

國立交通大學
工業工程與管理學系

碩 士 論 文

利用 UML 和 Java 建構生產與運籌教學輔助遊戲
Using UML and Java Constructing a Production and Logistics Game

研 究 生：王柏詠

指 導 教 授：張永佳 教授



中 華 民 國 九 十 六 年 十 二 月

利用 UML 和 Java 建構生產與運籌教學輔助遊戲
Using UML and Java Constructing a Production and Logistics Game

研究生：王柏詠

Student : Po-Yong Wang

指導教授：張永佳

Advisor : Dr. Yung-Chia Chang

國立交通大學
工業工程與管理學系
碩士論文

A Thesis

Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Industrial Engineering and Management

December 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年十二月

利用 UML 和 Java 建構生產與運籌教學輔助遊戲

學生：王柏詠

指導教授：張永佳 教授

國立交通大學工業工程與管理學系碩士班

摘 要

隨著企業競爭環境的快速變遷與全球化的影響，供應鏈管理儼然是一個非常重要的課題。在實務上，企業決策者為了因應此種複雜化的競爭環境，除了必需考慮到企業本身內部營運的因素之外，還要考慮到整個供應鏈系統的問題；在教學上，教學者除了以傳統方法教授課程之外，也面臨到如何以實務上的問題吸引同學對課程的興趣。而為了提升教學效果，近年來有許多輔助教學工具被發展出，其中一種方式則是利用遊戲。遊戲不僅可讓教學更生動活潑，利用競賽的方式亦可增加學生學習的興趣。然而，現有之供應鏈遊戲往往只限於使用在一個課程中，較缺乏彈性，而導致教學者及學生必須花費時間適應不同的遊戲介面。本研究希望能結合教育與實務，發展出一套較具彈性的網際化(web-based)供應鏈遊戲教學平台，讓教學者能在不同的情境下經由相同介面透過參數的設定執行適當的遊戲，使學生能藉由遊戲的方式增加對供應鏈相關課程的學習興趣，以提高學習效率。

本研究以反覆與漸進式的軟體開發模式來發展此遊戲軟體，利用統一塑模語言(Unified Modeling Language, UML)與Java作為物件導向分析與撰寫的工具，將軟體設計的各個功能模組化，建立整合容易與方便擴充的軟體架構，並使用Servlet技術與網頁技術處理後端伺服器。未來可配合教學做適當性的修改，讓本研究在工業工程相關課程的教學上會更有效率。

關鍵字：Java、UML、生產與運籌、教學輔助工具。

Using UML and Java Constructing a Production and Logistics Game

Student: Poyong-Wang

Advisor: Dr. Yung Chia-Chang

Abstract

Due to the rapid change in business competitive environment and the trend of globalization, supply chain management has become a critical issue in recent years. In practice, business decision makers have to consider not only the operations within their organizations but also the entire supply chain system in order to keep up with the highly complicated competition.

On the other hand, when teaching Production Management, Logistics Management, Supply Chain Management and many other courses in decision sciences, instructors are often confronted with the challenge to continuously develop and adopt new teaching aids to facilitate learning, to increase learning interests, and to get students prepared for their professional career and/or for further research. Among those teaching aids, gaming has been proven to be an effective tool. Through playing games, students can not only be able to internalize their knowledge learned from class but also get familiar with the course concepts. However, current available teaching games are restricted to some specific concept. Instructors and students have to take time to get familiar with different game interfaces for different concepts taught. Therefore, the research aims to develop a flexible web-based game so that instructors can adjust game settings to simulate different scenarios via the same interface to facilitate learning efficiency.

The research uses an iterative and incremental software development process to develop this game. Unified Modeling Language is first applied to analyze the system requirement and Java and Java Servlet are applied used to implement the systems.

keyword: Java, UML, production and logistics, teaching aid.

誌謝

論文的完成，首先最感謝的就是張永佳老師給我的指導，真的讓您費心了，這段期間我的最大的收穫不僅是有關文獻資料的整合、程式技巧的琢磨，最重要的在完成這些事情的态度與堅持！這也讓我知道态度以及應變能力的重要性。

也非常謝謝我的朋友在我身邊鼓勵我，avi、papa、KC、宗佐學長、阿立、慧純、元易哲豪等很多好友、以及最重要的父母和小芝在旁邊的陪我，沒有妳們的鼓勵，這個難關我想也是很難熬下去的！我會帶著這份心情往下一個階段邁進！

柏詠 于竹北



目錄

	頁次
摘要	iii
Abstract	ii
誌謝	iii
目錄	iv
表目錄	v
第一章 緒論	1
1-1 研究動機與背景	1
1-2 研究目的	2
1-3 研究架構	2
第二章 相關文獻探討	4
2-1 國內外有關供應鏈遊戲平台之介紹	4
2-2 軟體系統開發的介紹	15
2-2-1 系統設計的三層式架構概念	15
2-2-2 軟體開發的模式	18
2-3 統一塑模語言的介紹	20
2-3-1 統一塑模語言的特點	20
2-3-2 統一塑模語言的圖形種類	20
2-4 相關教學文獻探討	32
第三章 系統架構與研究方法	35
3-1 系統架構	35
3-2 需求擷取	35
3-3 分析	36
3-3-1 原物料供應商	36
3-3-2 製造商	37
3-3-3 物流中心	37
3-4-1 建立使用案例圖	39
3-4-2 概念塑模	43
3-4-3 建立類別圖	49
3-4-4 建立互動圖	49
第四章 系統實作與介面介紹	52
4-1 系統開發環境	52
4-2 系統實作	52
4-3 系統介面介紹	53
4-4 工具列項目介紹	61
第五章 結論	71
5-1 研究成果與貢獻	71
5-2 未來研究方向	71
參考文獻	73

表目錄

表 1 各遊戲整理表	12
表 2 各遊戲整理表	13
表 3 統一塑模語言所用的圖形	21
表 4 類別圖屬性的可見度	23
表 5 文獻整理表	34
表 6 使用個案描述：選擇角色	40
表 7 使用個案描述：訂購產品	41
表 8 使用個案描述：選擇模組	41
表 9 使用個案描述：顯示圖表	41
表 10 概念塑模：選擇角色	43
表 11 概念塑模：訂購產品（執行遊戲）	45
表 12 概念塑模：顯示圖表	48
表 13 統一塑模語言統計圖表	71



第一章 緒論

1-1 研究動機與背景

隨著企業競爭環境的快速變遷與全球化的影響，供應鏈管理儼然是一個非常重要的課題。市場上的競爭不單侷限於個別企業間的競爭，而是發展出以一條供應鏈對另一條供應鏈的競爭。

也由於企業上的競爭策略越來越複雜，在實務上，決策者所需要考量因素也隨之增加，除了企業本身內部的營運之外，必須考慮到整個供應鏈系統的問題，如上下游廠商的合作關係、顧客的滿意度等。同時，在與工業工程相關的教學上，教授也面臨到如何以實務上的問題吸引同學對課程的興趣，而解決大部分實務問題所需的知識通常是多個課程的整合，例如：要了解企業決策對整條供應鏈的影響必須具備有關需求預測、生產排程、存貨管理、工廠設置、財務等各個相關課題的概念。

為了增加教學的效率，開始有越來越多的輔助教學工具被發展出來，使學生在操作過程中，可以對整個企業的生產、物流的流程架構有更充分的了解，而其中一種輔助教學工具就是遊戲。透過遊戲的方式，可以讓原本整個複雜的體系變得更簡單而有趣，而更進一步以競賽的方式進行，亦可提高同學參與的興趣。在過去十幾年來，多種型態的遊戲已經成功地應用在與生產管理相關課程的教學上，而事實上也證明遊戲會讓學生對於管理上的興趣大增[6]，在教學上的功效亦可彌補傳統教學方法的不足[5]。

第一個眾所皆知有關生產以及物流的遊戲就是啤酒遊戲(the beer game)，它是在1960年代由美國麻省理工學院(MIT)所發展出來的角色扮演遊戲，其目的主要是說明在生產、物流、以及顧客所創造出來的模式(pattern)如何造成非預期的結果。啤酒遊戲一開始是用紙本的方式進行，後來才被移植到電腦上。其所引發之長鞭效應相關探討，是在學習供應鏈管理時必接觸的議題。在啤酒遊戲之後，由於網際網路的盛行，結合幾個網際化(web-based)平台的同性質遊戲陸續發展出來，例如：Trading Agent Competition (TAC) game[15]、Global SCM game[16]等。這些遊戲不僅在課程教學上有所幫助，也提供學生，教授以及研究學者經由網路化的遊戲平台而有所互動。

到了1990年初期，IFIP(International Federation for Information Processing)[22] Working Group 5.7 整理與分析當時應用在教學上且與生產與物流相關的遊戲後發現，沒有一個遊戲可以適用於所有不同的生產以及物流的情境，但在一個課程中通常必須使用多個遊戲，因此在遊戲的選擇上必須考慮到使用遊戲的成本與其所帶來的效益。本研

究能結合教育與實務，發展出一套較具彈性的生產及物流模擬遊戲，提供教學使用，讓使用者能在不同的情境下經由相同介面透過參數的設定執行適當的遊戲，以提高學習效率。

在軟體的開發上，由於統一塑模語言(Unified Modeling Language, UML)擁有視覺化易懂的特性，可以讓程式開發人員一目了然，所以本研究選擇UML作為物件導向分析的工具。另外，由於Java語言有”Write once ,run anywhere”的平台可攜性，除了可以讓本系統的維護更簡潔之外，未來可配合系統硬體或環境升級，因此以在開發系統上，本研究選擇Java作為撰寫的工具，並結合網頁技術和使用Servlet技術處理後端的伺服器的運作來設計此遊戲教學平台。

1-2 研究目的

基於上述，本研究的主要目的如以下兩點所示：

- 1.設計以遊戲為主的平台，以應用於供應鏈管理相關課程的教學上，使學生經由此遊戲方式對於越來越複雜的供應鏈有所了解，並提高其學習興趣。
- 2.利用UML和Java來分析和撰寫此系統，並結合網頁技術和使用Servlet技術發展前端介面與後端的伺服器的運作來設計此教學平台。



1-3 研究架構

本論文的內容共分五章：第一章為緒論，包括論文之研究背景與動機、研究目的、研究架構；第二章則是相關文獻的回顧；第三章則說明系統的方法架構，其中包含開發系統的工具與環境，系統的功能，系統發展的時程與需求分析以及系統分析與系統設計；第四章介紹系統發展與實作；第五章則是結論。研究架構如圖 1 所示。

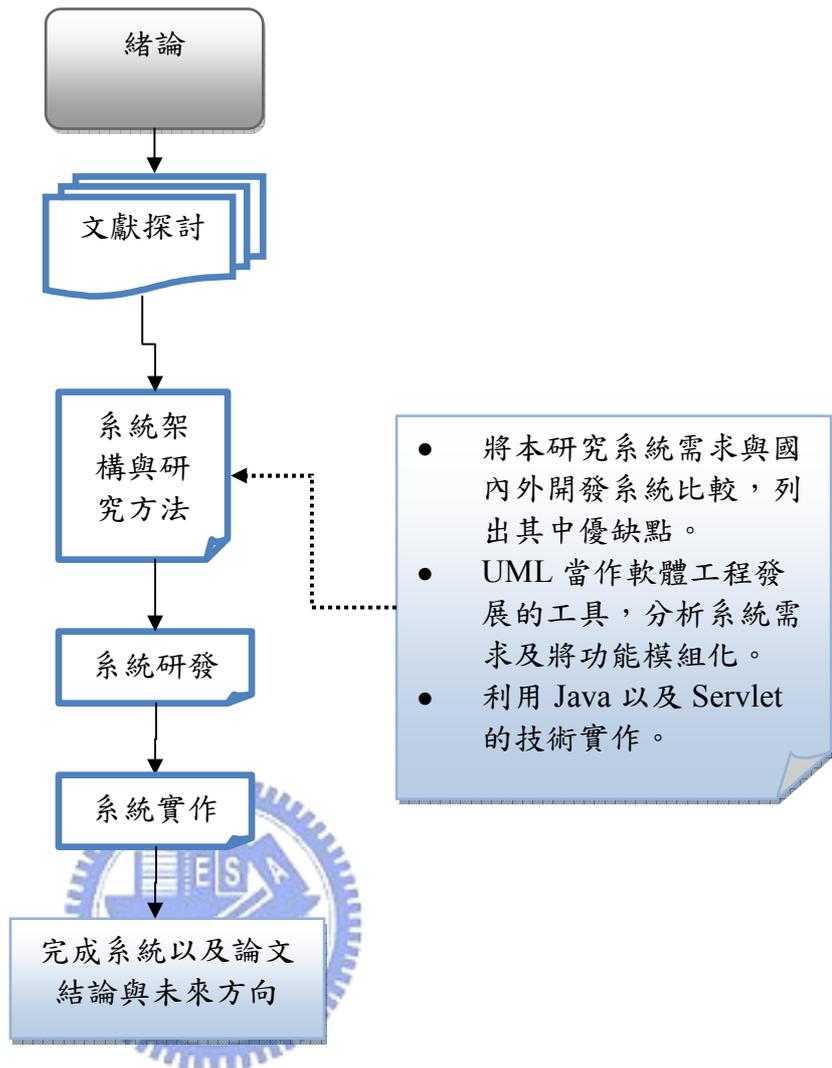


圖 1 研究架構

第二章 相關文獻探討

本章有四小節，第1節為國內外有關供應鏈遊戲的探討，主要介紹幾個有名的遊戲，再針對這些遊戲作比較，分析其中的優缺點。第2節則針對一般軟體開發作簡單的介紹，並討論為何使用三層式架構以及一些軟體開發的模式；在軟體研發上，統一塑模語言可作為不同程式設計背景的程式設計師之溝通橋樑，因此，第3節簡介統一塑模語言。最後，第4節對有關遊戲應用於教學績效上之文獻作個探討。

2-1 國內外有關供應鏈遊戲平台之介紹

以下介紹幾個有關生產及物流的遊戲，並為日後在開發此教學平台時可有個參考：

一、電腦化的啤酒遊戲[15]：



圖 2：MIT 啤酒遊戲

資料來源：<http://beergame.mit.edu/>

啤酒遊戲是由1960年代由MIT發展出來，當時被學者用來廣泛討論系統動態的長鞭效應。整個供應鏈的組成有工廠,配銷商，批發商，零售商所組成，如圖3所示（以單機的畫面來說明）：

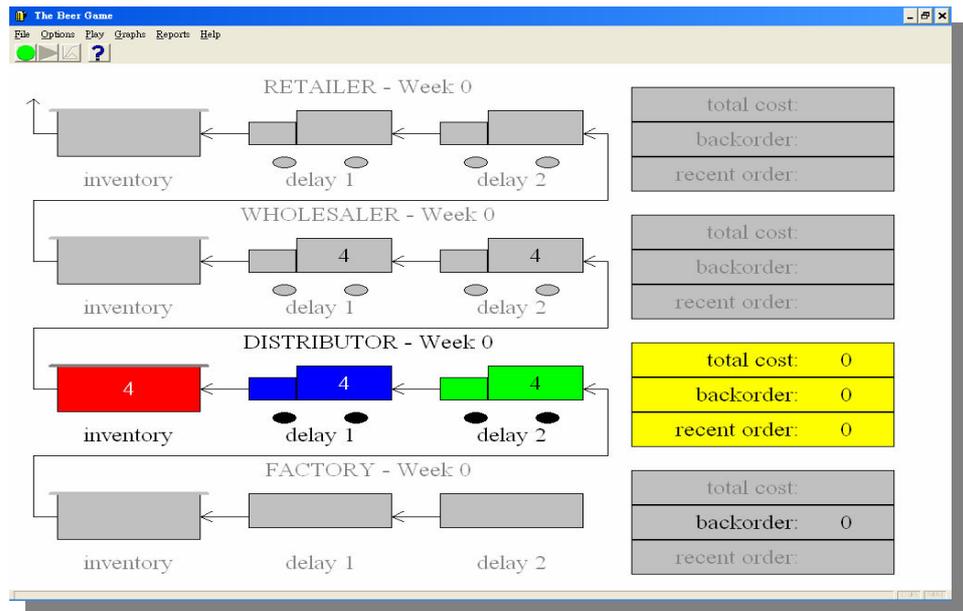


圖 3 電腦化啤酒遊戲單機畫面[10]

電腦化的啤酒遊戲每一階的組成(component)參數設定幾乎相同，遊戲一開始，玩家可選擇扮演的角色及存貨策略；資訊流的前置時間為一個禮拜，物流的前置時間為兩個禮拜，也可以縮短為一個禮拜；缺貨成本為1元，存貨成本為0.5元，沒有訂購成本，遊戲可進行為50週；每一階產能都是無限的，沒有外包、運輸等策略；顧客需求可為隨機(Random)或已知(deterministic)；另有集中化管理以及資訊透明化的選項。

二、Hulia game[16]：

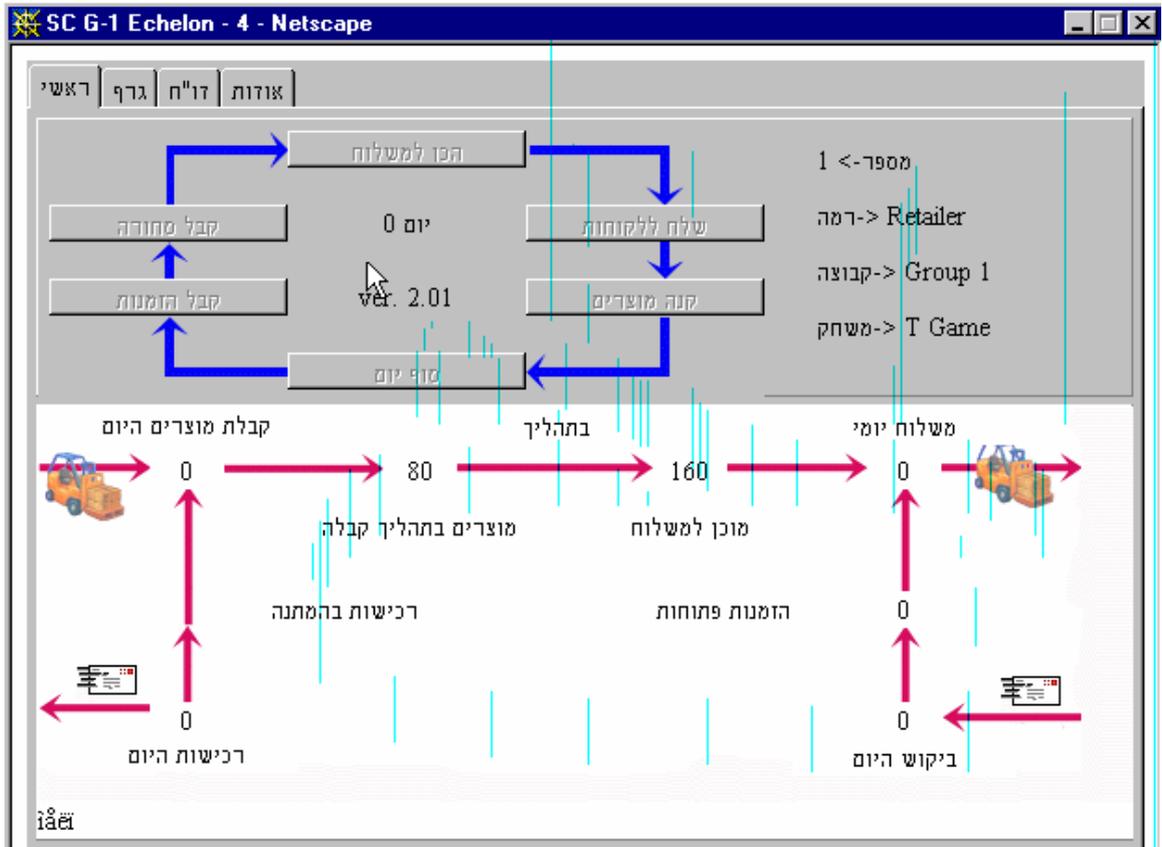


圖 4 Hulia game

資料來源：<http://hulia.haifa.ac.il/Eng/hulia.html>

Hulia game[16]由Haifa大學所發展，主要是模擬動態之決策系統，目的是讓玩家在面對不同的決策時有其依據。如圖2-3，此遊戲是以啤酒遊戲為基礎之動態系統管理的模擬遊戲，也是一個教學的工具，可幫助玩家作動態之決策，目前有英語和希伯來語的版本，目的是讓玩家對整個動態系統有所了解，增進公司組織的績效；在教學上它所包含的議題有供應鏈管理、資訊系統、企業流程再造、電子商務與動態系統決策等。

