

移植電腦網路遊戲藉由代理伺服器輔助

Porting PC-based Network Game to Mobile Devices Through Agent Server

汪益賢	李柏甫	廖家茵	吳毅成
國立交通大學資訊科學與 工程研究所	國立交通大學資訊科學與 工程研究所	國立交通大學資訊科技產 業研發專班碩士班	國立交通大學資訊科學與 工程研究所
wangys@csie.nctu.edu.tw	fou.cs96g@g2.nctu.edu.tw	cyliao.iit97g@nctu.edu.tw	icwu@csie.nctu.edu.tw

摘要

本論文發展了一套軟體移植流程，將電腦的網路遊戲平台移植至行動裝置。首先我們提出代理客戶端模式，將遊戲客戶端分為行動裝置客戶端與代理伺服器，同時代理伺服器能過濾資訊，讓行動裝置程式碼單純化。其次我們考量到要移植到多種不同平台的行動裝置上，我們以 Qt Framework 開發手機上的遊戲程式平台，以達到在多種平台的行動裝置上運行。

關鍵詞:電腦網路遊戲、移植、行動裝置、代理客戶端模式

Abstract

In this thesis, we develop a fast software process that ports network-based game platforms from PCs to mobile devices. First, we propose an architecture model, called an agent model. In this model, game clients are split into mobile code and agent servers, such that agent servers can filter unnecessary messages and make mobile code simpler to fit in mobile devices. For porting games to mobile devices, we use Qt framework to develop mobile game client.

Keyword: network-based game、port、mobile device、agent model

一、研究緣由與目的

現今各式行動裝置(手機、PDA)已相當普及，性能也較以往進步許多，也已有相當大量的多媒體娛樂應用。根據台灣 2007 數位年鑑 [8]，美日各國的手機遊戲市場日益上升，然而台灣在行動電話的附加服務僅有 10%，仍有相當大提升的空間。

市面上的手機上的遊戲，目前仍是以單機為主。使用者在網路上找尋遊戲或透過手機加值服務下載後，遊戲單機進行。手機上網遊戲發展則較為緩慢，遊戲數量遠低於單機遊戲，是尚待開發的市場。

將電腦網路遊戲移植到行動裝置，主要會面臨的問題有：

- 行動裝置的限制
- 現有遊戲系統平台之移植
- 行動裝置平台歧異

1.1 行動裝置的限制

行動裝置為了攜帶性與電池續行力，效能較一般電腦弱，螢幕大小也受到限制。接下來會介紹行動裝置的資源與先天的限制。

以下是在本研究開始時，幾款高階手機的資源列表[9]。

表 1-1、高階手機資源表

	Nokia N82	HTC Touch Elfin	HTC Touch Cruise
--	-----------	-----------------	------------------

CPU (MHZ)	332	201	400
Ram (MB)	128	128	128
LCD (pixel)	240 * 320	240 * 320	240 * 320

