

國立交通大學

資訊學院資訊科技（IT）產業研發碩士專班

碩士論文

棋類教學平台之多點觸控系統

Multi-Touch System for E-Learning Platform of Board Games

研究生：廖家茵

指導教授：吳毅成 教授

中華民國九十九年九月

棋類教學平台之多點觸控系統
Multi-Touch System for E-Learning Platform of Board Games

研究生：廖家茵

Student : Chia-Yin Liao

指導教授：吳毅成

Advisor : I-Chen Wu

國立交通大學

資訊學院資訊科技（IT）產業研發碩士專班

碩士論文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Industrial Technology R & D Master Program on
Computer Science and Engineering

September 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年九月

棋類教學平台之多點觸控系統

研究生：廖家茵

指導教授：吳毅成

國立交通大學資訊學院資訊科技（IT）產業研發碩士專班

摘要

本論文開發出一套適用於電腦棋類教學平台的多點觸控系統。系統內包含以 WPF 開發而成的棋類教學平台和以此平台為主設計而成的棋類遊戲模型，讓符合此棋類遊戲模型的遊戲，都可以在此平台上實作，此棋類遊戲模型以六子棋為主要應用。



Multi-Touch System for E-Learning Platform of Board Games

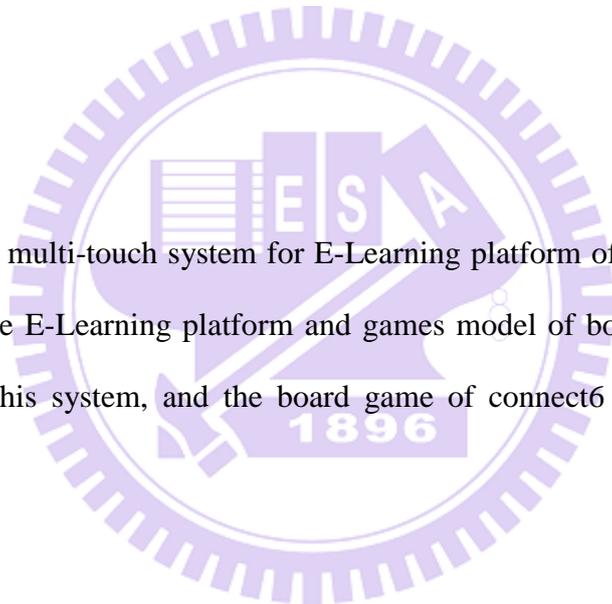
Student : Chia-Yin, Liao

Advisor : I-Chen, Wu

Industrial Technology R & D Master Program on
Computer Science and Engineering
National Chiao Tung University

Abstract

We develop a multi-touch system for E-Learning platform of board games. This system contains the E-Learning platform and games model of board games. We use WPF to develop this system, and the board game of connect6 which is our main application.



誌謝

能夠完成這篇論文，首先感謝我的指導教授吳毅成老師，給予我的協助與意見，另外也感謝所有口試委員所給予的指教，讓這篇論文的內容上更加豐富，並感謝台灣微軟公司及微軟創新研究中心的支持，使我能更加專心的投入研究。

再來感謝實驗室的學長，林秉宏、林宏軒、孫德中學長，在研究的想法與方法上，給予我相當多的意見；接著也感謝我的好夥伴林彥成，與我共同將系統開發完成，而鄒忻芸、陳柏廷則在程式想法上給予我相當多的協助，另外還有陳駿嶧與黃郁雯，兩年的風風雨雨，我們一同度過；還有非常感謝實驗室的學弟韓尚餘、甘崇緯、楊景元、陳昱維、蔡心迪的陪伴與力挺，讓我擁有愉快的碩士生活；非常感謝各位在我的研究期間，不吝嗇給予的協助與幫忙。

最後感謝我的家人，在我研究的期間，給我相當多的鼓勵。謹以此篇論文，獻給所有在我這兩年研究路上，協助及陪伴我的老師、朋友及家人，謝謝你們。

民國九十九年八月 於 新竹交大工程三館 CYC-511 實驗室

目錄

摘要.....	I
ABSTRACT	II
誌謝.....	III
目錄.....	IV
圖表目錄.....	VII
第一章、介紹.....	1
1.1 研究動機.....	1
1.2 研究目的.....	2
1.3 章節介紹.....	2
第二章、情境.....	3
2.1 教學模式.....	3
2.1.1 建議著手.....	4
2.1.2 棋型搜尋.....	4
2.1.3 棋局搜尋.....	6
2.1.4 棋書.....	8
2.2 對戰模式.....	9

第三章、研究背景	11
3.1 多點觸控技術	11
3.2 WINDOWS PRESENTATION FOUNDATION	12
3.2.1 <i>eXtensible Application Markup Language</i>	12
3.3 PLAY-ON-TABLE 遊戲模型	13
3.4 棋類應用	16
3.4.1 六子棋	16
3.4.2 其他棋類	17
3.5 雲端運算系統	18
3.5.1 <i>GridGain</i>	18
3.5.2 <i>Windows Azure</i>	18
第四章、系統介紹	20
4.1 遊戲模型	20
4.2 系統架構	20
4.3 系統實作	21
4.3.1 六子棋系統功能	21
4.3.1.1 教學模式	21
4.3.1.2 對戰模式	25

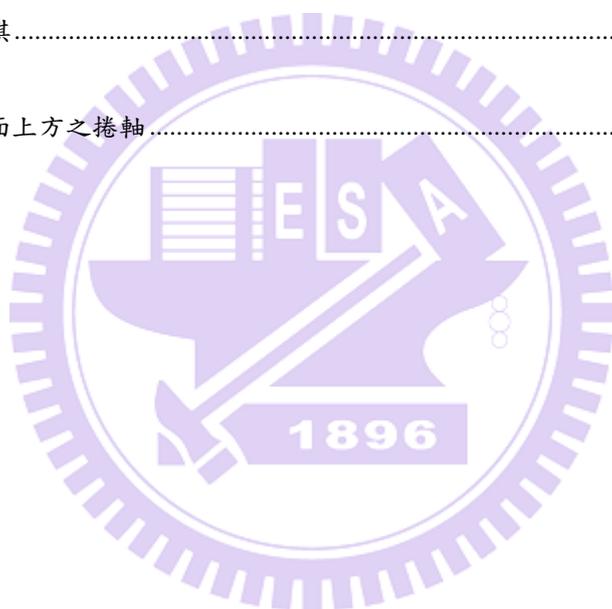
4.3.2 象棋系統功能.....	26
4.3.3 遭遇困難.....	28
第五章、結論.....	29
參考文獻.....	30



圖表目錄

圖 1 教學模式起始畫面	3
圖 2 建議著手介面	4
圖 3 棋型搜尋介面	5
圖 4 棋型搜尋之結果	6
圖 5 棋局搜尋介面	7
圖 6 棋局搜尋之結果	8
圖 7 棋書	9
圖 8 對戰模式之設定	10
圖 9 對戰模式	10
圖 10 技術架構	11
圖 11 .NET Framework 架構圖	12
圖 12 $G=(A, O, S, M)$	14
圖 13 兩狀態間的移動	14
圖 14 撲克牌的移動	15
圖 15 撲克牌的上下切換	15
圖 16 撲克牌的正反面	16
圖 17 六子棋編輯器	17