三、Trading Agent Competition(TAC) game[17]：



圖 5 TAC game

資料來源：<http://www.sics.se/tac>

此遊戲由Swedish Institute of Computer Science所發展，是一條三階的供應鏈，由工廠，經紀人與客戶所組成，主要是決定客戶的最佳服務；此遊戲有兩種情境，分別說明如下：

(1)TAC classic：此遊戲有八個客戶以及八個經紀人，每一個經紀人都是旅外的經紀人，負責將旅行包裹在五天內從TAC市(TAC Town)運送到坦帕市(Tampa)，旅行包裹裡面包含每一個顧客的來回班機、旅館訂位以及到娛樂公園、博物館的票，而每一位經紀人的目標就是讓顧客滿意度最大化。如圖6所示：

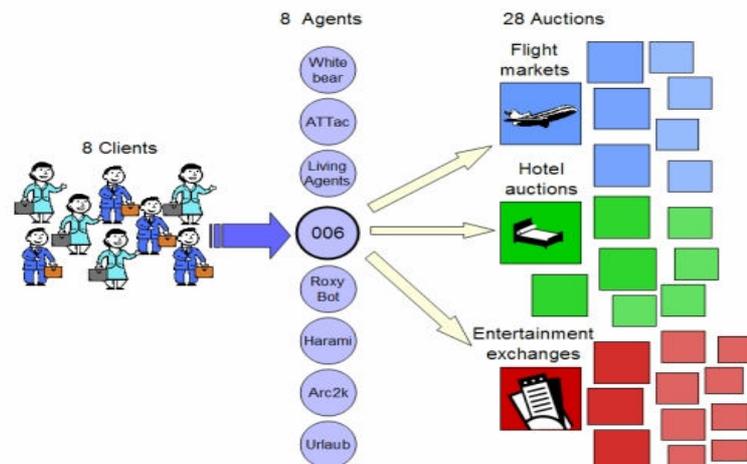


圖 6 TAC classic

資料來源：<http://www.sics.se/tac/>

(2)TAC SCM：此情境有六台個人電腦經紀人去取得客戶訂單以及生產各式各樣的元件，而組成一台個人電腦的零組件有中央處理器、主機板、記憶體以及硬碟；每一天不同的顧客需求會根據零組件的運送日期跟價格去挑選不同的經紀人，每一個經紀人必須從有限的產能跟產線滿足顧客的需求，如圖7所示。

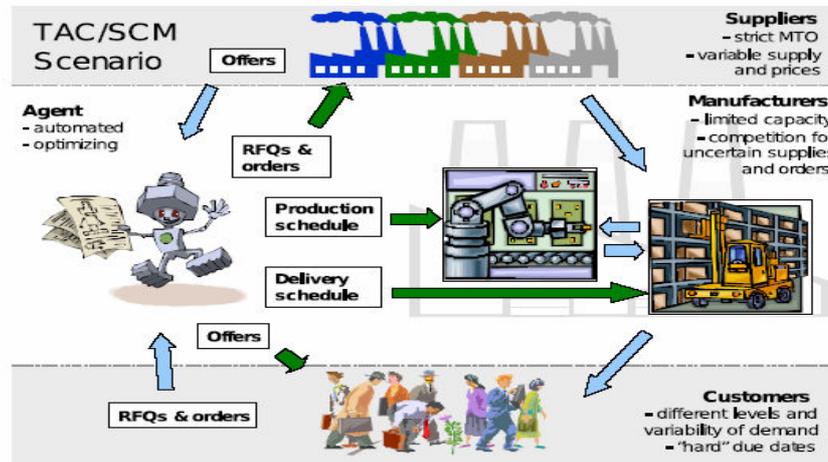


圖 7 TAC SCM

資料來源：<http://www.sics.se/tac/>



四、Littlefield Technology[18]：



● Day: 77 ● Team ID: operators ● Cash Balance: 1,031,961

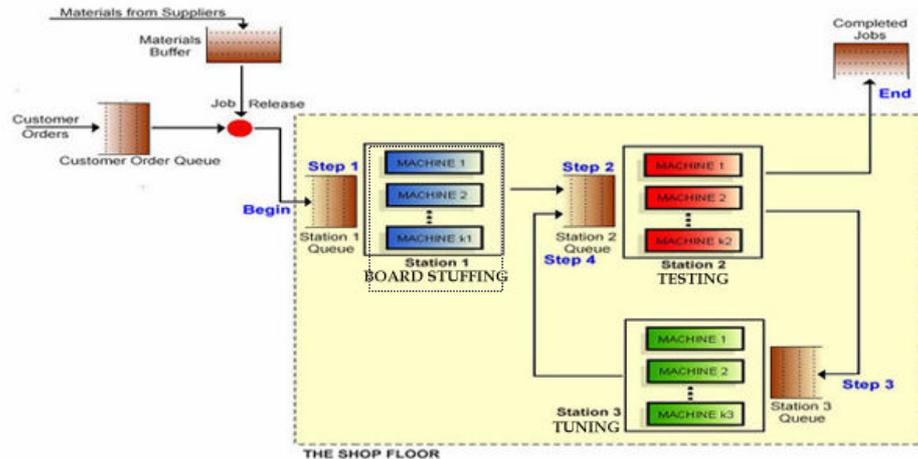


圖 8 Littlefield Technology

資料來源：<http://littlefield.responsive.net>

此遊戲是由史丹佛大學所發展，主要是模擬工廠在電子零件接單式組裝(job shop)的環境下的遊戲，為線上的工廠管理模擬器，而這個工廠是接單式的衛星組裝工廠，組裝過程中有四個階段以及三個組裝站(分別是board stuffing, testing, tuning)。每個組裝站都有自動化機台，在指派的過程中可以添增機台；顧客需求為隨機，不同的產品有不同的機台順序，每個玩家在遊戲開始都有固定金額，可決定是否要買機台或原料來組裝衛星，而賺最多錢的玩家即是贏家。

五、Global SCM Game[19]：



[LARGE MESSAGES VIA MENU]

In contrast with small news messages, large news messages are used to present a larger amount of information to players. These messages typically inform a player about a certain event, for example a growth in economy. Furthermore these messages are used to present market information on a regular basis (e.g. on a quarterly basis). In order to retrieve the screen with the large messages, select the 'News messages' menu item from the top menu.



圖 9 Global SCM Game

資料來源：<http://www.gscg.org>

此遊戲由TU Delft所發展，主要強調配銷者的責任與互動，其目的是提供一個供應鏈教學遊戲。學生跟管理者可以經由此平台了解供應鏈之基本觀念。此遊戲有兩種情境，分別為需求激增(Demand Surge)和區域化與全球化(local vs global)兩種情境；另外有遊戲的流程說明，以供玩家模擬之功能及主機安裝資訊。

六、SCM game[20]：



The Supply Chain Game

Edit Scenario: default

Starting team settings:

Time and Administration

No of simulated days per real day:

Random number generator seed: (0 means system clock)

Total number of simulated days:

Student access begins on day

Student access ends on day

After completing screened period, simulator should: run suspend

Financial

Initial cash per team: (no commas)

Interest rate per year:

Are students ever allowed to increase debt? yes no

If allowed, loan becomes available on day

If loans allowed, initial cost of loan as a fraction of loan

Debt rate per year:
(Interest is compounded daily)

圖 10 SCM game

資料來源：<http://sc.responsive.net/SupplyChain/SCAdmin>

此遊戲是由RLT(Responsive Learning Technologies)的Sam Wood所發展，整個供應鏈只有兩階(工廠及倉庫)，可當作供應鏈的網路模擬器，玩家可設定生產和存貨控制的參數以及傳輸選項，決定何時要擴廠及倉儲。

遊戲情境說明：

主要有兩種情境，一種為單一區域指派(One Region assignment)，另一種則是網路指派(Network assignment)情境。單一區域指派問題是指在同樣區域中，只有單一的工廠及倉儲，在未來有關需求指派可從原本的單一區域擴展到不同區域；而網路指派情境中，玩家可在不同的區域中決定是否要增加工廠或倉儲，以及是否要增加公司的產能。

針對以上這些遊戲，本研究歸納了一些重點如表1。

表 1 各遊戲整理表

遊戲名稱	平台描述/ 階數	決策範圍	發展者
Beer game[15]	啤酒的生 產跟配銷 在多階配 銷通道/4	扮演製造 商、經銷 商、分配商 決定啤酒生 產量或訂購 量	MIT
Hulia game[16]	擬動態決 策系統/4	扮演製造 商、經銷 商、分配商 決定啤酒生 產量或訂購 量	Haifa College
Trading Agent Competition[17]	貿易與出 價供應練 遊戲/3	決定客戶的 最佳服務	Michael Wellman
Littlefield Technology[1 8]	模擬接單 後製造的 電子系統 /4	扮演製造商 決定利用 率、生產排 程以及存貨	Sunil Kumar and Samuel C. Wood

Global SCM Game[19]	強調配銷者的責任的互動/3	扮演配銷商決定產程、路徑以及排程等議題	Stijn-Pieter van Houten
SCM game[20]	供應鏈的網路模擬器/2	區域間的指派問題	Sam Wood

接著，以本研究 and 上述列舉之遊戲作比較，每個遊戲在應用上都有其適合和不適合的地方，整理如表2所示。

表 2 各遊戲整理表

遊戲名稱	適合	不適合
Beer game	讓玩家可以了解長邊效應	變數太少,過於簡單
Hulia game	與本研究方向接近	Demo影片很不清楚
Trading Agent Competition	Layout 清楚 加入價格變數	只有三階 擴充性有限
Littlefield Technology	圖表詳細,閱讀方便,有 demo 可以試玩	介面不夠親和 需付費

Global SCM Game	介面清晰好看	需付費
SCM game	參數自由度高 雖然只有兩階(工場跟倉庫) 但是擴充性高	需付費 參數介面的設定繁雜



2-2 軟體系統開發的介紹

2-2-1 系統設計的三層式架構概念

由於本研究所設計的教學平台是最終希望是網際化已達到教學目的，因此首先介紹客戶端與主機在網路上訊息的溝通方式。一般主機上的傳統的Web應用程式不外乎就是滿足下列兩項要求：接受客戶端的需求以及將結果傳給客戶端。如圖11所表示：

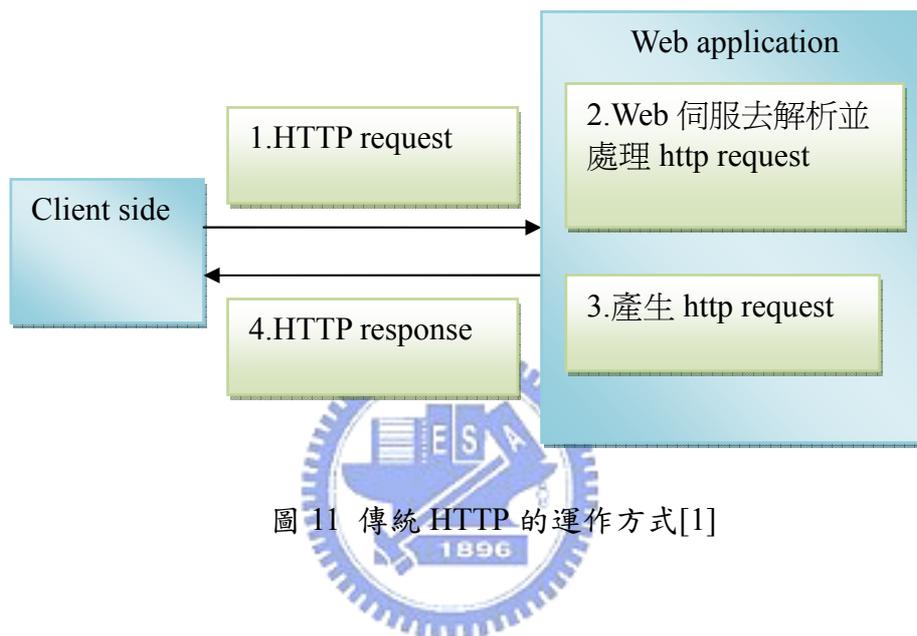


圖 11 傳統 HTTP 的運作方式[1]

超文件傳輸協定(HyperText Transfer Protocol; HTTP)是目前應用最廣泛的通訊協定(protocol)之一，原因是它的不具狀態性，也就是每當HTTP伺服器將處理結果回傳至客戶端，不會保留其客戶端執行狀態，並中斷其連線，可以讓伺服器的資源在執行更有效率。所以，在撰寫Web應用程式時至少要考慮以下的議題[1]：

- (1)網際網路的協定。
- (2)控制資料封包如何在網路上傳遞。
- (3)解析資料封包的內容。
- (4)根據客戶端的需求產生是當的回應。
- (5)傳送或顯示回應的訊息。

由上可知，開發一個功能完整的伺服器來執行應用程式是非常困難的。

因此為了解決上述問題，可以將整個Web應用程式架構分成兩個部分：Web應用程

式和Web伺服器。

Web應用程式負責處理客戶端的請求跟回應，網路上的所有訊息封包由Web應用程式負責接收，不去理會其處理的情形然後將結果傳給客戶端；而Web伺服器則包括只需要將客戶的需求處理產生正確的結果即可，就算資料形式需要變更也不會影響資料的傳遞方式。

在過去開發人員是以CGI程式(Common Gateway Interface Script)來撰寫Web應用程式，但是現在越來越多的開發人員開始用Servlet程式，所謂Servlet就是以Server端為運算平台，不僅可以傳送HTML的資料，也可以用來開發包含語音、影像、視訊等多媒體資料，是以需求/回應(Request/Response)為基礎的Java程式[13]。主要是因為Servlet是以Java技術為基礎的Web應用程式，擁有下列優點：

- (1) 執行效率佳：每一個Servlet均對應一個獨立的執行緒(Thread)，所以在執行的效率上會比CGI好。
- (2) 進入門檻低：Servlet是以Java為基礎，只要熟悉Java的程式設計師就可以輕易去撰寫各式各樣的Servlet。
- (3) 跨平台的能力：藉由Java本身跨平台的能力，Servlet也可以支援跨平台，可以讓系統架構更具可攜性。

Servlet的基本的運作方式如圖12所示。



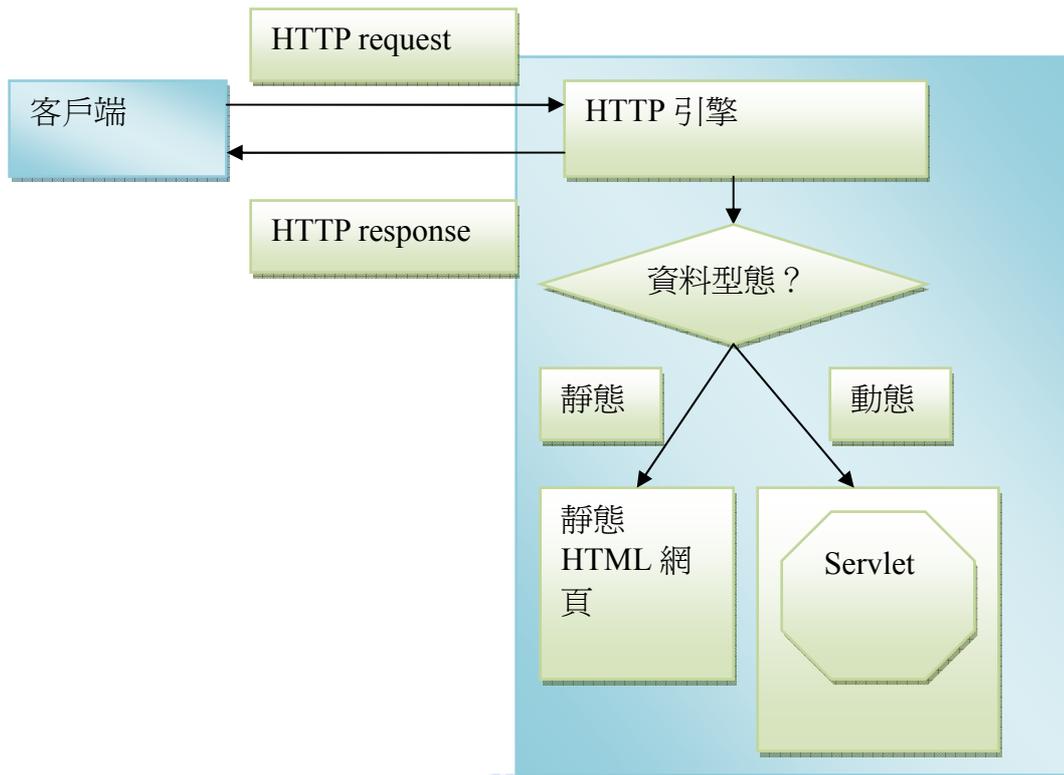


圖 12：Servlet 的運作情形[1]

當客戶端向 Web 伺服器送出 HTTP 請求時，Web 伺服器會先判斷請求內容：如果是靜態的 HTML 網頁資料，Web 伺服器會自行處理，然後產生回應訊息；如果是動態資料，Web 伺服器會將 HTTP 的請求轉交給 Servlet 容器處理，結果會送回給 Web 伺服器，再由 Web 伺服器傳給客戶端。

2-2-2 軟體開發的模式

軟體工程上，軟體的生命週期(software development life cycle)包含軟體開發過程的所有活動和建立的工作產品，主要分成兩類[2]：

- 1.以活動為主(activity-centered)：專注於整個軟體已開發過程的活動。
- 2.以實體為主(entity-centered)：專注整個軟體開發過程中，上述活動所建立的工作產品。

