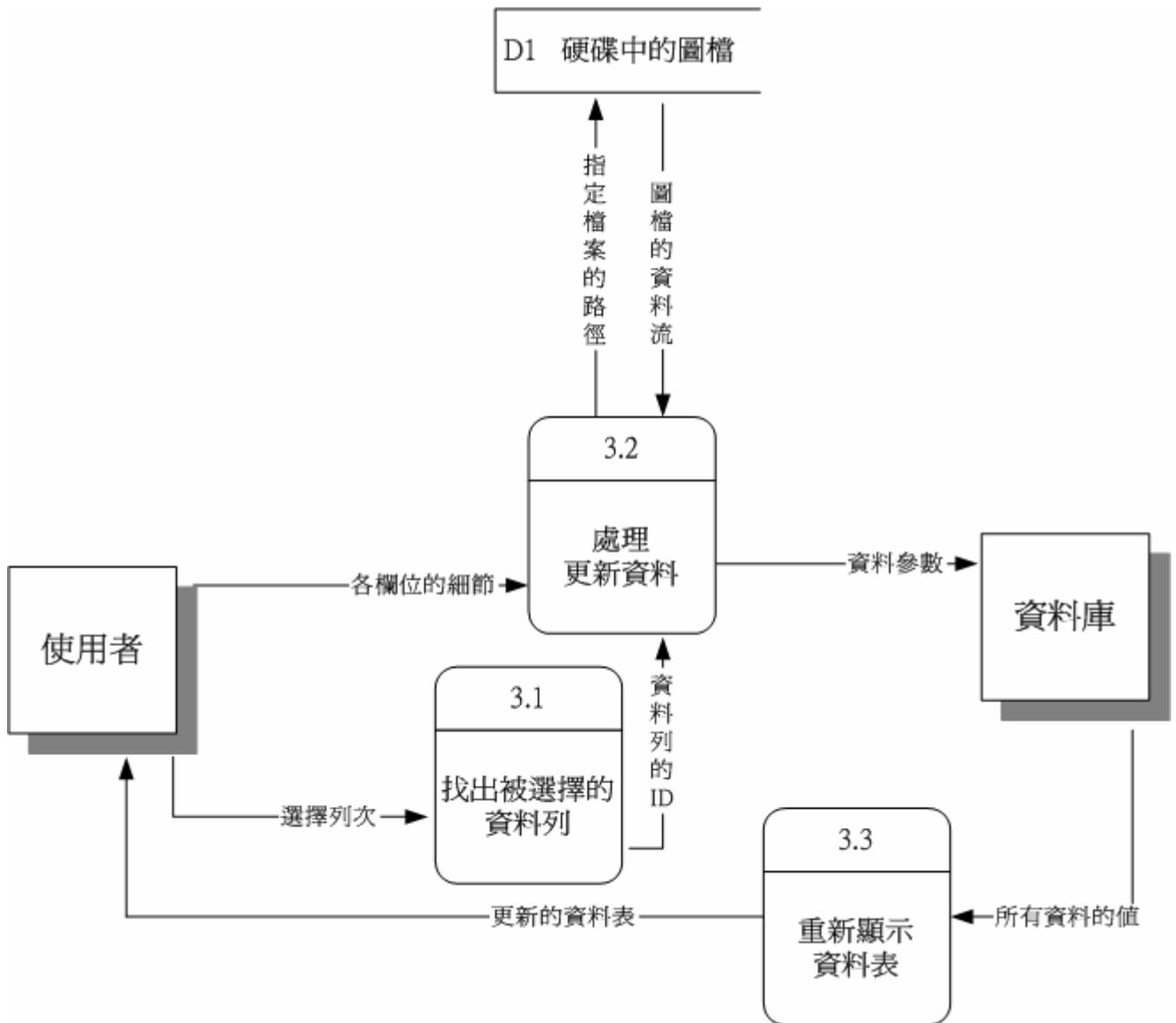


圖 4.5 Diagram 3

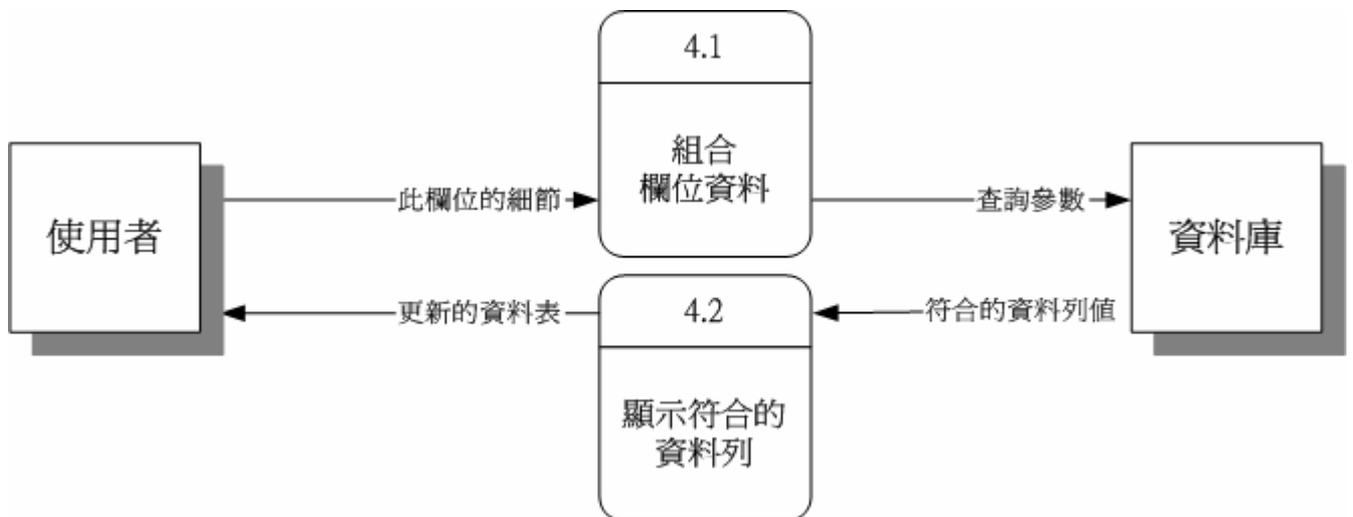


圖 4.6 Diagram 4

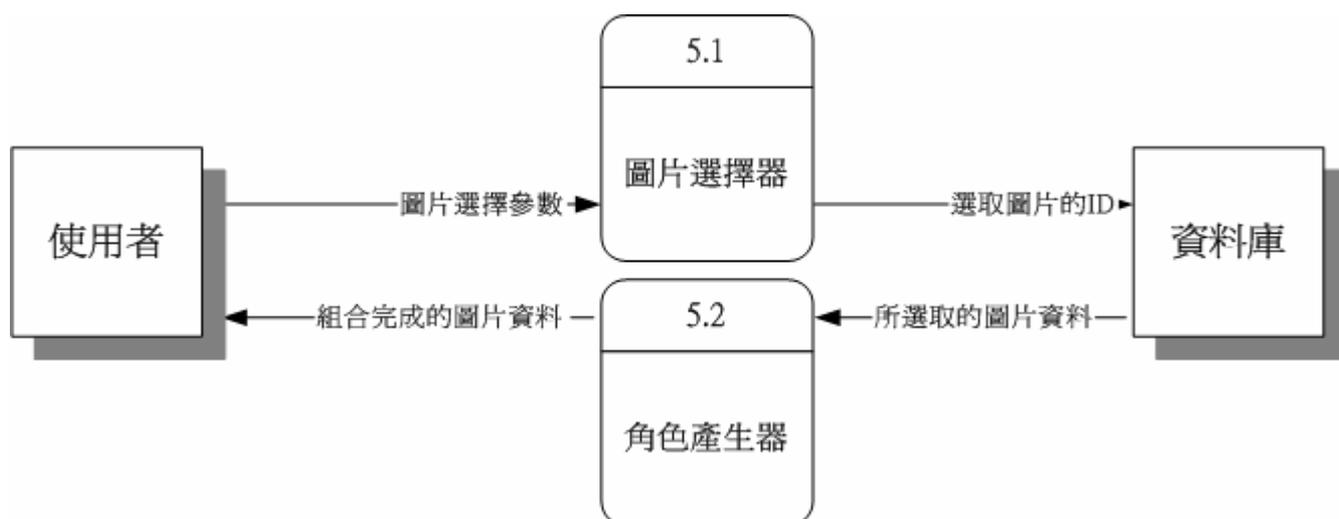


圖 4.7 Diagram 5

4.3 未來發展

此系統是針對人物圖形作分類管理組合，但是如此自動化的想法，在遊戲的製作中，可以應用在許多的地方，例如在：

- (1) 地圖的編輯，把地圖切成一個個元件，像是樹、房子、岩石等，當然還可以細分，像是不同數量的樹，或是把房子切割成更細的元件，像是窗戶、門等，有很多種切割方式，然後再以一定的組合規則，組合出各式各樣的地圖，這樣在構思地圖的時候，也可以比較有系統的編輯，不會把地圖畫的雜亂無章。
- (2) 道具裝備的圖形管理組合，將道具分成幾個種類，然後分成幾個元件來組合，例如把劍分成劍柄與劍鋒，加以組合，就也可以將兩個完整的道具，直接拿來組合，成為新的道具，對於創造新道具，就可以有更不一樣的創意。
- (3) 甚至在任務劇情上，都可以以這樣的原理來組合，把劇情分為幾個片段，依照劇情的相容性分類，然後再加以組合，這樣一來，可以使原本數量不多的任務劇情的重複性減少，以少量的劇情製作出多變的故事。

第五章 結論與心得

5.1 結論與心得