圖 18 建議著手回傳圖	22
圖 19 棋型搜尋回傳圖	23
圖 20 棋局搜尋回傳圖	24
圖 21 棋書	24
圖 22 棋書樹狀結構	25
圖 23 對戰模式流程圖	26
圖 24 象棋	27
圖 25 介面上方之捲軸	28



第一章、介紹

本系統為實作了一個多點觸控之教學平台，並在平台上設計了一個棋類遊戲模型，除了以六子棋為主要應用的範本外，另外也實作了象棋兩人對奕的部分。本論文分成幾個部分做說明，分別是介紹、情境、背景和系統。介紹裡會說明動機、目的介紹，情境裡以六子棋的情境為主，背景裡會簡介本系統會應用到的技術和理論，最後系統的部分會說明棋類遊戲模型的設計理論、系統架構、系統功能介紹和開發時所遭遇的困難。

本章節會說明本論文的研究動機及研究目的，另外針對系統的兩個模式，以其內的功能為主，介紹系統會使用到的情境。

1.1 研究動機

現今已有許多棋類的系統可以輔助棋類教學，例如編輯器和對弈系統。編輯器可以編譯棋譜、建議下一個著手，如 LGC 傳奇網路圍棋軟體、六子棋編輯器和象棋編輯器等，另外像傳奇網路圍棋廳[6]則提供圍棋線上對弈、CYC 遊戲大聯盟[2]裡則有多種棋類的線上對弈系統，這都屬於對弈系統的一種。

上述所說皆為以滑鼠和鍵盤為主，多點觸控是屬於比較少見的操作方式，因此本系統希望能開發一個棋類教學平台之多點觸控系統，因為如棋盤的旋轉、縮放和移動都適合應用在多點觸控上，而六子棋又有一次下兩顆子的特性，配合多點觸控，可作為棋類教學平台系統的一個良好的範本。

另外配合與微軟合作的「六子棋及相近棋類遊戲之雲端教學及運算平台」計

畫，將此計畫的前端做為多點觸控的系統，重新設計符合多點觸控的遊戲介面，與後端的雲端系統做整合，使玩家有較為直覺和親和的使用經驗。

1.2 研究目的

根據研究動機，本論文的研究目的是開發適用於棋類教學平台之多點觸控系統，系統的功能以現有棋類編輯器的功能為主，因為本論文以六子棋為主要應用，所以參考的是六子棋編輯器，本系統的操作為觸控方式，應用多點觸控的技術讓玩家有較為直覺與親和的使用經驗，另外在本平台上會設計一個棋類遊戲模型，讓符合這個模型的棋類遊戲，都可以在本系統上實作，例如：六子棋、象棋、圍棋和五子棋等，本論文以六子棋為主要應用，象棋的部分則實作了兩人對奕的部分。

1.3 章節介紹

本篇論文第二章是說明系統情境，以六子棋為主要的說明範例，第三章是研究背景，其中會介紹本篇論文使用的相關技術，及相關研究，包括多點觸控、POT 遊戲模型、Windows Presentation Foundation 和 Azure 等，第四章是系統介紹，說明本系統的實際做法和遭遇的困難，第五章就描述結論及未來展望。

第二章、情境

本章節的情境以六子棋為主要的應用說明，在本系統的六子棋主要分成兩個模式：教學模式和對戰模式。教學模式顧名思義就是讓玩家學習如何下六子棋及增加下棋的功力，對戰模式則是讓黑白雙方可以對弈的棋類對弈系統。

2.1 教學模式

圖 1 是教學模式的起始畫面，教學模式的所有功能都散佈在畫面中的物件上，例如左下角的錦囊、下方的摺扇、右下方的書本、上方的捲軸都是功能之一。



圖 1 教學模式起始畫面

2.1.1 建議著手

這個功能在畫面中的顯示鈕為左下角的紅色錦囊，當玩家不知道下一步該如何下時，可按建議著手的錦囊，將目前的盤面傳送到後端作運算，等後端的電腦 AI 運算過後會回傳建議的著手，本系統設定須回傳三個著手，如圖 2。



圖 2 建議著手介面

2.1.2 棋型搜尋

在捲軸上方左右兩側有兩個功能分別為棋型搜尋和棋局搜尋，棋型搜尋用在當玩家想知道自己盤面中的某個棋型是否其他棋譜也有相同的棋型時，即可使用這個功能。

按下棋型搜尋的顯示鈕可直接利用觸控技術在螢幕上做圈選的動作，會將圈選過的地方用淺藍色的方塊標示並填滿其內的區域，如圖 3。



圖 3 棋型搜尋介面

按下確認鍵後，會將圈選的棋型(包含空點)傳送到後端，若想要重新圈選棋型，可以按取消鍵，畫面中淺藍色的區域就會消失，即可重新做圈選的動作，將圈選的棋型傳送到後端後，等電腦 AI 比對後會回傳符合的棋譜，再將回傳的結果盤面一一會顯示在上方的捲軸中，如圖 4。



圖 4 棋型搜尋之結果

2.1.3 棋局搜尋

上方卷軸的左方為棋局搜尋，這部分的功能是用在當玩家想要查詢某個棋士所下過並儲存在資料庫中的所有棋譜時，可以在上方的捲軸內輸入棋士的暱稱 (ID)，並按下查詢，如圖 5。



圖 5 棋局搜尋介面

當後端回傳符合此 ID 的盤面後，將其結果盤面一一顯示在上方的捲軸上，若點選捲軸中的小棋盤即會在下方出現同一棋譜的大棋盤，可讓玩家更清楚盤面上的棋譜並可按摺扇上的前後步來操作，觀看上下手之間的關係，如圖 6。另外兩個大棋盤皆可以移動並縮放，方便玩家比較。



圖 6 棋局搜尋之結果

2.1.4 棋書

在畫面中右下角的書本圖標就是進入棋書的功能鍵，棋書主要的目的是讓玩家讀取棋譜並瀏覽，如圖 7。



圖 7 棋書

2.2 對戰模式

這是一個使用多點觸控的對弈系統，進入對戰模式前需要先設定時間和角色，如圖 8。圖中的時間指得是各玩家所能花費的總時間，下方的角色可讓黑白方各自選擇要讓玩家本人或是電腦 AI 下子，選擇好按確認即可進入對戰模式的介面，如圖 9。

選擇電腦 AI 下子的話，畫面中會出現機器人的圖案，代表電腦 AI，一開始黑方計時器會開始倒數，等到玩家下子後黑方計時器才停止，換白方計時器開始倒數，若想要中途投降，可以按下方的白旗，讓對方獲勝。