軟體開發過程必須具備的基本活動有：需求擷取(Requirement Elicitation)、分析(Analysis)、設計(Design)、實作(Implement)和測試(Testing)。而國內軟體生命週期開發的模式有很多種，有反覆與漸進式、瀑布式、螺旋式、V型、鋸齒狀、鯊魚齒狀、問題導向多種，但是除了瀑布式與反覆與漸進式這兩種，其他多半是為了解決某些問題而特別提出來的，下面就分只針對這兩種開發模式做深入介紹：

1 瀑布式開發過程：

瀑布式開發過程對程式開發者來說是一種比較直覺的開發過程，它就是執行一序列的開發和管理過程，此開發過程由於是一個階段完成在完成下一個階段，由於開發時間太長而無法快速的滿足客戶需求，所以被認為是過時的開發過程，不過由於軟體工程上有相當多的開發過程是瀑布式模型的變形，所以目前仍然是廣泛使用的開發過程，如圖 13 所示。

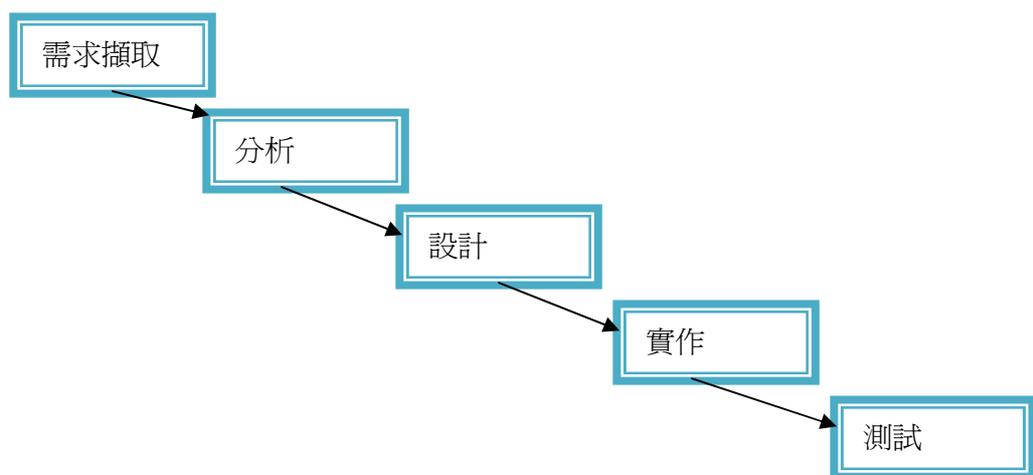


圖 13 瀑布式開發過程

其主要的問題如下：

- (1)開發的時間比較長，直到最後才能建立出應用程式。
- (2)當初的系統需求如果沒有定義完善，等到分析、設計或實作才發現後就必須回頭作一次，也就是問題越晚發現，所付出的成本就越高。

2 反覆與漸進式開發過程：

隨著物件導向技術的成熟以及瀑布式開發過程無法快速建立產品且缺乏彈性的問題，所以逐漸被反覆與漸進式開發過程取代。它主要會針對幾個主要需求來進行開發，建立初期版本的產品，然後交給顧客試用，程式開發者針對試用結果的回應修正系統，以便儘早發現錯誤。反覆的過程一直到滿足所有顧客的需求後，才開發出最新的版本的應用程式。例如：微軟的作業系統都會釋出 Beta1、Beta2 版等，並且釋出的版本會越來越快，直到完成所有應用程式開發。

而最著名的反覆與漸進式開發過程是 Rational 公司的 Rational 統一開發過程 (Rational Unified Process, RUP)，它與 UML 都是同源的方法論。



2-3 統一塑模語言的介紹

2-3-1 統一塑模語言的特點

軟體工程上，統一塑模語言(Unified Modeling Language ;UML)是非常重要的，它是一套可以用來架構軟體藍圖的標準模型語言，也是一種視覺化的物件導向的分析語言，可以讓程式開發人員可一目了然；由於統一塑模語言都是圖形表示的，亦即透過統一塑模語言表現設計的理念，可以讓第三者很清楚的了解整個設計的架構；且統一塑模語言沒有牽涉到實作的程式語言，只表達系統的規劃與設計理念，因此可以應用在不同領域；藉由統一塑模語言可以彈性的改變系統功能，因此可以避免因新增功能而必須將實作的程式做很大的變動，以達到可能增加的改變與複雜度所帶來的風險。

2-3-2 統一塑模語言的圖形種類

統一塑模語言定義了九種型態的圖形(Diagrams)：使用案例圖(Use Case Diagram)、類別圖(Class Diagram)、物件圖(Object Diagram)、循序圖(Sequence Diagram)、合作圖(Collaboration Diagram)、狀態圖(Statechart Diagram)、活動圖(Activity Diagram)、元件圖(Component Diagram)與部署圖(Deployment Diagram)，用來描述物件導向分析與設計的結果；而如果以觀點(view)的不同角度來分類，可以分為靜態觀點(static views)跟動態觀點(dynamics views) ，整理如表 3 所示[2]。

表 3 統一塑模語言所用的圖形[2]

符號圖形	說明	觀點
使用案例圖	描述系統功能以及所提供的服務	靜態
類別圖	描述物件的靜態結構	靜態
物件圖	描述物件的靜態結構	靜態
順序圖	使用時間軸方式描述物件間的互動，即物件間的訊息傳遞的時間順序	動態
合作圖	描述物件的互動，主要是表示物件間的關聯性，訊息流向和控制流程	動態
狀態圖	描述物件生命週期的物件和狀態轉換	動態
活動圖	描述使用案例的主要流程事件和操作，即工作流程所需的作業活動	動態
元件圖	表示系統實作的觀點，元件可以包含其他元件或多個類別	靜態
部署圖	表示一個系統的實地建置觀點，描述系統軟硬體的配置	靜態

接著對每一個圖形說明並舉例：

1. 使用案例圖：

使用案例圖是由動作者(actors)、系統邊界(system boundary)、使用案例(use case)之間的關聯性組成。

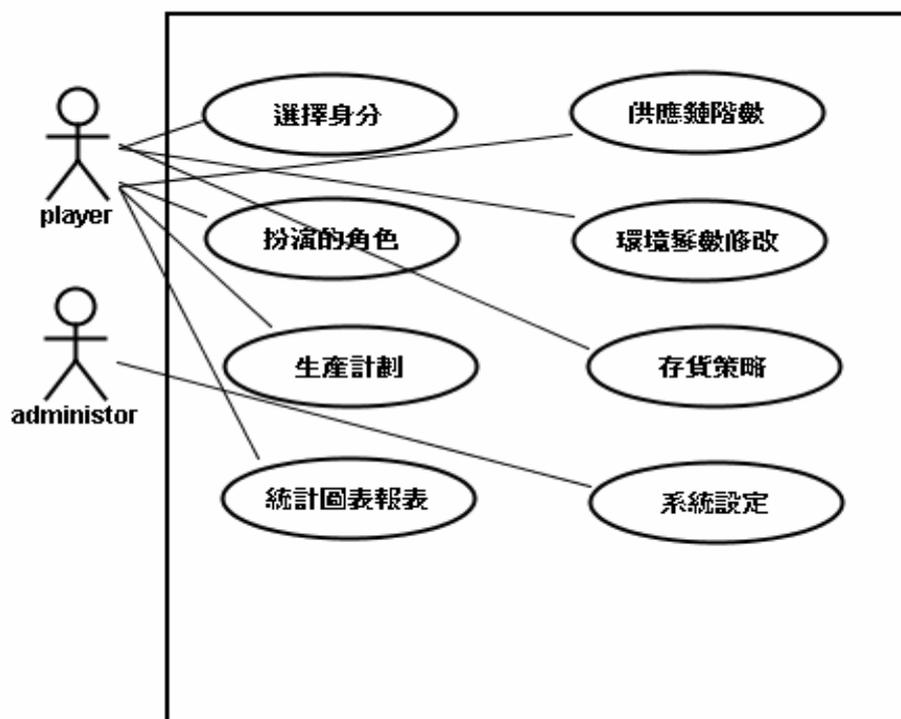


圖 14 使用案例圖(本研究整理)

上圖 14 中的的方框是系統邊界，在方框內是被觀察的系統。系統外部的動作者來連接橢圓形的使用案例來說明其關聯性。其中動作者是環境中，與系統有資訊交換的行為的人和事物；使用案例是描述動作者跟系統互動所產生的可能情境，用橢圓形來表示。

2. 類別圖：

類別圖是一般物件導向模型最常見的圖形，包含了一群的類別、介面和它們之間的關係。類別圖描述了一個系統的靜態結構，或是如何建立此結構，而非它的行為，如圖 15所示。

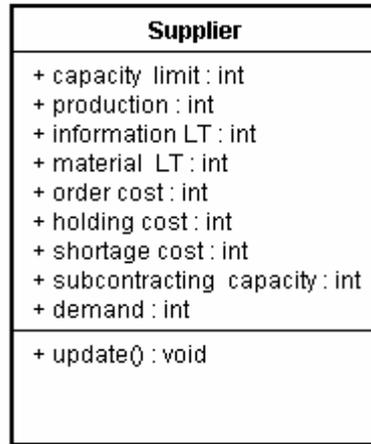


圖 15 類別圖(本研究整理)

類別圖可以分成三個區格(compartment)，最上層的部分是類別名稱，中間是類別的變數；最下層是類別的方法。變數跟方法前的符號為存取修識子(access modifier)，稱為可見度(visibility)。每個修識子都有不同的存取權限範圍，如表 4 所示。

表 4 類別圖屬性的可見度

存取修飾子	說明	存取範圍
+	public	全域範圍，包含子類別
-	private	只能在同一個類別存取，不可以在子類別存取。
#	protected	可以在同一個類別，其子類別或同一個套件中存取，包含不同套件的子類別。
~	package	預設範圍是同一個類別和套件中存取，但不包含不同套件的子類別。

變數的型別跟函數的型別，列於變數或是參數名稱冒號(:)的後面，同樣的函數的型別也是一樣。

類別的關連性(relationships)又有下列幾種，分別說明如下：相依關係(Dependency)，聚合關係(Associations)，聚合關係(Aggregation)，組成關係(Composition)，一般關係(Generalization)，分別說明如下：

(1).關聯關係：表示類別之間靜態的概念性關係，意味著一類別之物件知道另一類別之

物件的存在，或是使用另一類別之物件的服務。在每個關連端點可以用如 1、* 和 0..*（表示沒有或多個）等等的符號表示出它們彼此間的多重性。

關聯關係有兩種特定變異，一種是聚合關係（aggregation），一種是合成關係（composition）。前者是關係比較強的結合關係，屬於成品和零件(Whole-part)的類別關係，使用空心菱形的實線表示「同性質」的關係（a kind of），如圖16所示。

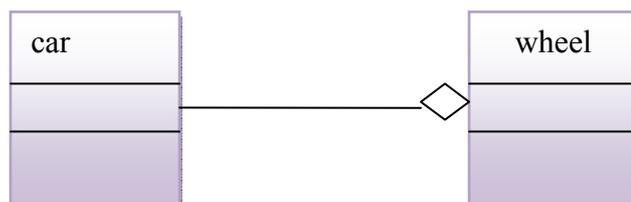


圖 16 關聯關係(Whole-part)

上述car類別的wheel類別，wheel類別是通用零件，所以相同的車輪可以使用在其他車上。

後者是關係最強的關聯關係，表示一種專屬零件，使用實心菱形的實線來表示「部份」（a part of）的關係，如圖17所示。

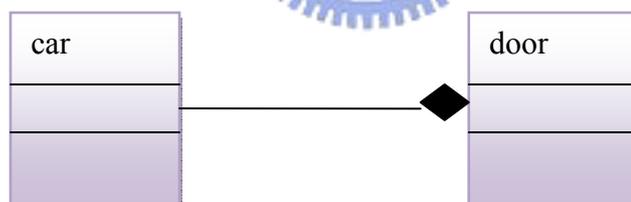


圖 17 關聯關係(a part of)

上述car類別的door類別，每一種車一定擁有車門，而且車門是一種專屬零件，並不能用在其他車輛上。所以由上可知道組合關係跟聚合關係主要差異為聚合關係的零件則是可以使用在其他成品的通用零件；組合關係的零件是只能使用在成品的專屬零件。

(2).一般化關係：指的是繼承關係，是一般性的類別（稱為父類別）與某些特定性質的類別（稱謂子類別）之間的關係，使用空箭頭的實線從子類別指向父類別，。其典型例子是企業中的“個人顧客”、“公司顧客”，兩者之間並不相同，但有許多的相似的特性，可以放在一般性“顧客”的類別裡，如圖18所示。

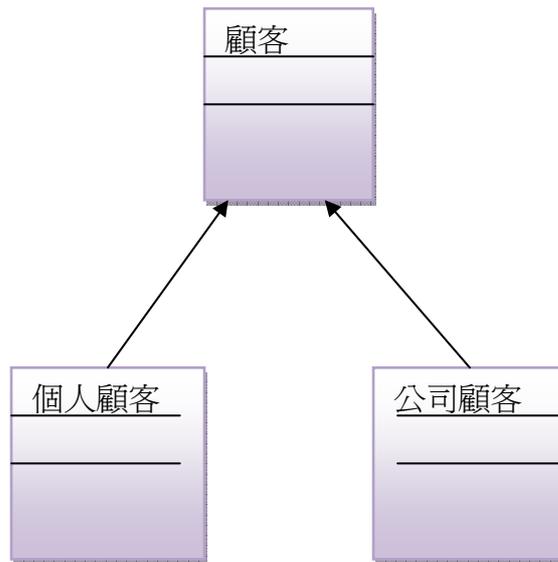


圖 18 關聯關係(一般化關係)

(3).相依關係：使用虛線箭頭表示的一種非常弱的結合關係。當一個類別相依其他類別時，而且被使用類別的改變可能會影響到使用它的類別，反之則不必然成立。

3. 物件圖：

物件圖類似類別圖，可以表示某特定時間的點的物件關係。其差別為顯示物件實例 (instance)，而不是類別，如圖 19 所示。

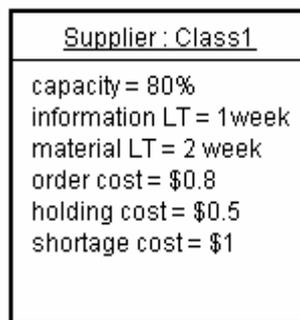


圖 19 物件圖(本研究整理)

4. 順序圖&合作圖：

順序圖是以時間軸顯示物件之間的訊息傳遞；合作圖是以空間的角度顯示物件的之間傳遞的路徑。彼此之間可以互相轉換。它包含以下元素：

(1). 物件：

以方框表示，在物件完整名稱“：”冒號前是物件名稱，之後是類別名稱。

(2). 生命線 (Lifeline) & 活期棒 (Activation Bar) :

物件方框下的垂直虛線下是生命線，中間窄長方形稱為活化期棒，表示物件存活的時間。

(3). 訊息 (Message) :

表示物件之間所進行的溝通，表示從一個物件的生命線到另外一個物件的生命線，可以讓接收的物件進入活化期棒執行所需的操作。

以下圖 20 來說明：玩家 Y 要登入某一遊戲系統時，在輸入的名稱、角色以及參數設定之後送出訊息給 HTML 物件，HTML 物件送出訊息給 Servlet 物件，對 Y 所輸入的名稱、角色以及參數設定做合法性的確認，看是否輸入有錯誤，若沒有錯誤，Servlet 就會回傳正確訊息回傳給 Y 玩家。如果確認有誤，就會回傳警告的訊息，要求 Y 玩家重新輸入。