雖然晚了一個學期發表，很高興最後終於還是把專題做出來了，回想這漫長的一年每逢假日或有空就是坐在電腦前寫程式，不斷地做專題幾乎完全沒有休息，而今天終於能全部完成真的覺得很開心。我負責的部份是資料庫相關和系統整合，剛開始最困難的就是將圖片存到資料庫，和用資料表顯示資料庫中的圖片這兩件事。因為市面上都沒有提及相關方法的書，到圖書館找書也找不到有類似的範例，所以一開始十分沮喪，甚至覺得做出來是不可能的事。後來請教過老師，並多方嘗試，對 JAVA 這個語言比較熟悉後，在網路的論壇上終於找到了類似的實作方法，於是才完成了這初步的工作。之後的工作雖然也不容易，但也許是因為我變的比較了解 JAVA 語言的關係，或較知道要從哪裡找資料看的緣故，做起來的挫折感就沒有一開始那麼嚴重，寫程式的速度也比以前快的多了。做完這個專題真的學到了很多東西，感覺比以前混混的大三要懂了許多，雖然中間的過程很辛苦，但最後完成時真的很有成就感，也覺得收穫良多。

資訊四丁 林佳慧

這次所製作的產生器與資料庫，我所負責的部份是介面和產生器的設計與製作，所用的語言是不太熟悉的 JAVA，所以一開始熟悉 JAVA 的方面，花了不少時間。而所負責的介面部分，一開始因為對於元件功能與使用方法不熟練，所以遇到許多錯誤，例如視窗的連結，顯示圖片元件的種類選擇等，所以整個視窗介面更改了很多次；而接下來製作的圖形產生器，主要在圖片的疊合，不過在製作時遇到一個大問題，就是圖片在儲存的時候，JAVA 只有提供 JPEG 的 encode，所以在儲存圖形的時候，並沒有辦法使背景透明，buffer image 有提供每個圖素(pixel)是 4 BYTES，其中三個 BYTES 是 RGB 