圖 8 對戰模式之設定

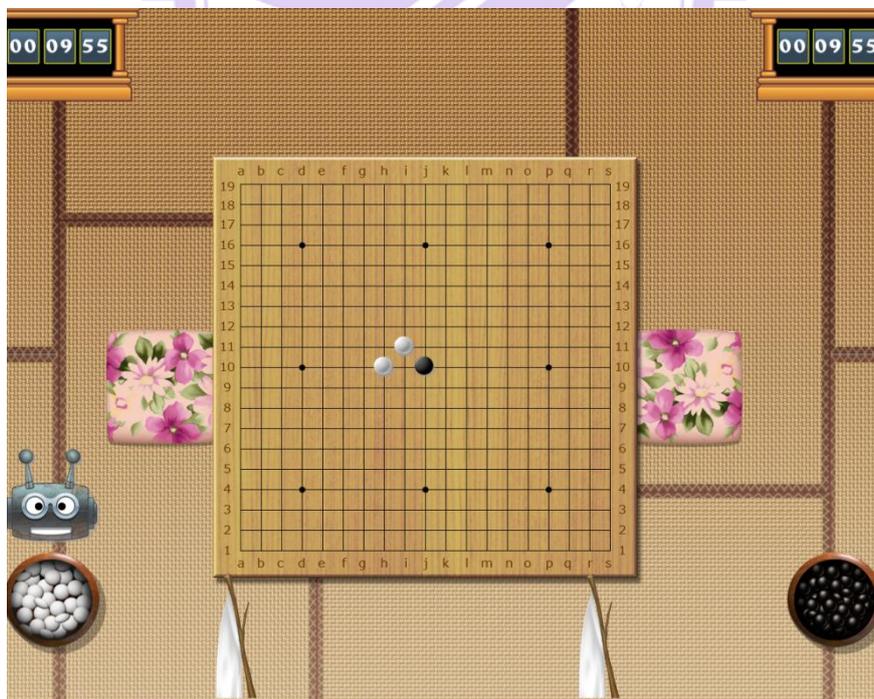


圖 9 對戰模式

第三章、研究背景

本章介紹本篇論文會使用到的技術與理論，如圖 10，分成前後端技術、嵌入遊戲之設計理論和棋類應用四個部。前端技術的部分配合系統的架構，由於本系統是由 Windows Presentation Foundation 開發而成的多點觸控平台，所以主要介紹多點觸控技術和 Windows Presentation Foundation(簡稱 WPF)，而在平台上的則是依據 play-on-table(簡稱 POT)遊戲模型設計而成的棋類遊戲模型，而其模型應用到的棋類主要是六子棋，另外圍棋、象棋和五子棋也是很好的應用，與後端連結的則是雲端系統，根據以上所述，將一一介紹。



圖 10 技術架構

3.1 多點觸控技術

關於多點觸控可追溯到 1982 年，多倫多大學發明了多點觸控螢幕，而同年貝爾實驗室也發表了第一篇跟觸控技術有關的學術文獻，但是人們還是對多點觸

控技術沒有很深的概念，到了 2006 年 Jeff Han 在 TED 大會展示了多點觸控上的創新應用[3]，大家才較了解多點觸控可應用在哪些方面，隔年 2007 年蘋果公司也發表 iPhone，而微軟則發表了 Surface Computing，這些皆為多點觸控的產品。

3.2 Windows Presentation Foundation

Windows Presentation Foundation(簡稱 WPF)為微軟.NET Framework 的組成之一[13]，也被微軟喻為下一代使用者開發介面技術，是在.NET Framework 3.0 之後才有的，如圖 11。

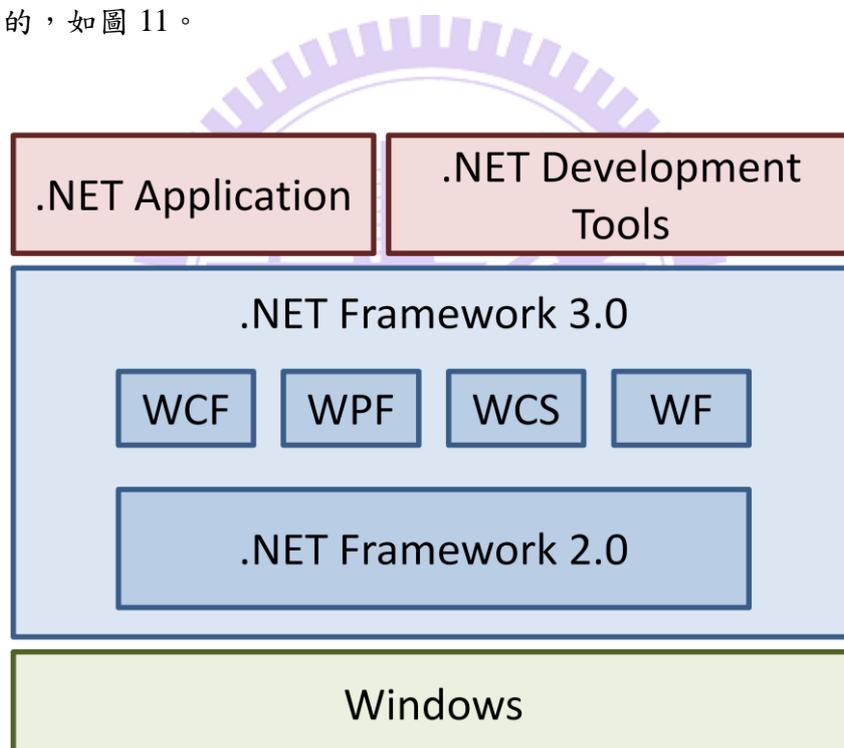


圖 11 .NET Framework 架構圖

3.2.1 eXtensible Application Markup Language

eXtensible Application Markup Language 縮寫為 XAML，念法為「Zammel」，

是一種讓 WPF 用來定義介面的標記語言，主要用在介面的開發上，可將前台的介面開發和後台的邏輯程式分開。

3.3 Play-on-table 遊戲模型

Play-on-table 遊戲模型是根據吳毅成教授和徐建智學長在 2004 年發表的一篇文章[15]，裡面將 Play-on-table(POT)遊戲用正規的方法表現： $G = (A, O, S, M)$ ， G 代表的就是 POT 遊戲。

- A 指得是區域，也就是一個二維的座標。
- O 指得是物件， $O = \{O_1, O_2, O_3, \dots, O_n\}$ ，而每一個 O_i 都有 m_i 個 face Φ_i ，而 $\Phi_i = \{\Phi_{i1}, \Phi_{i2}, \dots, \Phi_{i,mi}\}$ ，例如撲克牌 A 有 2 個 face{正面, 反面}，本論文定義 $O = A \times Z \times \Phi_i$ ，代表物件在遊戲中所表示的狀態，例如圖 12 中的撲克牌，設定撲克牌正面 Φ_1 ，反面為 Φ_2 ，所以撲克牌 A 目前的狀態為 $O1 = (X1, Y1) \times 1 \times \Phi_1$ ，而我們用 $\Gamma(A, O)$ 來代表所有可能發生的遊戲狀態。
- S 指得是狀態，也就是一個 POT 遊戲狀態的集合，每個遊戲都包含開始與結束兩種狀態，用正規化的方式來表示，可得 $S = \{(start), (end)\} \cup S^-$ ， S^- 也就是 $\Gamma(A, O)$ 的子集合。
- M 指得是操作，也可以說是遊戲的規則，例如把遊戲中的 S_1 變成 S_2 就是操作的一種，如圖 13，所以 M 是 $S \times S$ 的子集合，而在那篇論文裡定義了三種操作模式。