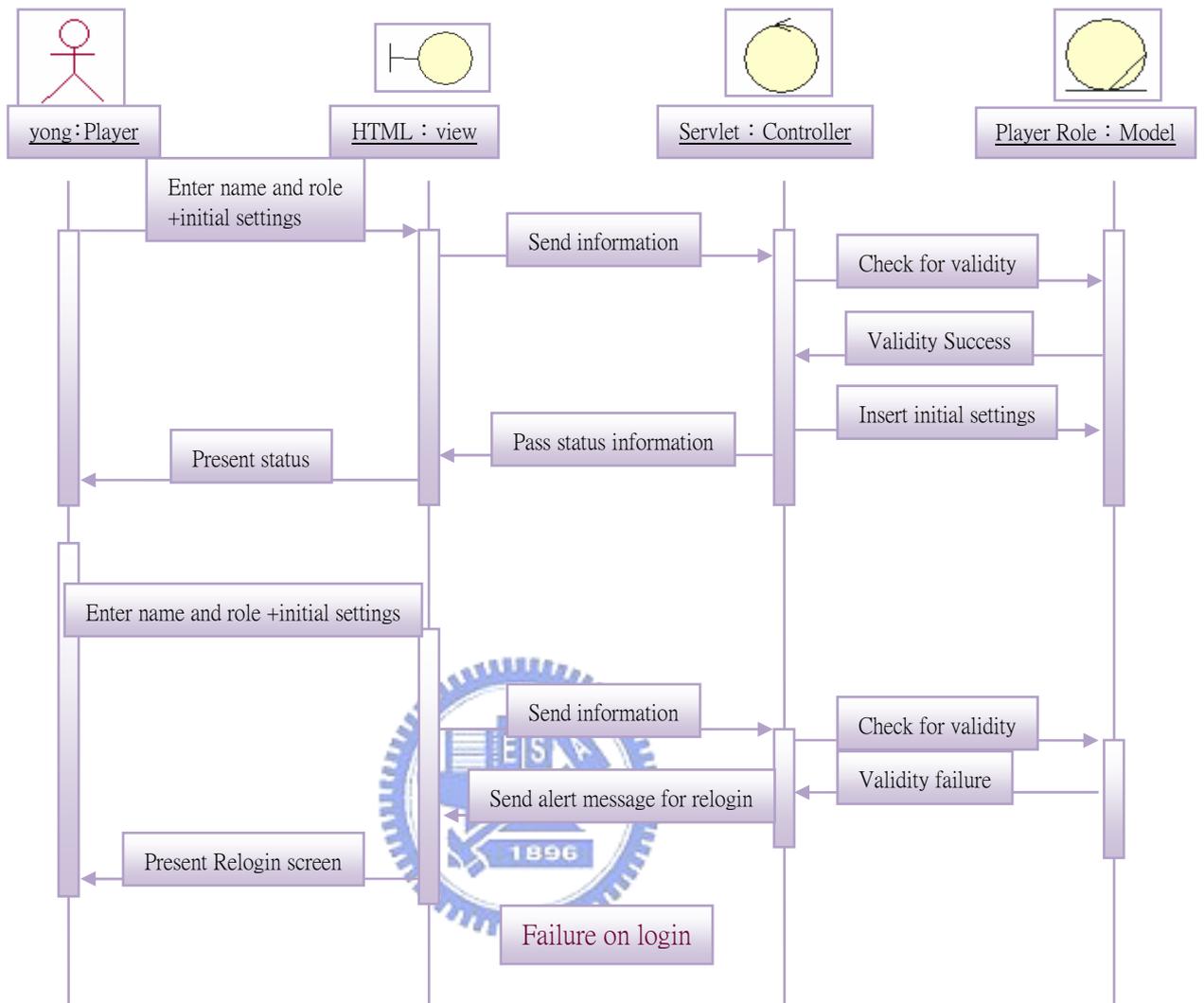


圖 20 順序圖

5. 狀態圖：

狀態圖是用來描述物件生命週期的事件與狀態轉換，狀態因為事件而改變，並且會影響下來的操作。以圖 21 來說明：

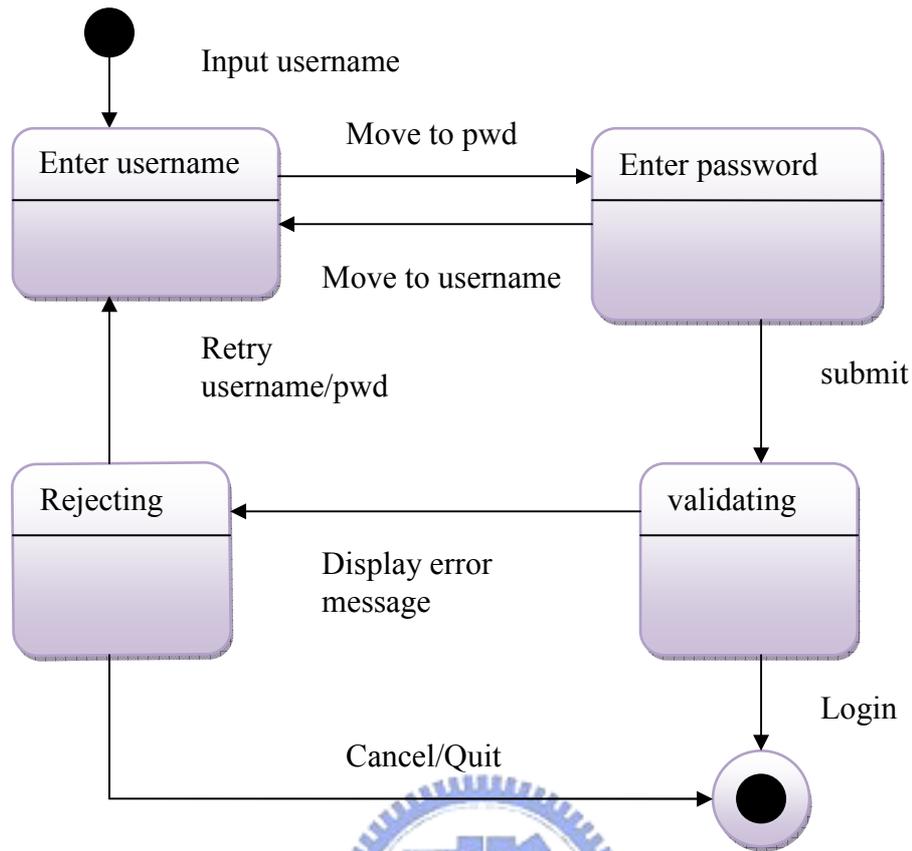


圖 21 狀態圖[2]

上述狀態圖中，實心圓跟同心圓表示狀態的開始與結束，圓角長方形是狀態，狀態間的箭頭表示狀態間的轉換。狀態圖首先從游標移至使用者帳號欄位開始，當輸入後，就會觸發 move 事件移至密碼欄位輸入密碼，然後在觸發 submit 事件執行驗證，如果通過驗證就成功登入系統，沒有就進入拒絕狀態。

6. 活動圖：

活動圖是用來描述使用案例的主流程事件與操作，就是整個流程所需的作業活動，簡言之就是統一塑模語言的流程圖。以下圖來說明：下圖有兩個活動區 (Swimlane)，分別屬於 Player 和 System 兩個角色的活動，玩家登入後，系統就會檢查玩家登入的帳號跟密碼是否吻合，若吻合則顯示決策表，玩家在填完決策表後，系統就會檢查其決策表的資料是否合法，若合法就會預先調整其系統的狀態 (如定時器等)，接著系統會問玩家是否要查看決策表的摘要，最後玩家可以決定是否繼續，若否就會離開遊戲，反之則回到決策表，讓玩家再做下一個決策。

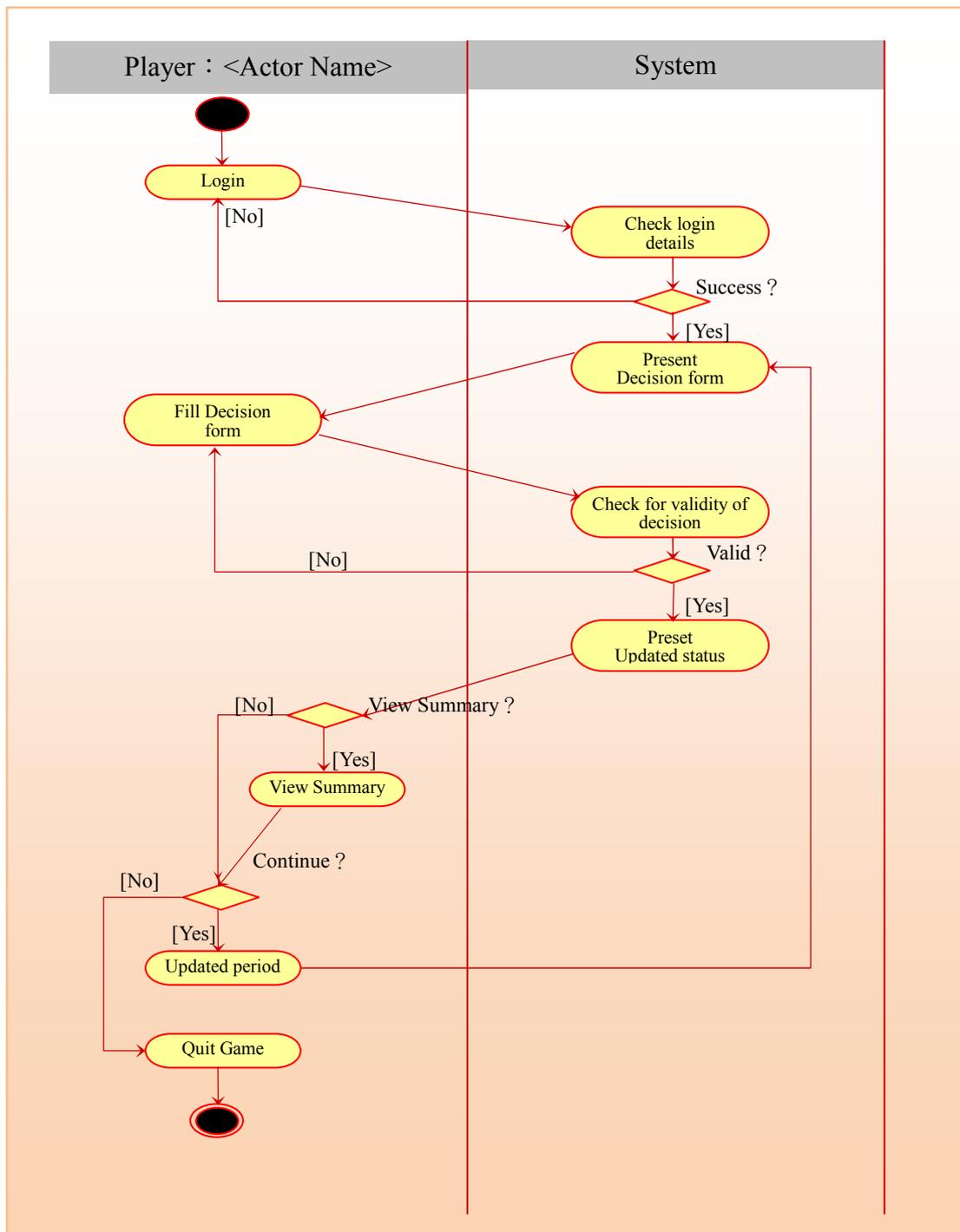


圖 22 活動圖[12]

7. 元件圖：

元件圖可以提供系統的實際觀點，可以顯示組成軟體系統元件之間的相依關係，並描述各元件的組織架構。如圖 23 所示：

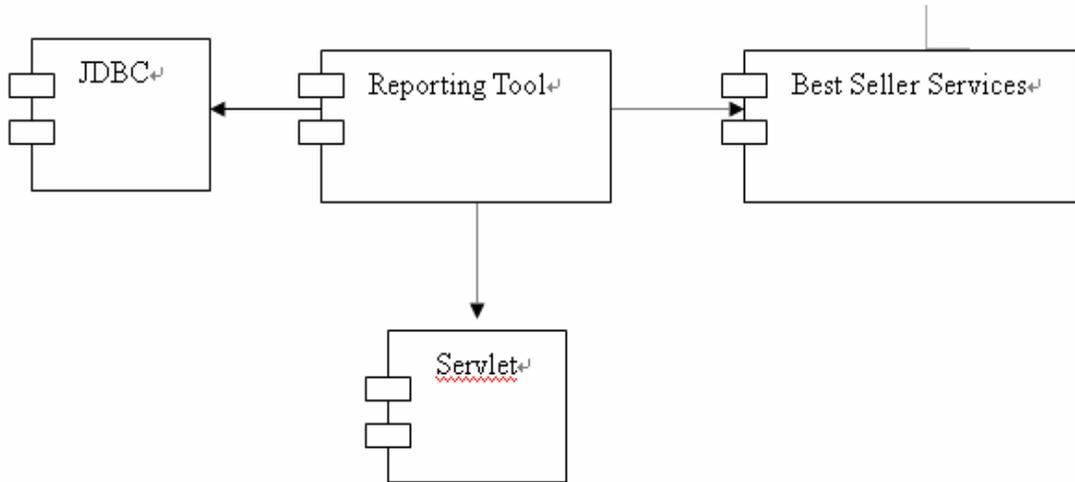


圖 23 元件圖 [2]

上述元件包含 JDBC，Reporting Tool，Best Seller Services 和 Servlet 共四個元件。Reporting Tool 元件同時相依於其他三個元件。

8. 部署圖：

部署圖則是顯示系統如何部署在電腦軟體系統的環境，可顯示電腦系統和周邊設備上軟硬體的環境。

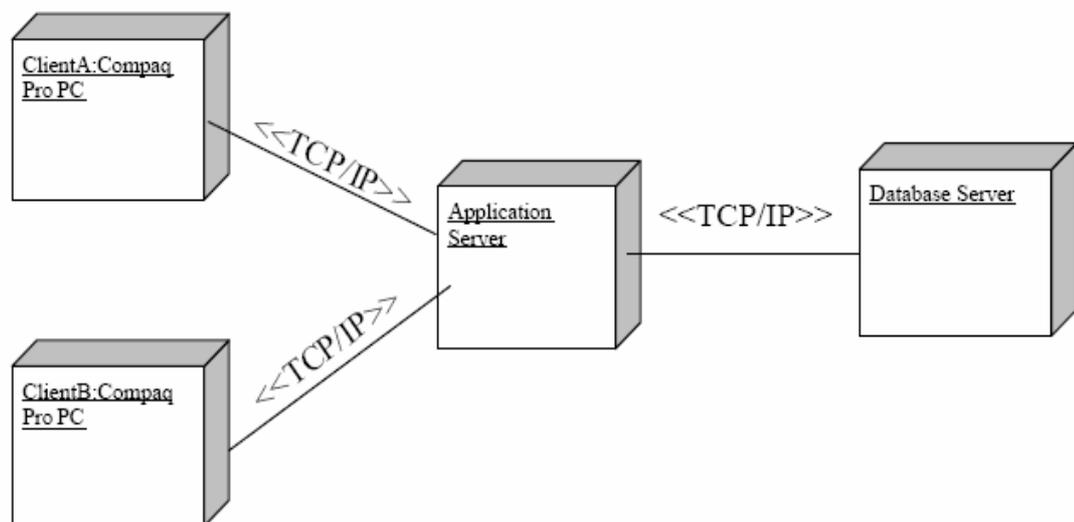


圖 24 部署圖[3]

上圖24說明個人電腦透過網際網路向伺服器請求一些資訊的示意圖。Client 透過 TCP/IP協定連上Application server，Application server從資料庫裡搜尋資料，再將結果回傳給Client。



2-4 相關教學文獻探討

由於本研究是要建立一個遊戲的教學平台，之前在如何應用遊戲以提升教學的效率上，許多人也做了很多相關的研究，例如：

許瑛瑄等[7]以情境學習理論為基礎，此理論強調學習者與其本身所處環境互動之歷程，但一般傳統教室無法提供此逼真的情境，而隨著網路科技的快速竄起，網路超媒體特性及互動模式，為情境理論提供良好的發展空間。因此，此篇融合情境教學之要素及網路學習環境之特色，發展一套地球科學網路輔助教材，以高中生為研究對象，探討學習者在情境式網路輔助教學中概念學習成效與對此網路輔助教材之綜合意見。研究結果發現：學習者在此情境式網路輔助教學後，概念測驗得分幾乎達到顯著進步，顯示情境式網路輔助教學有助於增長相關知識；在情境式網路教材整體觀感方面，大多數亦認為此情境式網路學習環境是有效且值得推廣的，且認為以生活為議題的情境，能提高學習者的學習動機及幫助其整合新舊知識。

由於電子商務應用為資訊時代新興的經營模式，也是一種需要高度整合的複雜應用科技，黃雲龍[8]嘗試規劃電子商務虛擬企業的情境教學策略，增加學習者對於真實問題的體驗與相關知識與技能的訓練，並利用網路輔助教學建構行動學習歷程檔案與合作學習的課堂創新效果。此情境教學命名為：「2002 Virtual EC 電子商務競賽」，情境設計包含六個相關的決策：市場商機分析、經營模式、顧客界面、網路行銷與品牌塑造、實作導入以及評估。以因素分析檢視內在結構，得到七個主要向度包含：教學者領導效能、課程設計理念、小組團隊運作、個人學習風格、資源取得運用、學習成長活動、網路環境氣氛，將其依據各因素的相關性，以加權平均方式得到一個指標分數（4.55），顯示課堂整體的創新氣氛佳，特別是在小組團隊運作、學習成長活動、網路環境氣氛因素感受更強。證實本研究提出的虛擬企業情境教學模式，能夠營造一個鼓勵創新思考的學習環境。此經營競賽遊戲教學之特點為透過電腦設備配合可將學習成果迅速回饋，並運用電腦強大的計算統計功能製造出良好的動態環境，此遊戲能讓學生磨練其數字分析、決策與人際互動的能力，並協助建立整體觀念。

蔡學偉[9]針對國中一年級學生為對象，以電腦樂高網路輔助教學方法設計教學網站，評估此方法實施成效及瞭解學生學習反應情況。經實驗結果(問卷填答)顯示，在學習者上課之初能有效引起學習者之學習動機，對課程內容及目標有初步的瞭解。就學習者在課後使用而言，利用網路教材可達成課後複習之目的，增進學習成效，不同於以往的管道，可與更多的使用者進行互動交流。

O'Leary et al. [13]以醫學系學生為研究對象，針對「產科」此學科將學習者分成兩組施以不同的教學方式，即傳統授課與教學遊戲，比較不同方法對學習者之效果及滿意度是否有差異。依據事前、事後測驗顯示，此兩組雖透過不同的教學方式，但學習者在此門學科的學習上皆有顯著的改善；但在滿意度方面，施以教學遊戲的學習者對此門課有較大的興趣，且此教學遊戲具有模擬等優點，也較能激起學習者之參與感。

Mann et al. [14]認為電腦輔助遊戲對醫學教育實為一有效之教學工具，依據此概念，應用電腦技術更能模擬出精確的病患情境讓學習者深入其境。在此，對學習者在使用此教學遊戲前、後做測試，結果發現玩過此遊戲的成績顯著高於之前，且在測試問題中有一半以上的問題皆有顯著進步；另外，在相關知識方面對學習者也有正面價值的成效。



以上文獻經由整理如下表 5 所示：

表 5 文獻整理表

	作者	應用領域
1	許瑛珺、廖桂菁(2002)	地質學-地球科學
篇名	情境式網路輔助學習環境之研發與實踐	
結論	1. 幫助學習者增長相關知識 2. 提高學習動機及幫助整合新舊知識	
2	黃雲龍(2004)	電子商務
篇名	電子商務情境教學之行動研究	
結論	1. 營造出一個鼓勵創新思考的學習環境 2. 協助建立整體觀念	
3	蔡學偉(2004)	自然科學
篇名	問題導向學習於網路輔助電腦樂高課程之研究	
結論	1. 有效引起學習者之學習動機 2. 幫助複習及提供互動交流	
4	O'Leary et al.(2005)	醫學-產科
篇名	Educational games in an obstetrics and gynecology core curriculum	
結論	1. 學習效果有明顯改善 2. 激起學習者之參與感	
5	Mann et al.(2002)	醫學-外科
篇名	The development of an interactive game-based tool for learning surgical management algorithms via computer	
結論	1. 學習效果有顯著進步 2. 相關知識亦有正面幫助	

第三章 系統架構與研究方法

3-1 系統架構

本研究主要是架設以一個遊戲為主的生產及物流的教學平台，以一條簡單四階的供應鏈為基礎，其角色包括原料供應商、製造商、物流中心與零售商。未來會根據系統管理者的設定，可以更有彈性地增加系統中各角色的數量與階數改其階數及其多樣化的參數。圖 25 為本系統供應鏈模式示意圖。

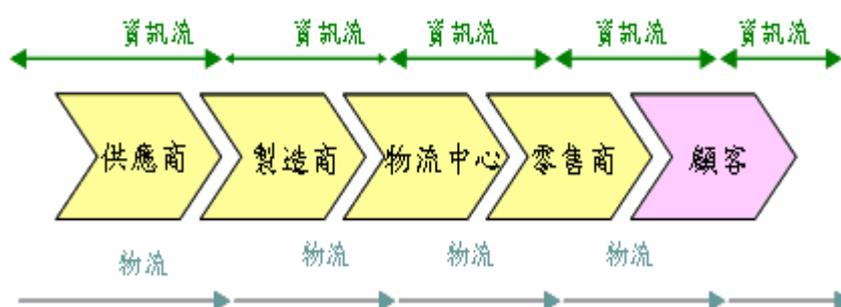


圖 25 系統模式說明

遊戲一開始當使用者登入系統的時候，就可以選擇所扮演的角色，有原料供應商，製造商，物流中心，零售商四種角色可供選擇，再設定其環境變數（如增加產能、外包策略等）跟營運生產變數（如存貨策略），參數設定完畢後，遊戲將根據系統管理者所設定的執行時間進行，玩家在遊戲執行中可視情況更改策略，並於結束後檢討遊戲結果。