三原色，而最高 byte 就是 transparency，set 為 0 就會透明了；java buffer image 有限制，也要想辦法突破，以 buffer image 為基礎，雖然格式有 4bytes，但因為存檔時用的是 jpeg encode，所以沒有辦法表現出透明，能做出透明的規格，只有 PNG 和 GIF 兩種規格，所以，上網搜尋到 JAVA 的相關網站[Java Glossary:Glossaries]，有 GIF Encoder 的相關說明，找到了一個 gif.jar，gif.jar 作了 gif 格式的 decode 和 encode，所以可以順利的解決圖片透明的問題，至此，專題也告一段落了，總算是完成專題。

資訊四丁 藍卞鴻

5.2 參考資料

- [1] 林志揚、范明翔、陳錦輝著，Java Swing 程式設計，金禾資訊股份有限公司 2002 年九月初版
- [2] Steven Holzner 著，張裕益、劉春成譯，徹底研究 Java2 Java2 Black Book，博碩文化出版社 2002 年十二月初版
- [3] Matthew Robinson、Pavel Vorobiev 著，鄒修銘、吳俊儀、胡凱智譯，Swing 實作手冊<第二版>，
- [4] 蕭仁惠、陳錦輝著，JDBC 資料庫程式設計，金禾出版社。2003 年九月
- [5] [Java Glossary:Glossaries]
<http://mindprod.com/jgloss/glossaries.html>