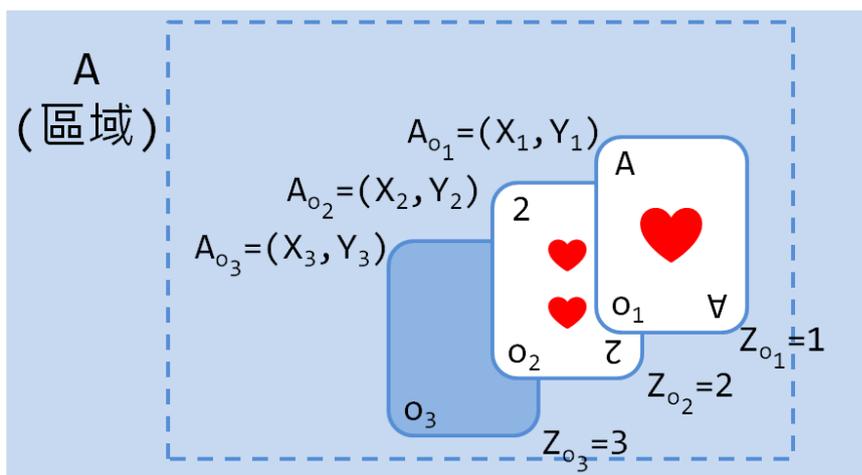


圖 12 $G=(A, O, S, M)$

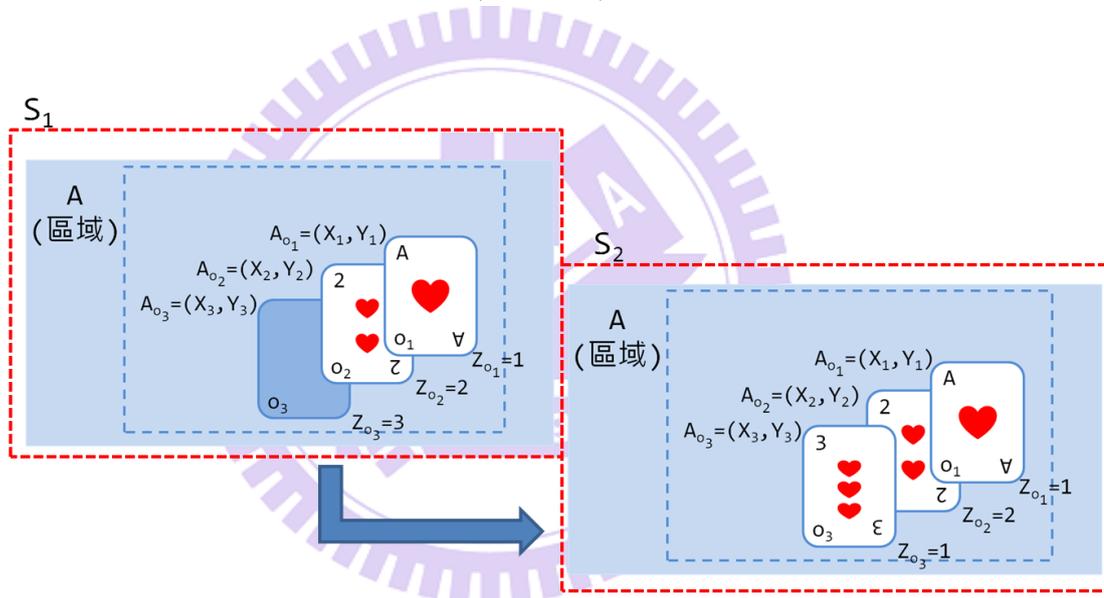


圖 13 兩狀態間的移動

POT 遊戲面所提到了三種操作模式，分別為 Move-Object, Change-Z, Change-face 這三種，只要一個系統可以提供以上三種操作模式，並可定義 A 和 O，這樣的遊戲屬於 POT 遊戲，可用 POT 遊戲模型來模擬。以下將一一介紹這三種操作模式：

- Move-object：這種操作模式指得是改變物件的位置，就是 X、Y 軸的移動，

如圖 14。

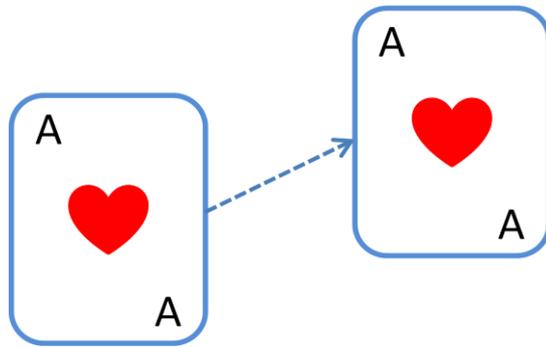


圖 14 撲克牌的移動

- Change-Z: 指得是改變交疊物件得上下位置，也就是 Z 軸的順序，如圖 15。



圖 15 撲克牌的上下切換

- Change-face: 指得就是改變物件的面相，例如將撲克牌的正面翻到背面，就是一種 Change-face，如圖 16。

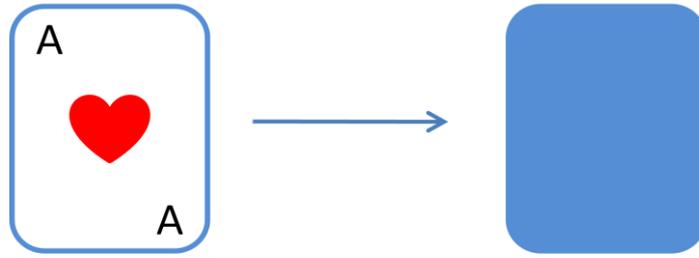


圖 16 撲克牌的正反面

3.4 棋類應用

本系統之應用以符合 POT 遊戲模型中的棋類遊戲為主，六子棋為主要的應用，其他像象棋、圍棋、五子棋也是很好的應用之一。

3.4.1 六子棋

六子棋為吳毅成教授在 2005 年發表，規則為黑方先下一子，其餘雙方各輪流下兩子，先連成六子的獲勝，特色是規則簡單、變化複雜、遊戲公平，打破了五子棋先手容易獲勝的特性，而吳毅成教授及其實驗室也研發一支六子棋程式，名稱為交大六號(NCTU 6)[7][16][17][19][20]，於第 11 屆及第 13 屆國際奧林匹亞電腦賽局競賽獲得冠軍[5]。另外其實驗室也開發了六子棋編輯器，如圖 17，主要用來提供交大六號功能和編輯六子棋棋譜所用。