3-2 需求擷取

需求擷取主要是以使用者的角度去思考系統中會用到的功能，主要分成功能需求與非功能需求，功能需求就是描述系統一定會提供的操作，例如本系統提供的選擇角色、訂購產品、顯示圖表等選項的功能。非功能需求就是描述系統特性或一些非限制的條件，例如每個角色策略的選擇。

程式開發者是以使用者案例模型(Use-case Model)來描述系統需求，包含行為者(Actor)和使用案例(Use Cases)。行為者是與系統互動的外部實體，包含使用者、其他軟體系統或子系統，例如：本系統的學生是行為者。使用案例是描述行為者跟系統互動所產生的可能操作，這是以客戶的語言來描述特定功能的情境(Scenarios)，並詳細說明可能產生的操作步驟和事件的劇本，例如以本系統來說明如下：

- 1.系統顯示使用者可以選擇的角色以及相關參數設定
- 2.使用者執行訂貨策略並顯示每回合結果
- 3.使用者是否繼續執行下一回訂貨策略
- 4.結束遊戲

3-3 分析

擷取系統需求之後，就可以針對需求進一步做需求分析，讓系統需求抽象化到應用系統中，也就是對每一個需求找到對應的解決方案。

本系統是以一個簡單的供應鏈為基礎，在供應鏈中，不同角色分別肩負不同的任務，才得以建構一條良好且具競爭力的供應鏈；為了突顯每個角色的不同，本研究分別做不同的參數與操作上的設定，除了可讓教學上應用更貼切，也可以更符合實際供應鏈的作業流程。本研究對每個角色針對它的訂購量、生產產能、倉儲空間以及前置時間的做不同的分析：

3-3-1 原物料供應商

原物料供應商是屬於供應鏈中最上游的，其功能為取得自然界中資源以提供下游製造商生產產品時所需之原物料。所以在定位上其訂購量跟生產產能跟其他角色會受到限制是不同的，分別說明如下：

1. 訂購量：無限制。
2. 生產產能：無限制。
3. 倉儲空間：
 - (1)無限制。
 - (2)有限制：初始倉儲空間為 30 單位，可以新增額外倉儲。
4. 前置時間：1 週至 5 週，可由玩家設定，預設值為 2 週。

訂購策略(此角色由電腦扮演)：

- (1) ROP-Q：存貨水準低於再訂購點(ROP)時，即訂購數量 Q。
- (2)ROP-S：存貨水準低於再訂購點(ROP)時，即將存貨水準補充至 S。

(3) Order-to-Q：每週即訂購數量 Q 。

(4) Order-to-S：每週訂購，將存貨水準永遠保持在 S 。

(5) Random：每週隨機訂購不同數量。

3-3-2 製造商

製造商向上游原物料供應商訂購原料，依產能規劃加工為最終產品，並接獲下游物流中心之訂單，盡可能出貨以滿足物流中心，分別說明如下：

1. 訂購量：向原物料供應商訂購原料，以一般產能與加班產能總和為極限。

2. 生產產能：一般產能(有限制)、加班產能(有限制)和外包產能(無限制)。

3. 倉儲空間：

(1) 無限制。

(2) 有限制：初始倉儲空間為 30 單位，可以新增額外倉儲。

4. 前置時間：1 週至 5 週，可由玩家設定，預設值為 2 週。

訂購策略(此角色由電腦扮演)：1896

(1) ROP-Q：存貨水準低於再訂購點(ROP)時，即訂購數量 Q 。

(2) ROP-S：存貨水準低於再訂購點(ROP)時，即將存貨水準補充 S

(3) Order-to-Q：每週即訂購數量 Q 。

(4) Order-to-S：每週訂購，將存貨水準永遠保持在 S 。

(5) Random：每週隨機訂購不同數量。

3-3-3 物流中心

製造商生產完之產品均運送至此，為產品的集散中心，在此處規劃產品的運送策略，即運送到何處的零售商及運送數量。本系統是假設所有的訂單都必須被滿足。

1. 訂購量：無限制。

2. 生產產能：無限制。

3. 倉儲空間：

(1)無限制。

(2)有限制：初始倉儲空間為 30 單位，可以新增額外倉儲。

4. 前置時間：1 週至 5 週，可由玩家設定，預設值為 2 週。

訂購策略(此角色由電腦扮演)：

(1) ROP-Q：存貨水準低於再訂購點(ROP)時，即訂購數量 Q。

(2)ROP-S：存貨水準低於再訂購點(ROP)時，即將存貨水準補充至 S。

(3)Order-to-Q：每週即訂購數量 Q。

(4)Order-to-S：每週訂購，將存貨水準永遠保持在 S。

(5)Random：每週隨機訂購不同數量。

3-3-4 零售商

在供應鏈中，唯一與顧客和市場需求有直接接觸之角色，可觀察到真實的市場需求；因此，零售商必須能夠準確預測未來市場需求，適時適量補充存貨，以便能夠及時提供顧客所需產品，進而提高顧客滿足度。

1. 訂購量：無限制。

2. 生產產能：無限制。

3. 倉儲空間：

(1) 無限制。

(2) 有限制：初始倉儲空間為 30 單位，可以新增額外倉儲。

4. 前置時間：1 週至 5 週，可由玩家設定，預設值為 2 週。

訂購策略(此角色由電腦扮演)：

(1) ROP-Q：存貨水準低於再訂購點(ROP)時，即訂購數量 Q。

(2) ROP-S：存貨水準低於再訂購點(ROP)時，即將存貨水準補充至 S。

(3) Order-to-Q：每週即訂購數量 Q。

(4) Order-to-S：每週訂購，將存貨水準永遠保持在 S。

(5) Random：每週隨機訂購不同數量。

3-4 設計

在設計是建立完整的解決方案，以更詳細的方式描述如何建立整個軟體系統，來滿足整個擷取出的系統需求。本研究先設定每個角色細部需求描述與分析，再利用 UML 中各種的塑模圖形完成本系統的設計，並產出各相關的文件，然後用特定語言去實作。

由於 UML 的方法論很多，本系統是以物件導向的方式[11]來設計作為其方法論，依照方法與步驟，配合 UML 將系統的概念表達出來。

在塑模的部份，本研究完成的有：

- (1).使用案例圖：描述本系統的抽象功能需求，並且以使用者的觀點來分析系統應該提供的功能，以對整個問題領域有個大輪廓。
 - (2).概念塑模：描述問題的內部概念意義，避免陷入太細節的系統設定，反而看不出整體的概廓。
 - (3).類別圖：以軟體的角度考慮系統設計的議題，將原先抽象的概念轉化成系統中一個個的類別。
 - (4).互動圖：以互動圖來表示物件之間動態的互動行為，據以描述系統與使用者的真實運作與互動情境。
- 其餘屬於實作細節的部份，則不納入討論。

3-4-1 建立使用案例圖

3-4-1.1 找出行為者

根據之前分析的結果，找出其行為者有使用者(一般玩家)跟管理者，分別敘述如下：

- 1.使用者可透過參數設定以模擬不同的情境。
- 2.使用者可選擇整個系統的模組(供應鏈或是工廠模式)。
- 3.使用者可透過圖表來了解整個系統成本花費的情形。

3-4-1.2 找出使用個案

根據系統需求，行為者與系統互動的相關事件可歸納如下：

1. 使用者透過本系統的參數設定選擇想要的情境，以配合實際與教學所需。

(1) 使用者選擇角色。

(2) 使用者訂購產品。

2. 使用者可透過本系統的模組設定來選擇遊戲的進行的方式，選擇供應鏈模式還是工廠直營模式；選擇集中化管理的選項時，在供應鏈的模式下，會只能扮演供應商的角色。

(1) 使用者選取模組選擇。

3. 使用者可透過圖表來了解整個系統成本花費的情形

(1) 使用者選取顯示圖表。

由上我們可以統整本系統有：角色選擇、策略、倉儲空間、集中化管理、物流前置時間、需求分配、資訊透明化、模組選擇和圖表等九種使用個案。

3-4-1.3 描述使用個案

描述使用個案焦點會放在互動的事件與功能的描述，而不是電腦細部的動作。使用個案之描述，應表達事件之起始行為者、動作、及參與動作之物件等，並描述事件之起始狀態與結束狀態本研究整理前述之使用個案如表6、7、8、9所示。

表 6 使用個案描述：選擇角色

使用個案：選擇角色
行為者：使用者
前提：無
結束狀態：參數存入系統
事件： 1. 使用者點選角色選擇 2. 系統找到對應的角色 3. 系統介面上會將所選的角色鎖住，其他角色會是反白的

表 7 使用個案描述：訂購產品

使用個案：訂購產品(供應鏈模式下)
行為者：使用者 前提：選擇角色、模組選擇以及相關的參數設定完成後 結束狀態：執行遊戲規定的回合數後，即完成訂購產品
事件： 1. 使用者開始執行遊戲 2. 使用者先輸入使用者的市場需求 3. 使用者在輸入使用者的訂購量 4. 重複3直到遊戲規定的回合數結束

表 8 使用個案描述：選擇模組

使用個案：選擇模組
行為者：使用者 前提：無 結束狀態：參數設定存入系統
事件： 1. 使用者可點選是否要點選模組選項 2. 系統會根據使用者選擇，在介面上會顯示供應鏈模式或是單一模式

表 9 使用個案描述：顯示圖表

使用個案：顯示圖表
行為者：使用者 前提：遊戲執行過程中或是執行完成後，便可以選擇圖表依照需要觀看其總成本圖、市場趨勢圖或是成本結構分析圖 結束狀態：系統顯示圖表
事件： 1. 在遊戲進行的途中或是結束，使用者可點選圖表的選項 2. 系統會根據使用者選擇：總成本圖、市場趨勢圖、成本結構分析圖，出現相對應的視窗，讓使用者方便討論

3-4-1.4 繪製使用案例圖

完成上述使用個案描述之後，所繪製使用個案圖26如下：

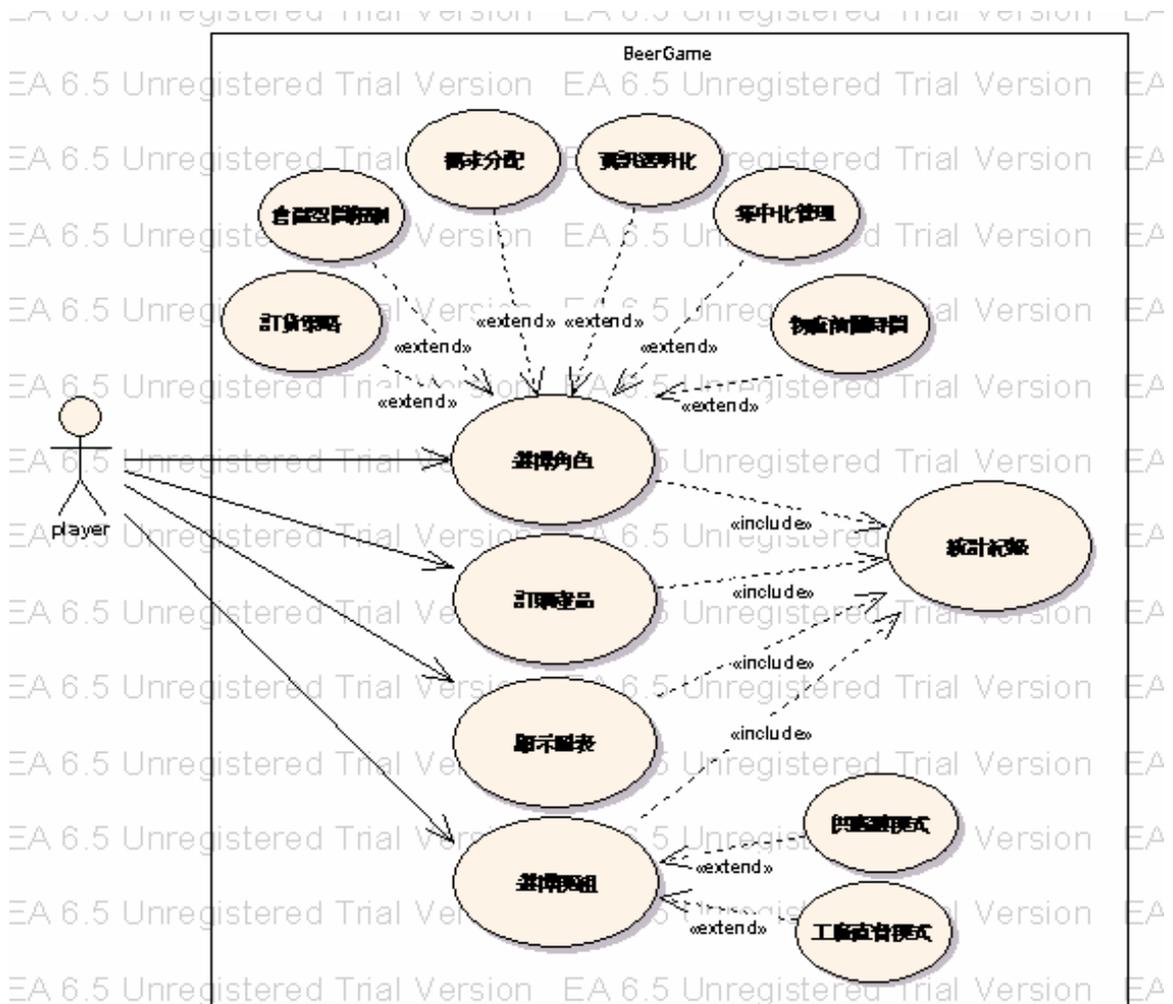


圖 26 使用案例圖

由於一開始執行遊戲時，使用者定會選擇自己所扮演的角色(供應商、製造商、分配商、零售商)，所以”角色選擇”為必需的使用案例，在來參數選擇方面，可以選擇電腦預設的，也可以選擇自行調整的，所以在“訂貨策略”、“物流前置時間”、“需求分配”、“倉儲空間限制”、“集中化管理”都 extend 到”選擇角色”這個使用案例。

3-4-2 概念塑模

使用個案完成系統的需求描述與分析後，我們對整個問題領域有了一個粗淺的描述，再來就是利用物件導向分析裡的概念模型來描述這些問題領域的概念意義。

使用個案之描述，應表達事件之起始行為者、動作、及參與動作之物件等，並描述事件之起始狀態（前提）與結束狀態，其格式應採主詞 + 動詞 + 受詞方式描述。其中主詞可視為一個行為者，動詞為一項處理，而受詞可為資料或另一個行為者。

在UML中，本研究使用靜態結構圖來做概念模型的塑模工具，利用先前完成的使用個案當作本研究塑模的對象，如下表10、11、12所示。

表 10 概念塑模 :選擇角色

使用個案：選擇角色
行為者：使用者 前提：無 結束狀態：參數設定存入系統
事件： 1. 使用者點選角色選擇 2. 系統找到對應的角色 3. 系統介面上會將所選的角色鎖住，其他角色會是反白的

根據找出的名詞，可以繪製UML圖形如圖27所示。

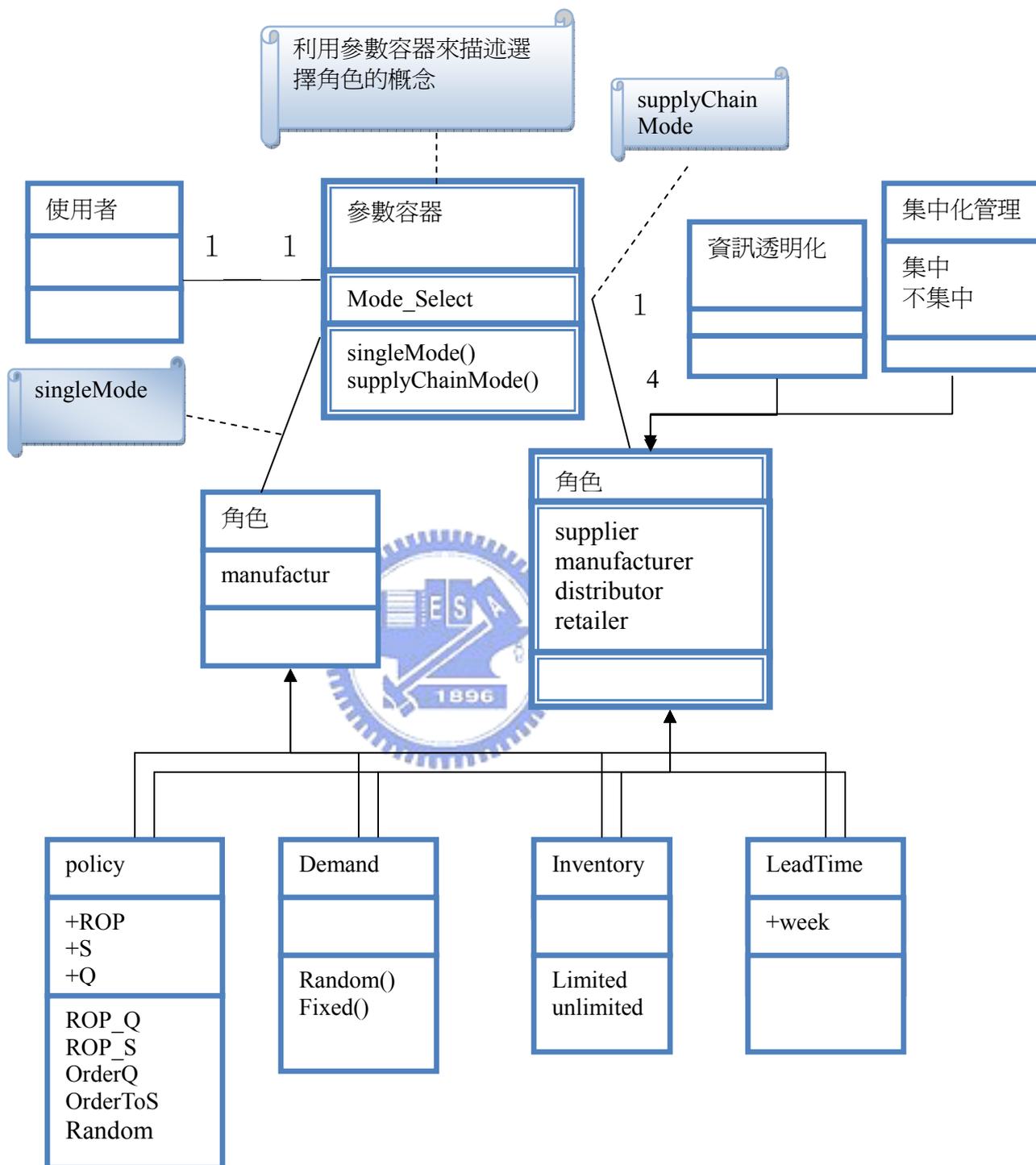


圖 27 類別圖 選擇角色

表 11 概念塑模：訂購產品（執行遊戲）

使用個案：訂購產品
行為者：使用者 前提：選擇角色、模組選擇以及相關的參數設定完成後 結束狀態：執行遊戲規定的回合數後，即完成訂購產品
事件： <ol style="list-style-type: none">1. 使用者開始執行遊戲2. 使用者先輸入使用者的市場需求3. 使用者在輸入使用者的訂購量4. 重複3直到遊戲規定的回合數結束