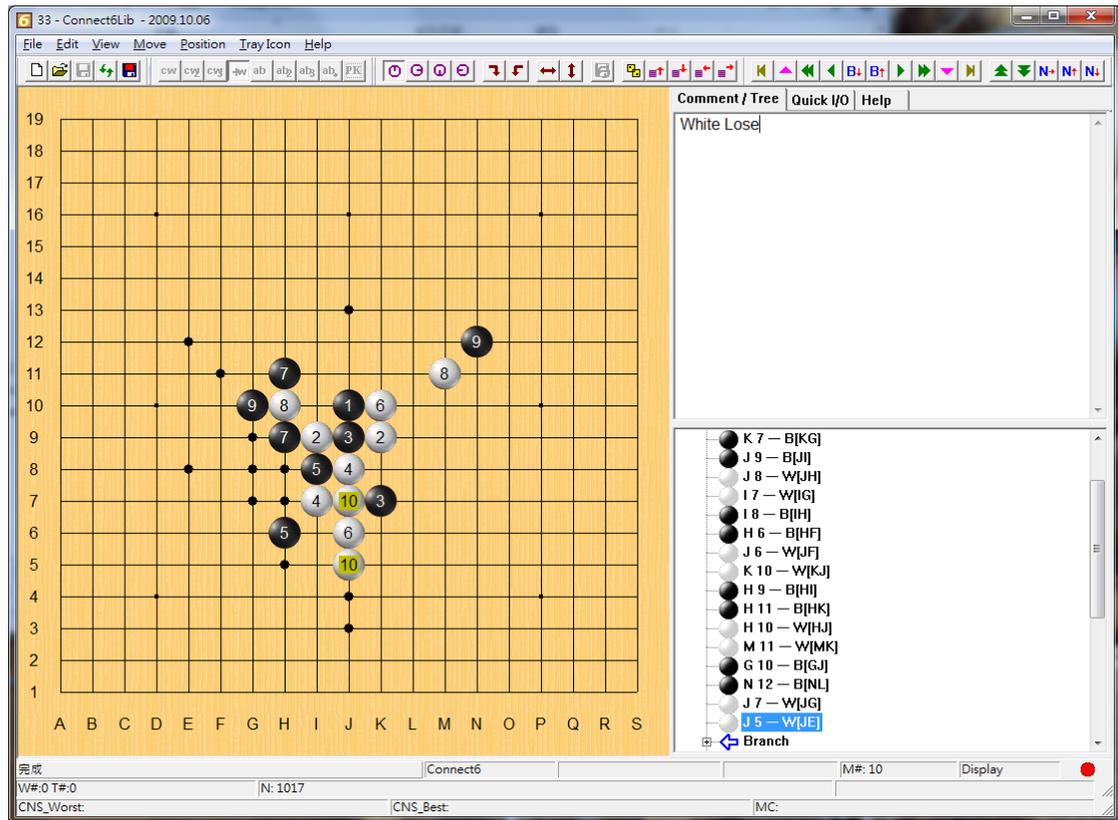


圖 17 六子棋編輯器

3.4.2 其他棋類

其他的棋類，例如：圍棋、象棋、五子棋等，都符合前面所述之 POT 遊戲模型，所以重點式的簡介一下這些棋類遊戲。

- 圍棋：約起源於公元前六世紀的中國古代，是一種爭地的遊戲，誰得地多誰就贏，目前遍佈全世界，以中國、香港、台灣、韓國和日本較為興盛。
- 象棋：中國的戰國時代就有關於象棋的記載，象棋的棋盤是由九條直線和十條橫線相交組成，棋子一開始是放在固定位置的交點上，中間會有一行沒有畫直線稱為楚河漢界[10]，現今約有兩億多的人口會下象棋。

- 五子棋：流傳於中國古老的棋藝之一，規則為先連成五子的獲勝，在日本發展成連珠棋，被廣為推廣。

3.5 雲端運算系統

雲端運算系統可說是透過網際網路以大量的伺服器主機和分散式空間，提供使用者運算和儲存的資源，而前端系統(本論文)會與後端的雲端運算系統做連結，透過雲端運算系統，計算出最好得著手等。1993 年昇陽電腦提出了「The network is the computer」，2006 年三月，Amazon 更提出了 Elastic Compute Cloud (EC2)，即是一種雲端服務，同年的八月，Google 正式提出「Cloud Computing」的概念，之後就陸續出現很多和雲端的學術計畫[9]。建立雲端系統的平台有很多種，例如：Hadoop、GridGain 和 Azure 等，本系統連結的雲端運算系統以 GridGain 和 Windows Azure 為主建立而成的。

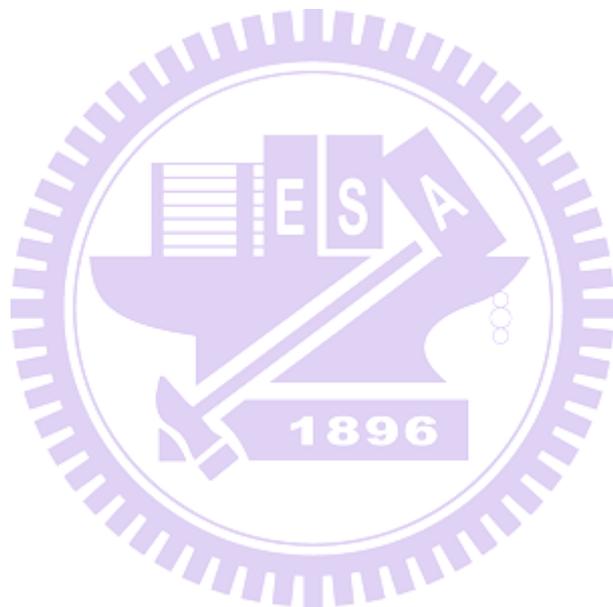
3.5.1 GridGain

GridGain 是以 JAVA 寫成的雲端平台，用一個公式來形容得話，可得 $\text{GridGain} = \text{Compute} + \text{Data} + \text{Cloud}$ 。換句換說就是結合了雲端計算和 Data Grid 的技術，支援 Java 和 Scala programming 語言[4]。

3.5.2 Windows Azure

Windows Azure 為微軟所提供的一個網際網路層級的雲端服務平台[14]，可將利用 Azure 開發的應用程式放在微軟機房裡，藉此省去使用者對於機器管理的成本。

Windows Azure 提供兩個雲端服務資源，分別為運算資源和儲存資源，而運算資源又分成 web role instance 和 worker role instance 兩種，本系統之後端系統主要運用到的部分為 worker role instance，由林彥成同學負責實作[8]。



第四章、系統介紹

本章主要分成三個部分來介紹，依序為遊戲模型、系統架構和系統實作，遊戲模型根據之前的 POT 遊戲模型理論而來，而系統架構裡則說明系統中的功能，最後系統實作的部分則介紹如何實作本系統和所遭遇的困難。

4.1 遊戲模型

本系統根據 POT 遊戲模型理論，設計了一個棋類遊戲模型，讓符合此遊戲模型的棋類遊戲都可在本平台上實作。而針對 POT 遊戲模型所提到的三種操作模式，本研究主要運用了 Move-Object 的操作模式，像六子棋、五子棋、圍棋都是屬於下子的方式，象棋則是移動子的方式，不考慮另外兩種 Change-Z 和 Change-face 的原因，因為棋類遊戲大多不會疊子，且幾乎都是完美訊息(perfect information)，所以本遊戲模型以 Move-Object 為主要操作模式，所以本棋類遊戲模型可以說是 POT 遊戲模型得子集合。

4.2 系統架構

本系統運用 WPF 開發棋類教學平台，平台上可嵌入適合棋類遊戲模型的遊戲，以六子棋為主要的應用，象棋則呈現對弈的部分，另外像圍棋、五子棋等也適用。系統是由 C# 撰寫，利用 .NET Framework 4.0 裡的 WPF 技術，結合 windows 7 的多點觸控支援技術實作而成。

4.3 系統實作

本章分成 3.3.1 系統功能和 3.3.2 遭遇困難兩小節來做介紹。

4.3.1 六子棋系統功能

本節說明六子棋系統的功能，主要分成教學模式和對戰模式兩種來介紹。

4.3.1.1 教學模式

六子棋的教學模式裡的功能主要有建議著手、棋型搜尋、棋局搜尋、棋書這四項。以下將一一介紹。

建議著手

教學模式裡的第一個功能就是建議著手，將欲建議的盤面傳送到後端，雲端系統運算完會回傳最佳著手，目前是設定回傳三個最佳著手，棋士可依造這三個建議著手選擇其一當作下一步的建議，如圖 18。

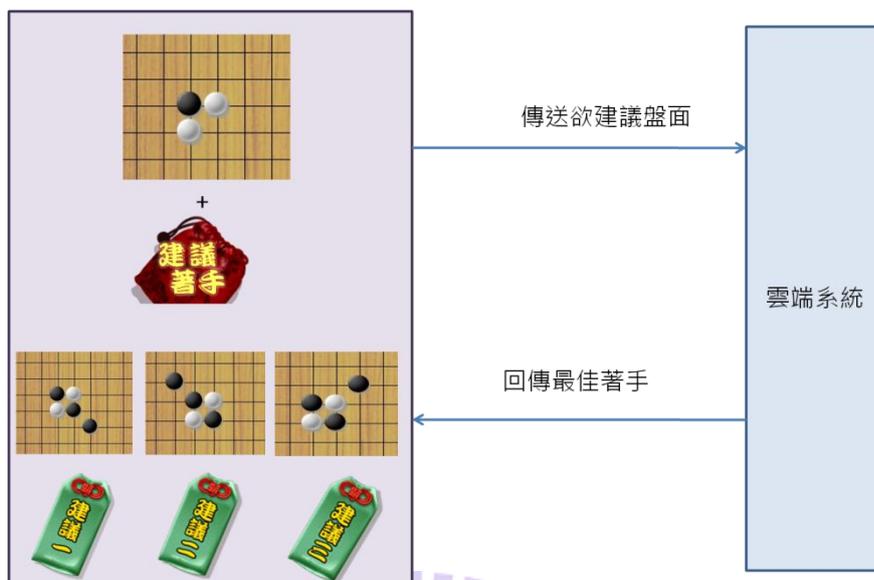


圖 18 建議著手回傳圖

實作的部分就是將欲建議的盤面轉成 SGF 棋譜格式，送到後端做 AI 運算，接收回傳結果後，將結果盤面存起來，如果是第一個結果，則直接顯示在畫面中，並根據結果顯示勝率，因為要顯示三個最佳著手，所以以上動作會在重複兩次。