根據找出的名詞，可以繪製UML 圖形如下，共分成兩類，一個是共應鏈模式，一個為工廠直營模式，分別繪圖如下：



1. 供應鏈模式所繪製的 UML 圖如圖 28 所示。

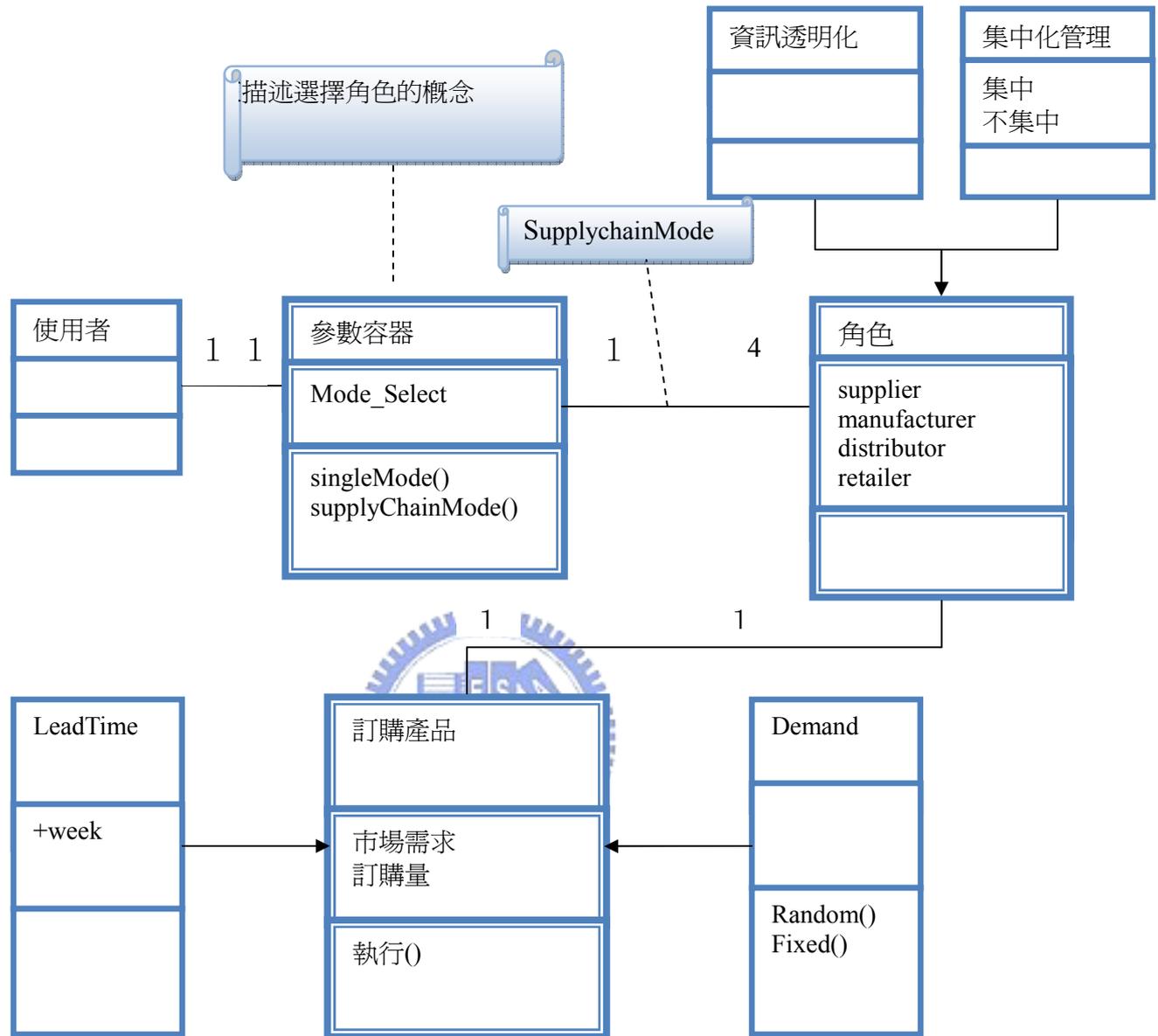


圖 28 訂購產品 (供應鏈模式)

2. 工廠直營模式所繪製的 UML 圖如圖 29。

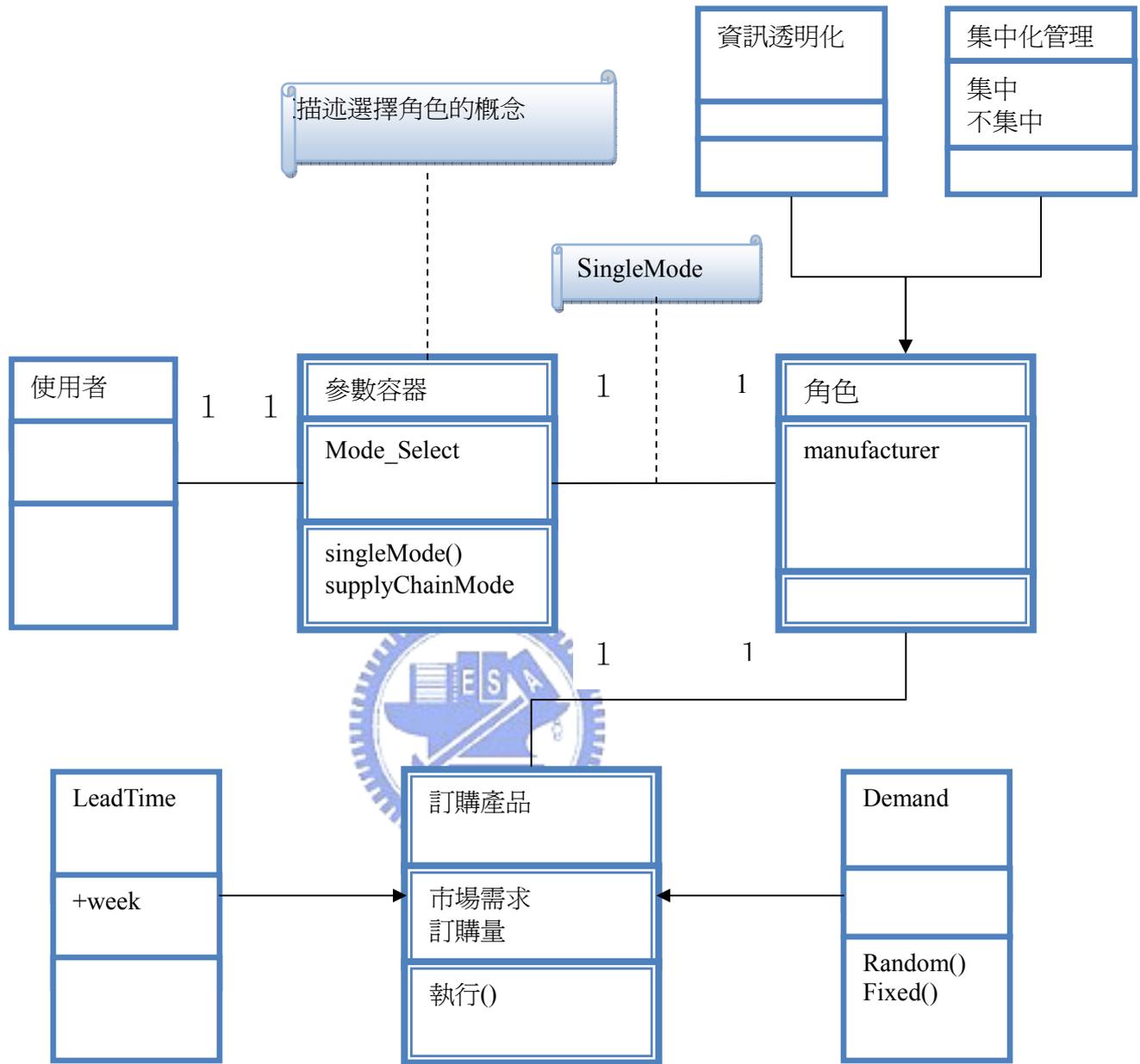


圖 29 訂購產品 (工廠直營模式)

表 12 概念塑模：顯示圖表

使用個案：顯示圖表
行為者：使用者
前提：遊戲執行過程中或是執行完成後，便可以選擇圖表依照需要觀看其總成本圖、市場趨勢圖或是成本結構分析圖
結束狀態：系統顯示圖表
事件： <ol style="list-style-type: none"> 1. 在遊戲進行的途中或是結束，使用者可點選圖表的選項 2. 系統會根據使用者選擇：總成本圖、市場趨勢圖、成本結構分析圖，出現相對應的視窗，讓使用者方便討論

根據找出的名詞，可以繪製 UML 圖形如圖 30 所示。

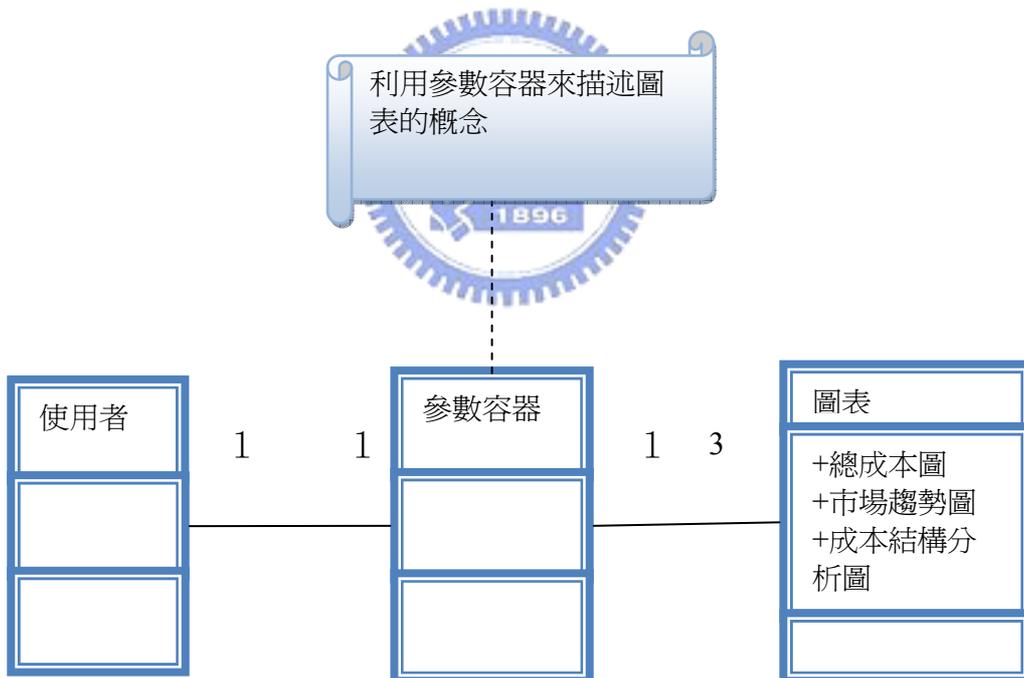


圖 30 顯示圖表

3-4-3 建立類別圖

以「概念模型」完成系統的問題領域概念的描述與分析後，我們對問題已有概念上的認知，也完成了系統應具備功能的骨架，再來就是以軟體設計的角度，利用類別圖來設計這些「概念」在實際系統中的定義。

設計類別圖時，便要開始以系統設計的角度考慮軟體實做的相關議題，將原先抽象的概念轉化成系統中一個個的實際類別。

類別圖用來描述系統中物件的型別 (Type)，包括自身的屬性與操作還有各型別之間的關連性。因為在概念塑模中，為了強調抽象的「問題領域敘述」而省略了實作的細節，使得類別圖必定在概念模型的基礎之上增加因應軟體實做而增添的要素，例如物件導向分析與設計中針對物件所強調的封裝、資訊隱藏、繼承等等。

因此我們以概念模型為基礎，配合因應實作設計產生類別圖，來對系統做完整的靜態結構塑模。

3-4-4 建立互動圖

在物件導向系統的開發過程中，完成了物件的靜態資料結構塑模（前述之類別圖）之後，便可進行物件的互動行為塑模來表達物件之間動態的互動行為，就是循序圖與合作圖。透過動態塑模，我們可以了解每個使用個案真實的運作情形和系統在運作時與使用者的互動情形。

1. 建立互動圖：選擇角色：

使用者在參數容器內選擇角色後，參數容器將值回傳給使用者，使用者得知自己所選的角色，說明如圖 31 所示。



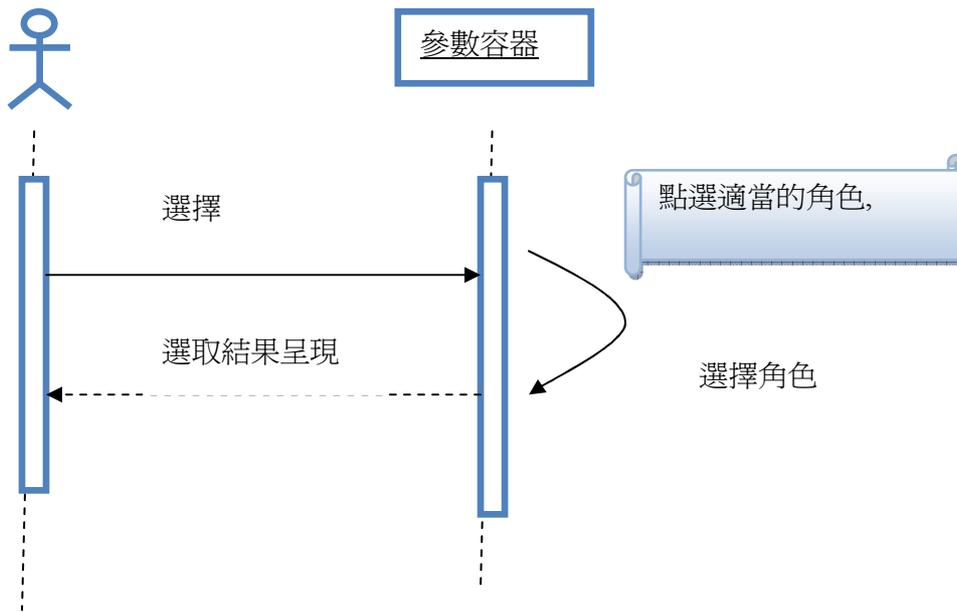


圖 31 角色選擇

2. 建立互動圖：選擇圖表

使用者在參數容器內選擇圖表後，參數容器將值回傳給使用者，使用者可藉由圖表來得知其成本情形，說明如下圖 32 所示。

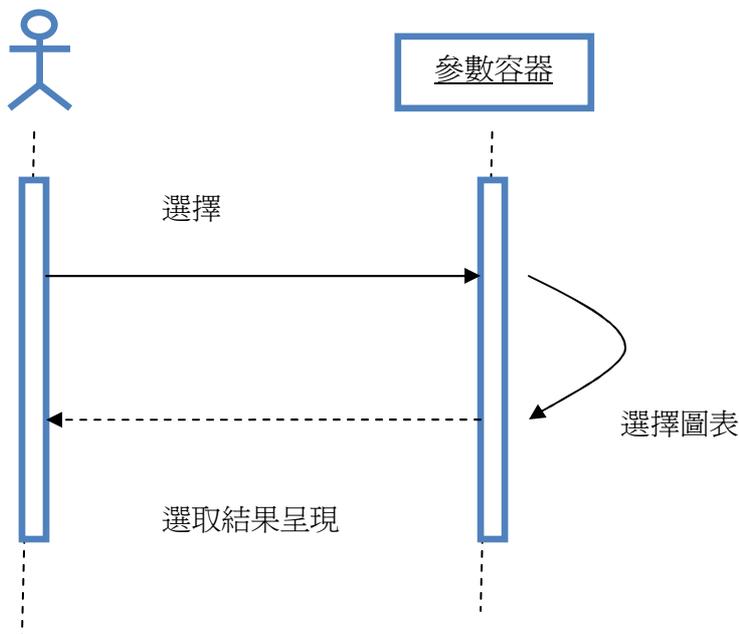
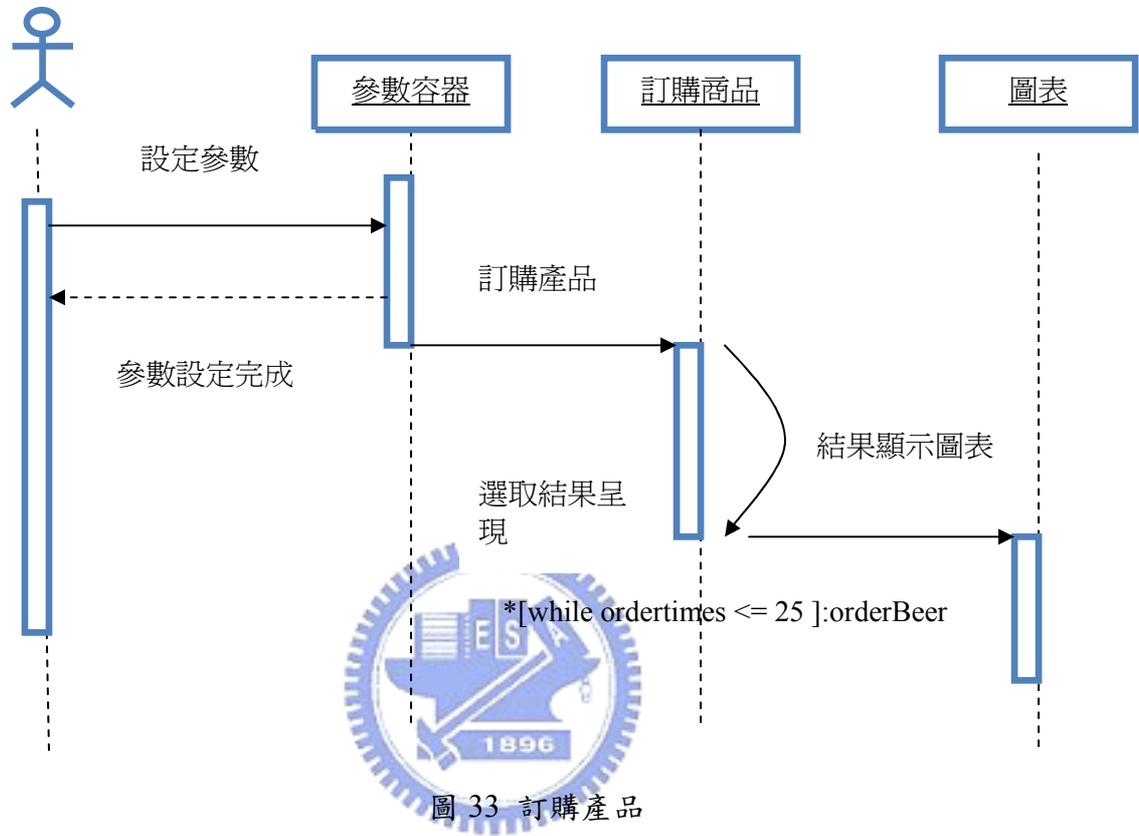