棋型搜尋

將想要搜尋的盤面傳送到後端的雲端系統，經 AI 運算後再回傳符合的盤面，將其顯示在上方的捲軸裡，如圖 19。

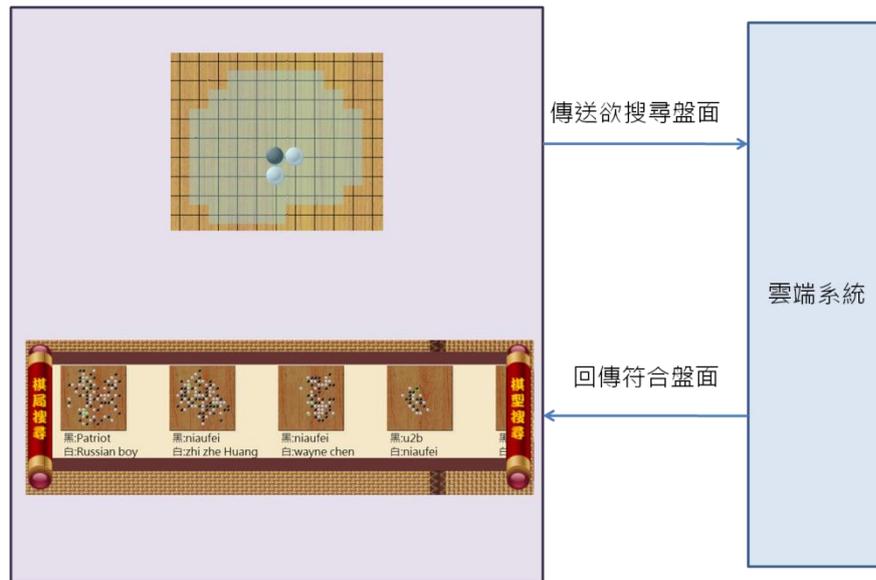


圖 19 棋型搜尋回傳圖

實作的部分就是將圈選出來的盤面解析成空點、黑子和白子，將其送到後端系統做搜尋，接收符合比對的盤面並存起來，再將裡的盤面一一顯示在畫面上。

棋局搜尋

棋局搜尋是將查詢的 ID 傳送到後端，雲端系統會傳符合此搜尋 ID 的盤面顯示在上方捲軸裡，如圖 20。



圖 20 棋局搜尋回傳圖

實作的方法就是將查詢的 ID 傳到後端與資料庫的 ID 做比對，將回傳回來相同 ID 的棋譜存起來，再依序將結果顯示在畫面中。

棋書

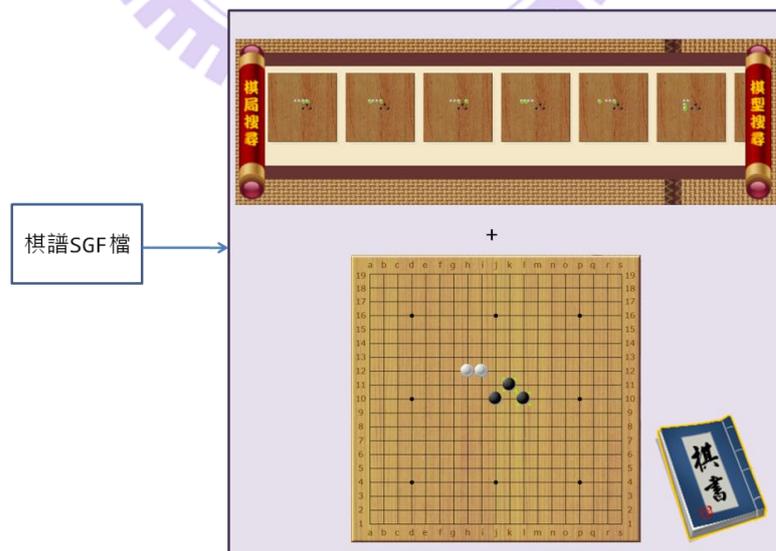


圖 21 棋書

教學模式的最後一個功能就是棋書，讀進棋譜的 SGF 檔，再將結果顯示在畫面中，如圖 21。

由於棋譜的 SGF 檔裡面包含樹狀結構，那如何將樹狀結構顯示在畫面中可由得知，大棋盤顯示根節點，上方捲軸裡的小棋盤顯示子節點，由此一層一層的顯示，如圖 22。

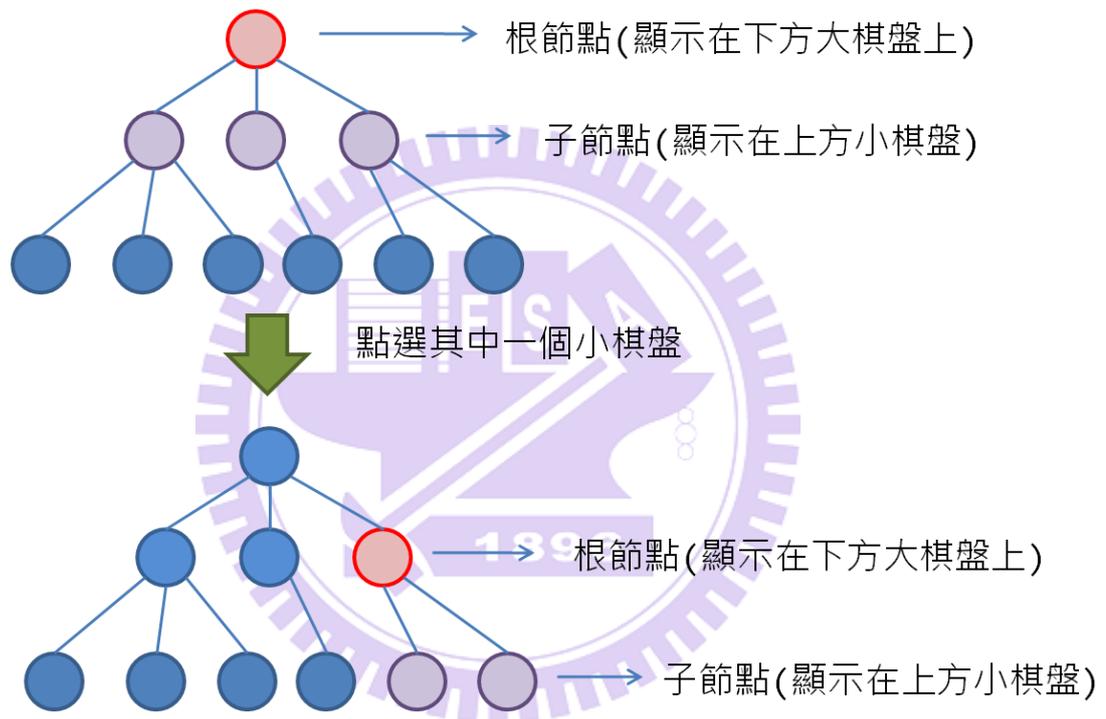


圖 22 棋書樹狀結構

4.3.1.2 對戰模式

對戰模式的流程為先判斷黑白雙方是否有機器人之後，黑方計時器開始倒數，等到黑方下子之後，判斷始否有無獲勝，若無獲勝，再黑方計時器停止倒數，白方計時器開始倒數，之後白方開始思考，等到白方下子後，依然判斷是否有無獲

勝，若無則白方計時器停止倒數，黑方計時器開始倒數，一直循環，如果超時對方就獲勝，如圖 23。

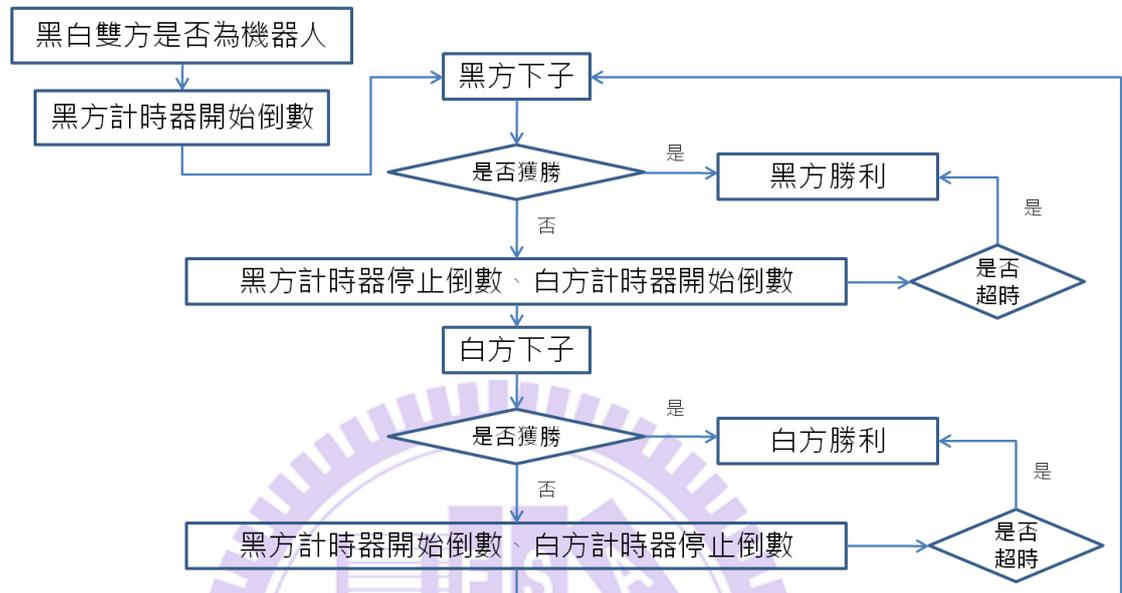


圖 23 對戰模式流程圖

4.3.2 象棋系統功能

象棋則實作兩人對奕的部分，如圖 23，象棋跟六子棋最大的不同點在於下棋的方式，六子棋屬於下子的方式，棋子會隨著下子而增多，而象棋屬於移動棋子的方式，盤面初始時，所有的棋子都已出現在棋盤上，玩家可按住棋子並移動它，所以判斷手指觸控點的座標並可以移動手下的棋子，變成首先要實作的部分，棋子可以移動後，就開始確定棋盤格線上交叉點的座標，讓棋子的移動點都準確並自動的落在棋盤交叉點上，自動校正移動點可讓盤面看起來較為整齊，也可避免電腦因座標點的不同而誤判，再來就是判斷該點座標上是否已有棋子，也就是重複點的判斷，用以達成吃子的目的。

由於象棋是實作兩人對奕系統，沒有電腦 AI 的部分，最後加上判斷輸贏的規則，即完成了，也可說明，若為符合本論文之棋類遊戲模型的棋類遊戲，在本教學平台上實作，可輕鬆的達成。



圖 24 象棋

4.3.3 遭遇困難

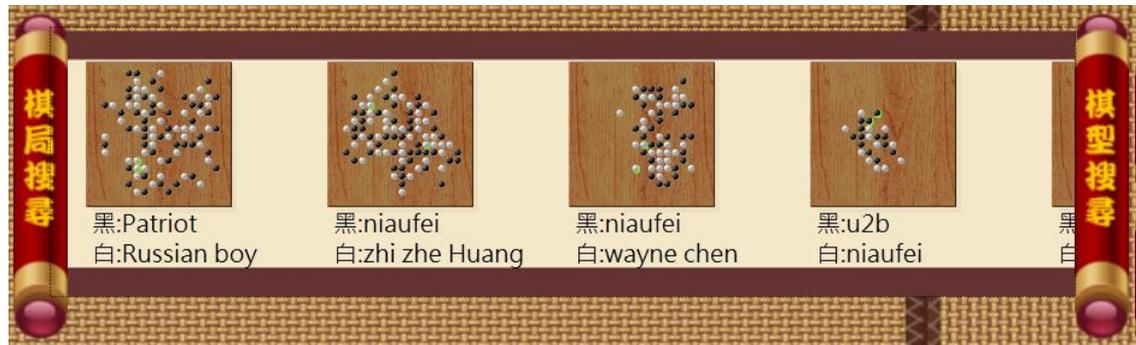


圖 25 介面上方之捲軸

系統開發時遭遇到的問題主要有兩個，這兩個都跟教學模式上方的顯示捲軸有關，如圖 25。第一個問題是搜尋結果過多，有時候甚至多達兩三百個，那系統繪製每個盤面的時間就會過長，等到全部繪製完畢才顯示，會造成時間上的差距，解決的方法就是先將前十個盤面繪出並顯示，其餘盤面在背景繼續繪製，第二個問題就是移動捲軸中的小棋盤時，會閃爍和晃動，因為一個棋譜上的物件超過一個，一起移動時容易有時間差的問題，解決的方法是將系統繪製出的盤面(包含棋盤、黑白子)存成一個圖檔。

第五章、結論

本論文實作了一套棋類教學平台系統，在平台上設計了根據 POT 遊戲模型所修改而來的棋類遊戲模型，也就是使用了 POT 遊戲模型裡 Move-object 的操作模式之遊戲，所以本棋類遊戲模型可說是 POT 遊戲模型之子集合，以六子棋為本系統主要應用，包含教學模式和對戰模式，教學模式裡可以使用建議著手、棋型搜尋、棋局搜尋和棋書等功能，而對戰模式則可以選擇與玩家或是電腦 AI 對奕，另外還實作了象棋兩人對弈的部分，藉由象棋遊戲的實作，讓其他符合遊戲模型的棋類遊戲也可以在本教學平台上實作，本系統主要是利用多點觸控的特性，讓棋類遊戲更具親和力也更加直覺，例如棋盤的旋轉和放大，六子棋一次下兩顆子的特性，並與後端的雲端系統[8]結合，增進電腦 AI 運算的效能，讓此棋類遊戲教學系統更加的完善，本系統在交大盃六子棋公開賽成功展示過了。

由於本系統是需要使用者下載安裝的應用程式，未來可望新增成網頁模式，讓使用者可以不必安裝直接連上網頁就可以使用，增加其使用的便利性，另外亦可新增 POT 遊戲模型裡另外兩種操作模式：Change-face 和 Change-Z 的相關遊戲，如撲克牌類的遊戲等。

參考文獻

- [1] Connect6 Homepage, available at <http://www.connect6.org/> .
- [2] CYC 遊戲大聯盟, available at <http://cycgame.com/> .
- [3] TED Iedas Worth Spreading, available at http://www.ted.com/talks/jeff_han_demos_his_breakthrough_touchscreen.html .
- [4] GridGain, available at <http://www.gridgain.com/> .
- [5] ICGA(International Computer Games Association) , available at <http://ticc.uvt.nl/icga/> .
- [6] LGS 傳奇網路圍棋廳, available at <http://lgs.taiwango.net/> .
- [7] Lin, P.-H., and Wu, I-Chen, NCTU6 Wins Man-Machine Connect6 Championship 2009, ICGA Journal, Vol. 32(4), pp. 230–232, 2009.
- [8] Lin, Yen-Cheng, Cloud Computing System for E-Learning Platform of Board Games, Master Thesis, National Chiao Tung University, 2010.
- [9] WIKIPEDIA-Cloud Computing, available at <http://zh.wikipedia.org/zh-tw/%E9%9B%B2%E7%AB%AF%E9%81%8B%E7%AE%97> .
- [10] WIKIPEDIA-Chinese Chess, available at <http://zh.wikipedia.org/zh-tw/%E8%B1%A1%E6%A3%8B> .
- [11] WIKIPEDIA-Connect6, available at <http://zh.wikipedia.org/wiki/Connect6> .
- [12] WIKIPEDIA-GO, available at <http://zh.wikipedia.org/wiki/Go> .
- [13] WIKIPEDIA-WPF, available at http://en.wikipedia.org/wiki/Windows_Presentation_Foundation .
- [14] Windows Azure, available at <http://msdn.microsoft.com/zh-tw/windowsazure/default.aspx> .

- [15] Wu, I-Chen, Hsu, Chien-Chih, The Model and Systems for Play-on-table Games, IEICE transactions on information and systems, E87-D(11), 2503-2508, 2004.
- [16] Wu, I-Chen., and Lin, P.-H., NCTU6-Lite Wins Connect6 Tournament, ICGA Journal, Vol. 31(4), pp. 240–243, 2008.
- [17] Wu, I-Chen, and Lin, P.-H., Relevance-Zone-Oriented Proof Search for Connect6, to appear in the IEEE Transactions on Computational Intelligence and AI in Games, 2010.
- [18] Wu, I-Chen, Cloud-based E-learning platform for Connect6 and related board games, 2010.
- [19] Wu, I-Chen, Huang, D.-Y., and Chang, H.-C., Connect6. ICGA Journal, Vol. 28(4), pp. 234-242, 2006.
- [20] Wu, I-Chen, and Yen, S.-J., NCTU6 Wins Connect6 Tournament, ICGA Journal, Vol. 29(3), pp. 157-158, September 2006.