圖 32 選擇圖表

3. 建立互動圖：訂購產品

使用者在設定完參數後，就開始執行遊戲，也就是一連串的連續的訂購啤酒，修正其策略，經過 25 週之後，將結果顯示在圖表上，說明如圖 33 所示。



第四章 系統實作與介面介紹

4-1 系統開發環境

系統設計是決定系統的軟硬體架構，使用的作業系統等，本研究選擇程式開發工具與相關軟體如下：

- (1) Java：使用該語言為唯一開發語言。
- (2) JBuilder X：Borland JBuilder 是一跨平台的 Java 的開發環境，可以符合業界的標準的 Java 應用系統，可應用於開發 EJB、Web、XML 資料庫以及各項應用程式。
- (3) JDK 1.5：使用 SUN 公司所開發的 Java Develop Kit (JDK) 為編譯發展軟體。
- (4) Tomcat：在以 Java 為基礎的 Web 開發領域中，Tomcat 是許多 Java 程式設計師相當喜愛的一項開放原始碼的產品。由於 Tomcat 是 Servlet 與 JSP 技術的參考實作，所以他不但可以當作是獨立的 Web 伺服器，也可以結合其他 Web 伺服器，只擔任 Servlet 容器或是 JSP 容器。



4-2 系統實作

系統實作就是撰寫程式語言，將建立的設計模型轉成原始碼，本研究是實作的步驟如下：

1. 用 JBuilder X 去撰寫成 java 程式，程式結果如下圖 34 所示。
2. 由於完成的是單機版，為了可以在網路上執行此程式，在用 JBuilder X 把原本的 java 程式改成 Applet。
3. 利用 JBuilder X 去部署寫好的 Applet，將之包裝成 JAR 檔案。
4. 部署成功後，把部署好的 JAR 和 html 這兩個檔案複製到 Tomcat 的適當目錄下，使用者就可以從瀏覽器去執行此 Applet 程式了。其網址如下

<http://140.113.71.250:8084/applet/applet/SimpleApplet.html>



圖 34 系統程式畫面

4-3 系統介面介紹

本系統主要介面分成七大部份分別去說明，主要是有：工具列、目前遊戲已進行週數、系統狀態、角色圖示、角色狀態資訊、市場需求、過去歷史資料查詢。

(1) 工具列：

有檔案、選項、執行、圖表以及說明；主要參數設定在選項功能內，且可由玩家自行設定，如圖 35 所示。

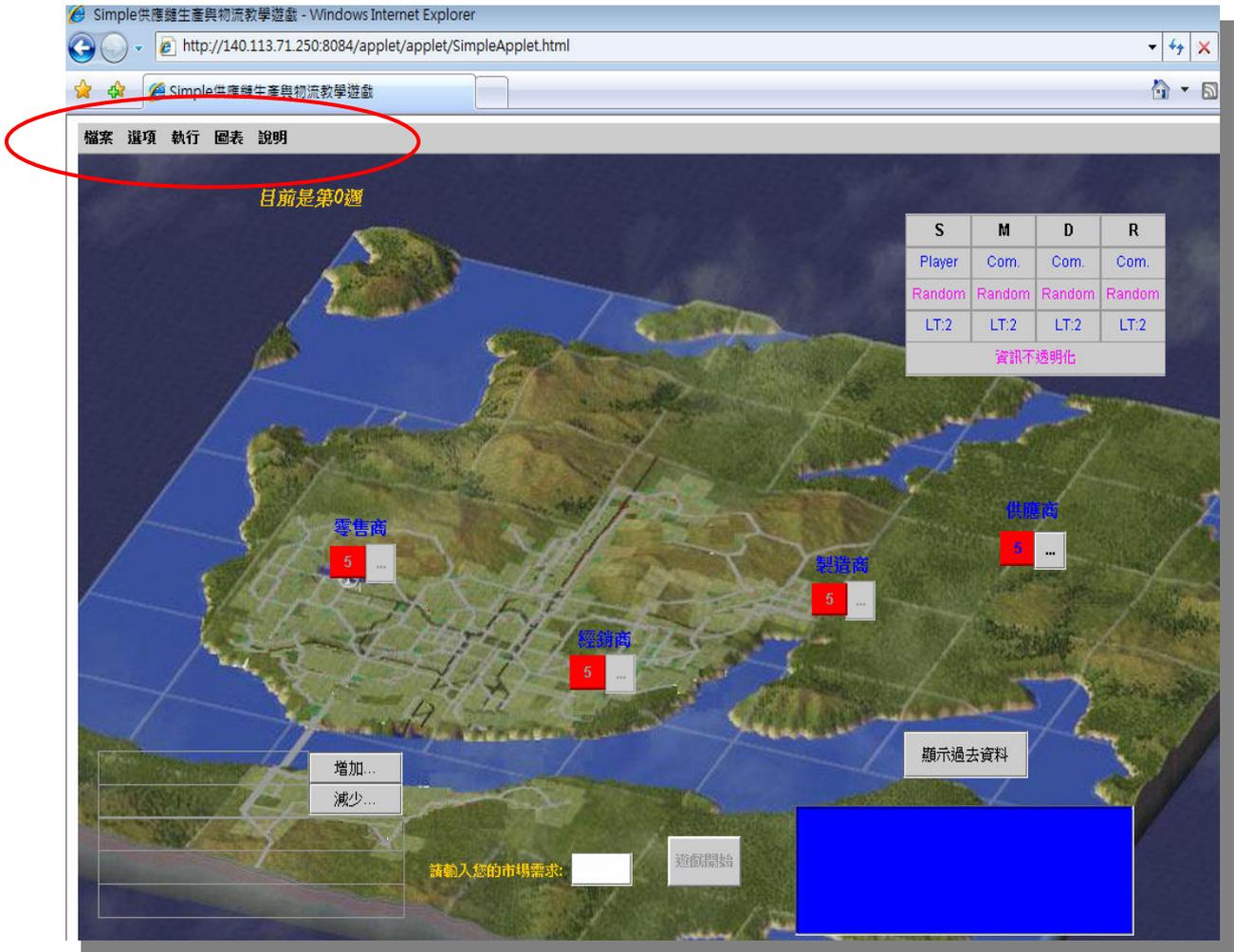


圖 35 遊戲主要介面(工具列)

(2) 目前遊戲已進行週數:

顯示玩家執行遊戲時的週數，遊戲總共進行週數有 25 週，如圖 36 所示。



圖 36 遊戲主要介面(目前遊戲已進行週數)

(3) 遊戲狀態：

如圖 37 所示，在遊戲執行的過程中，玩家和電腦所扮演的角色、前置時間和資訊是否透明化會顯示於此。



圖 37 遊戲主要介面(遊戲狀態)

(4) 角色圖示、角色狀態資訊：

如圖 38 所示，每一個角色都有兩個方格，右邊方格顯示若顯示電腦圖示（沒有反白）則代表是玩家所扮演的角色，左邊方格內之數字代表存貨，預設為 5。點擊右邊電腦圖案後，可在左下角欄位內看見該角色之即時狀態資訊，其中包含目前存貨情形、缺貨產品數量、下一週和下兩週的在途產品的資訊以及最近的訂購量。

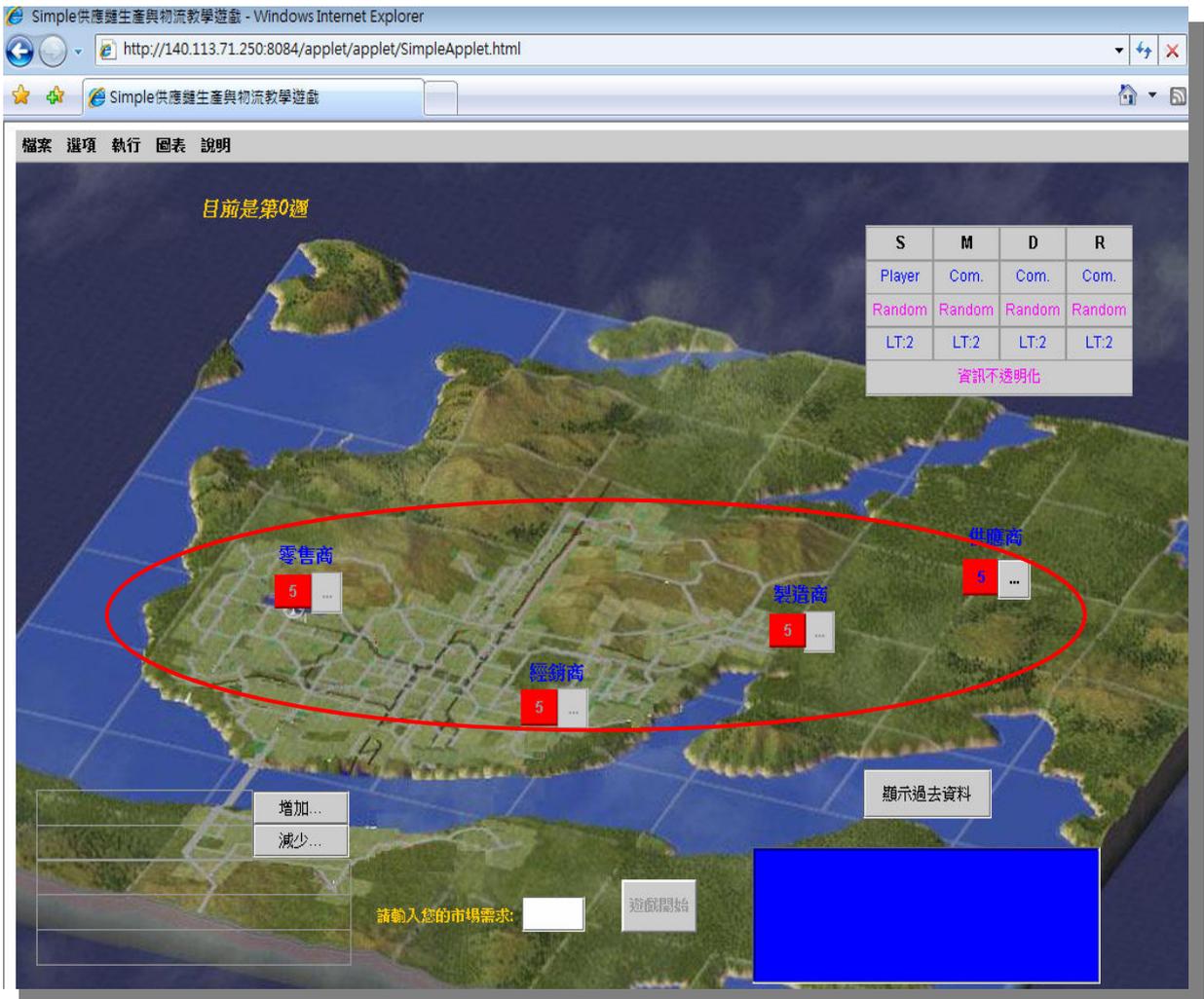


圖 38 遊戲主要介面(角色圖示)

(5) 市場需求：

如圖 39 所示，遊戲進行時會根據一開始設定的需求分配為固定還是隨機而有所不同；若為固定時，每週將保持與最後輸入之需求值相同的市場需求；若為隨機時，每週將以 0 至 10 的均勻分配決定市場需求。



圖 39 遊戲主要介面(市場需求)

(6) 過去歷史資料查詢：

如圖 40 所示，在遊戲進行中，可以觀看玩家所扮演之角色的訂購量以及下游角色的需求量。



圖 40 遊戲主要介面(過去歷史資料查詢)

4-4 工具列項目介紹

工具列的功能，主要有檔案、選項、執行、圖表、說明五個項目。

4-4-1 檔案

檔案按鍵內有離開功能，說明如下：

離開：執行此功能後，會跳出確認視窗以確認是否離開本遊戲。如圖 41 所示。



圖 41 工具列(確認離開視窗)

4-4-2 選項

選項按鍵內有「角色」、「策略」、「倉儲空間」、「集中化管理」、「前置時間」、「需求」、「資訊透明化」與「模組」等八項功能，現分別說明如下：

- (1) 角色：在選項內選擇角色時，會出現角色選擇列表，其中包含『供應商』、『製造商』、『物流中心』、『零售商』四種，如圖 42 所示。

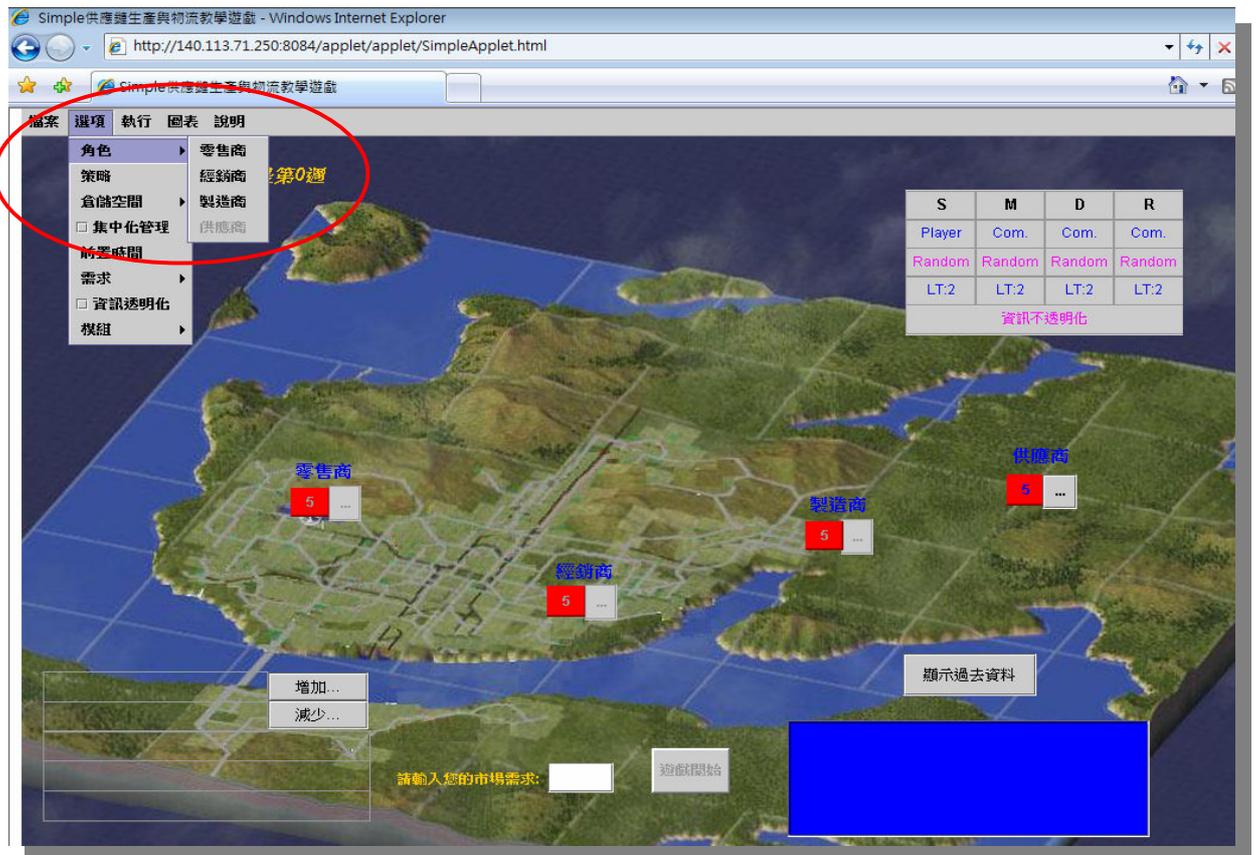


圖 42 工具列(角色)

(2) 策略：選擇策略時，會出現策略選擇視窗，而電腦則代管三種非玩家角色，必須透過此視窗分別選擇個別對應的訂購暨存貨策略，如圖 43 所示。



圖 43 工具列(選項/策略的參數設定)

(3) 集中化管理：可啟動集中化管理模式，其中四個階段之即時狀態資訊均可隨時觀看，如圖 44 所示。

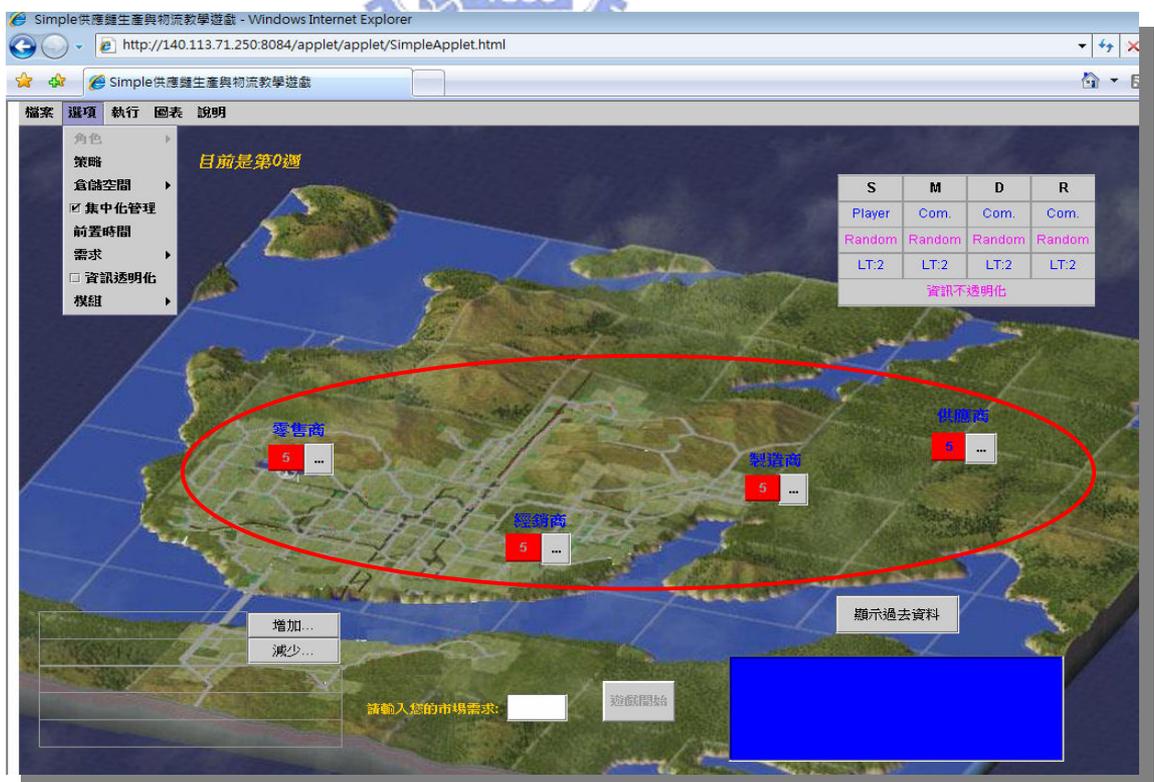


圖 44 工具列(選項/集中化管理)

(4) 前置時間：選擇前置時間的設定後，會出現如圖 45 所示。



前置時間

供應商	製造商	經銷商	零售商
<input type="radio"/> 一週	<input type="radio"/> 一週	<input type="radio"/> 一週	<input type="radio"/> 一週
<input checked="" type="radio"/> 兩週			
<input type="radio"/> 三週	<input type="radio"/> 三週	<input type="radio"/> 三週	<input type="radio"/> 三週
<input type="radio"/> 四周	<input type="radio"/> 四周	<input type="radio"/> 四周	<input type="radio"/> 四周
<input type="radio"/> 五週	<input type="radio"/> 五週	<input type="radio"/> 五週	<input type="radio"/> 五週

確定

圖 45 工具列(選項/前置時間的參數設定)

玩家可以設定本身以及其他角色的產品物流前置時間，起始預設值為兩週，加上資訊流有一週的前置時間，換言之當訂單下達的時候，總共需要三週的時間，貨品才會到達自己的手上。所以玩家可根據自己的倉儲設定以及前置時間來推算自己的訂購量，以避免長鞭效應的產生。

(5) 需求：玩家在點選了需求的選項後，會根據其設定為“隨機”還是“固定”。

(6) 資訊透明化：選擇資訊透明化的選項後，在供應鏈模式下，每一個角色的資訊彼此都會知道，包跨存貨量、缺貨量、訂貨量以及成本情形，如圖 46 所示：

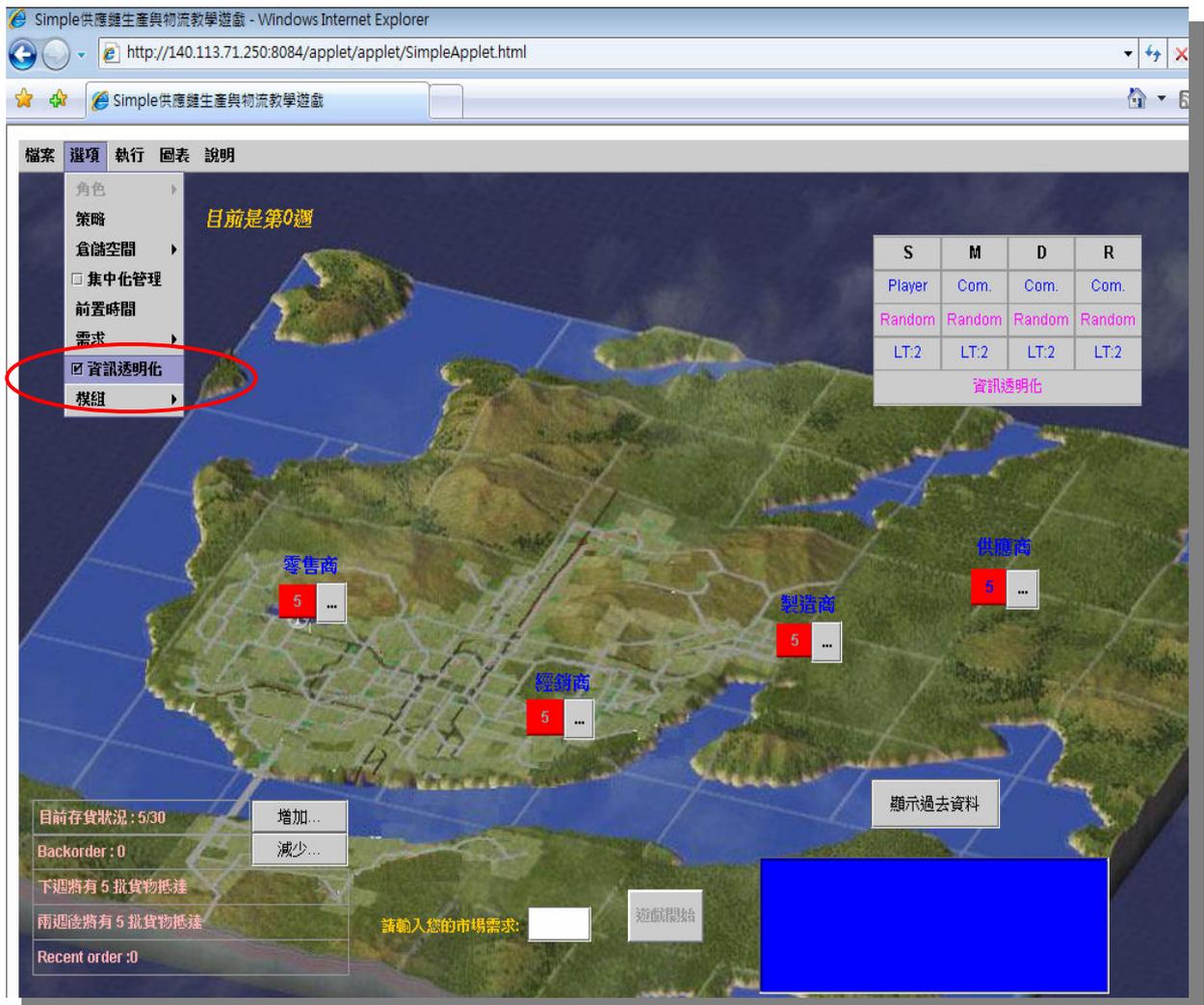


圖 46 工具列(選項/資訊透明化)

(7) 模組：模組的設定讓玩家提供了可以選擇“供應鏈模式”以及“單一模式”，分別說明如下：

1. 供應鏈模式：當玩家選取了這個選項，就進入供應鏈的情境，有零售商、經銷商、製造商、供應商可以選擇。
2. 單一模式：當玩家選取了這個選項，就進入製造商的情境，如圖 47 所示。



圖 47 工具列(選項/模組)

4-4-3 執行

內有「開始遊戲」功能，如圖 48 所示。

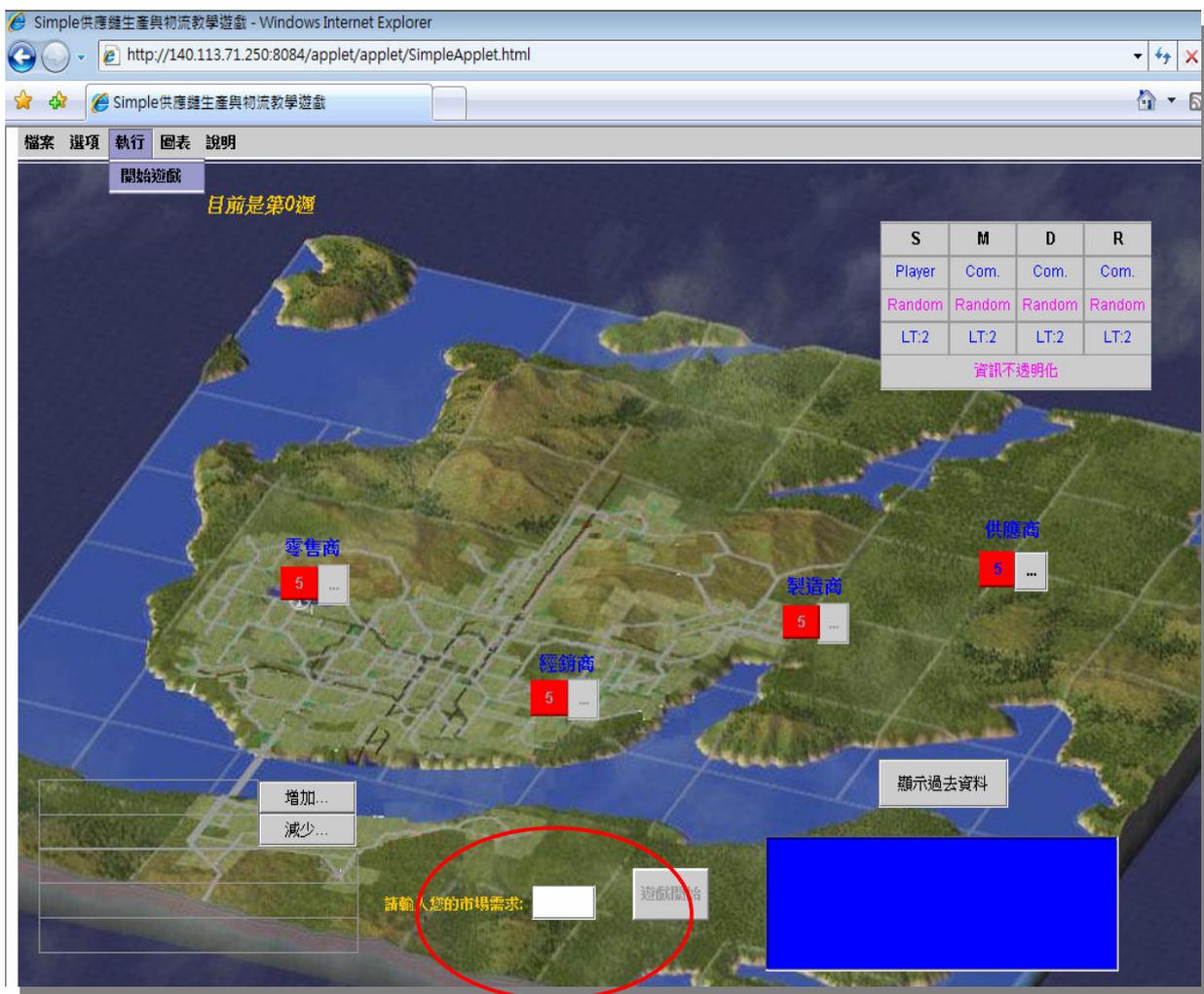


圖 48 工具列(執行/開始遊戲)

當完成遊戲參數設定之後，點擊「開始遊戲」，才得以真正開始遊玩遊戲。

4-4-4 圖表

有「總成本分析圖」、「成本結構分析圖」與「市場趨勢圖」，可以讓玩家更容易地掌握目前狀態的相關數據，也可以讓玩家了解長期之下成本或存貨狀況的趨勢。以下為本遊戲所提供圖表之說明：

1. 總成本分析圖：縱軸代表成本以「元」為單位，橫軸代表時間以「週」為單位，將以線條表示玩家在每一週的總成本，如圖 49 所示。

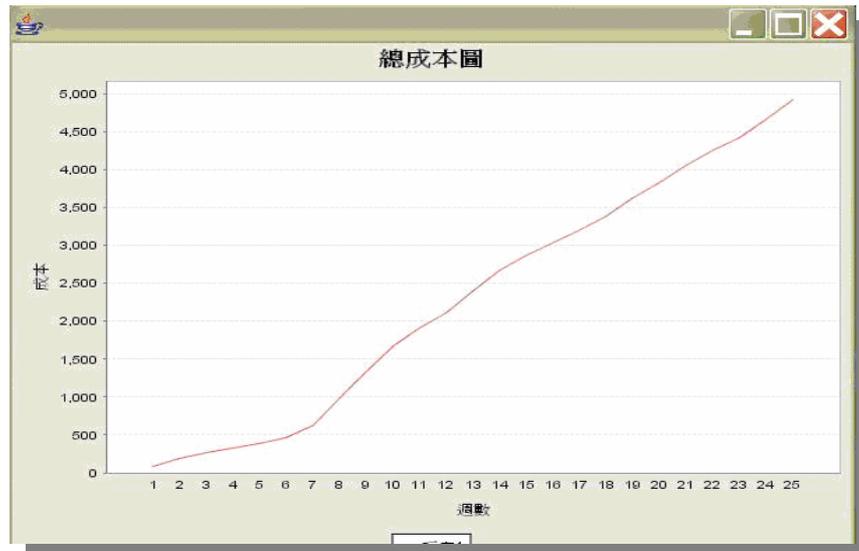


圖 49 工具列(圖表/總成本分析圖)

2. 成本結構分析圖：以圓餅圖(Pie chart)表示，以不同顏色之扇形面積表示玩家總成本內存貨成本、缺貨成本、懲罰成本與訂購成本的組成比例，如圖 50 所示。



圖 50 工具列(圖表/成本結構分析圖)

3. 市場趨勢分析圖：縱軸代表貨品數量以「個數」為單位，橫軸代表時間以「週」為單位，此分析圖利用三種不同顏色的折線來表示玩家向上游角色的訂購量、缺貨數量及下游角色對玩家的訂購數量，如圖 51 所示。

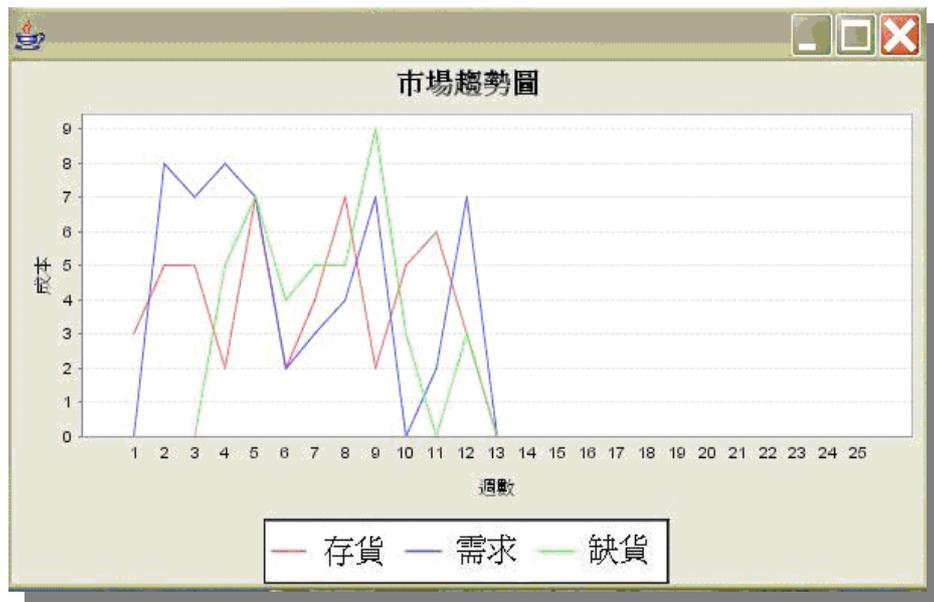


圖 51 工具列(圖表/市場趨勢圖)

當遊戲執行中，玩家僅可獲得該扮演角色之相關統計圖表，而不能觀看其它角色之相關統計圖表；若當遊戲結束，玩家即可閱讀遊戲內所有角色成本結算後之統計圖表。



第五章 結論

5-1 研究成果與貢獻

本研究是利用 Java 和 UML 建構供應鏈的生產與物流教學平台，主要針對目前市面上的供應鏈遊戲無法對於現在越來越複雜的供應鏈作一個有效的教學，所以本研究提供了整合的平台以及多參數的彈性。

本研究成果與貢獻如下：

1. 藉由遊戲的方式，可以讓激起學生興趣而有參與感，達到提高更好的教學效率。
2. 利用 UML，在未來遊戲有配合教學需要重大更改的時候，程式設計師可以藉由 UML 可以快速的了解其架構，參數。
3. 本研究利用的 UML 的圖統計數如表 13。

表 13 統一塑模語言統計圖表

UML 圖	個數
使用案例圖	1
類別圖	4
循序圖	3

本研究並未所有 UML 圖全部繪製出來，主要是因為 UML 的精神乃作為一個讓不同程式背景的人可溝通的平台，所以強調的是溝通泛用性，因此本研究把主要的 UML 圖繪製出來，以避免太多的 UML 圖反而會增加其溝通的困難度。

5-2 未來研究方向

未來研究方向可分成兩方面：

1. 在版本更新方面：本研究目前是單機版本，以後會開發網路版的型態，可以讓多人同時上線參與遊戲，達到相同資源多人競爭與合作的目的，進而思考供應鏈價值。
2. 在教學應用方面：遊戲預期未來可依教學內容的不同，調整相關的參數，驗證該

課程之理論。因此在未來程式撰寫過程同時，亦會利用網路問卷來收集多位相關課程教師及學生意見，完成相關改進，如庫存空間上限及如何使畫面更親和等，進而取代目前較少功能的啤酒遊戲。



參 考 文 獻

中文部分：

- 1.何致億，SCWCD認證專家應考指南，學貫行銷股份有限公司，2005。
- 2.陳會安，Java2 與 UML 物件導向程式範例教本，學貫行銷股份有限公司，2005。
- 3.周信邦，「製造模組控制器軟體設計與實作」，國立交通大學機械工程研究所論文，2001。
- 4.經濟部工業局，「軟體工業五年發展推動計畫」，軟體技術文件指引手冊，2001。
- 5.葉盛昌，「遊戲式數學教學模式對學生數學學習的影響」，台中師範學院教育系論文，2003。
- 6.林調風，「e-Learning 多媒體教材之實作研究」，國立中央大學電機工程研究所論文，2003。
- 7.許瑛珺、廖桂菁，情境式網路輔助學習環境之研發與實踐，科學教育學刊，第十卷第二期，頁 157-178，2002。
- 8.黃雲龍，電子商務情境教學之行動研究，電子商務研究，第二卷第三期，頁 315-332，2004。
- 9.蔡學偉，問題導向學習於網路輔助電腦樂高課程之研究，國立台灣師範大學工業科技教育學系碩士論文，2004。
10. David Simchi-Levi，Philip Kaminsky，Edith Simchi-Levi，供應鏈之設計與管理：觀念 策略 個案，2001。
- 11.吳仁和等，系統分析與設計：理論與實務應用，台北致勝文化，1997。

英文部分：

12. Jason Hunter, and William Crawford. , Java Servlet design , O'REILLY , 1996 .
13. Sharon O'Leary, Lisa Diepenhorst, Ruth Churley-Strom and Diane Magrane, Educational games in an obstetrics and gynecology core curriculum, *American Journal of Obstetrics and Gynecology*, Volume 193, Issue 5, November 2005, Pages 1848-1851.
14. Barry D. Mann, Benjamin M. Eidelson, Steven G. Fukuchi, Steven A. Nissman, Scott

Robertson and Lori Jardines, The development of an interactive game-based tool for learning surgical management algorithms via computer, *The American Journal of Surgery*, Volume 183, Issue 3, March 2002, Pages 305-308.

網站部分:

15. MIT(2005) Retrieved Dec. 6, 2006 from <http://beergame.mit.edu/>

16. Haifa college() <http://hulia.haifa.ac.il/Eng/hulia.html>

17. Michael Wellman (2002) Retrieved Dec. 6, 2006 from TAC game <http://www.sics.se/tac>

18. Sunil Kumar and Samuel C. Wood (2003) Retrieved Dec. 6, 2006 from Littlefield Technology game <http://littlefield.responsive.net>

19. Stijn-Pieter van Houten Global SCM game Retrieved Dec. 13, 2006 from <http://www.gscg.org>

20. Sam Wood SCM game Retrieved Dec. 17, 2006 from <http://sc.responsive.net/SupplyChain/SCAdmin>

21. TU Delft (2005) The Logic game Retrieved Dec. 20, 2006 from <http://www.moltho.dk>

22. IFIP working group Retrieved Dec. 24, 2006 from <http://iproduct.auc.dk/x-proj/gamespm/www-games.html#side2>