各款手機的 CPU 時脈、記憶體以及螢幕解析度都遠遜於一般電腦，手機網路也較一般電腦網路慢。

除了上述資源較電腦少的問題外，還有許多行動裝置的先天上的限制：

- 行動裝置上網路費率較高。
- 過小的鍵盤造成打字不方便。
- 電力有限，做越多的運算或透過網路傳輸大量的資料[10]會造成更多電力消耗。

1.2 現有遊戲系統平台之移植

在移植遊戲客戶端至行動裝置後，現有系統多了加入了行動裝置客戶端，可能會引發許多改變。而為了探討這個議題，我們先來看一般電腦網路遊戲平台的架構圖。

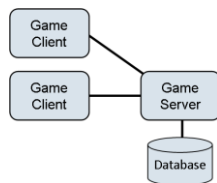


圖 1-1、一般電腦網路遊戲平台架構圖

遊戲伺服器(Game Server):供大量遊戲客戶端連線，進行遊戲的服務，功能包含遊戲大廳做遊戲者的配對，與廣播或轉送遊戲的訊息指令，進行遊戲邏輯的檢查。

遊戲客戶端(Game Client):安裝在使用者的電腦，功能包含遊戲畫面的顯現、操作，與邏輯運算等等。

遊戲資料庫(Database):儲存使用者資訊，包含遊戲狀態、使用者帳號密碼、遊戲歷程等等。

一般我們認為直覺的移植方式，是將原先電腦上的遊戲客戶端功能，移植至行動裝置上，行動裝置直接連線到遊戲伺服器取得遊戲服務，如同下圖 1-2。

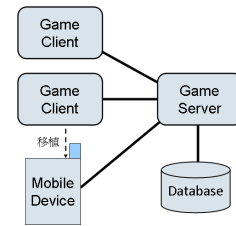


圖 1-2、一般電腦網路遊戲移植至行動裝置的平台架構圖

但由於先前提到行動裝置限制的問題，為了使行動裝置的遊戲客戶端能夠運行順暢，需要做下列的修改：

- 簡化圖形化操作介面(GUI):考量到行動裝置性能，要降低 CPU 以及記憶體的使用，並且讓圖形化操作介面的顯示適用於行動裝置。
- 減少廣播訊息與精簡大廳訊息:由於行動裝置網路傳輸的費率較高，而且無線網路裝置在傳遞資料時較待機消耗較多的電力。
- 刪除聊天系統:因應行動裝置打字不便以及螢幕的限制。

做以上修改，將會需要修改遊戲伺服器程式，將會使開發時間拉長與維護成本上升，每次修改需做電腦上的遊戲客戶端、遊戲伺服器程式、行動裝置上的客戶端三者間的完整測試。

1.3 行動裝置平台歧異

行動裝置上的平台種類非常多，要在平台上開發軟體，大都需要針對特定平台開發，彼此間無法互通。在本研究開始時，行動裝置上一般常見的平台有下列三種：

- Symbian[17]
- WinCE[13]
- Linux Embedded[18]

各平台上的開發框架(Framework)差異極大，因此當要將遊戲移植至不同行動裝置時，需將遊戲程式改寫為符合該平台開發框架。原先只需維護一份遊戲程式碼，為了在三個常見平台上執行，需維護三倍程式碼，會造成遊戲

程式碼後續維護的困難。

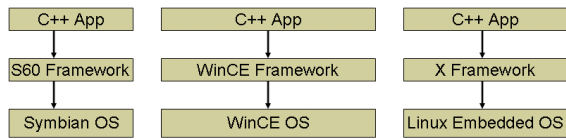


圖 1-3、在各平台上開發軟體的示意圖

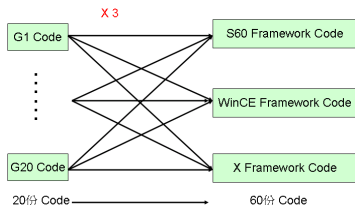


圖 1-4、程式維護困難示意圖的示意圖

1.4 研究目的及大綱

本論文最主要的目的是要將電腦網路遊戲快速移植到行動裝置上，解決行動裝置的限制、對於現有遊戲系統平台之移植、平台歧異這三大問題。在行動裝置上實作出與原先電腦客戶端功能接近的行動裝置客戶端。本研究將把群想的 CYC 遊戲平台實驗系統移植到行動裝置上作為展示實例，來驗證本方法是否有效，由於移植時使用的是實驗系統，與實際系統有所差距，但足以顯現本研究之成效。

本論文第一段描述研究動機、目的與面臨的問題，第二段介紹移植平台、開發環境與論文考量，第三段介紹移植方法，第四段展示各項實驗結果，第五段說明結論與未來展望。

二、研究背景

本章會先介紹 CYC 遊戲平台，之後介紹本研究開發語言的考量與選擇，最後以遊戲開發的角度，提出我們在設計行動裝置客戶端時的考量。

2.1 CYC 遊戲平台介紹

CYC 遊戲平台原則上依循圖 1-1 之架構，客戶端以 Java 撰寫，可運行在 Windows、Linux、Mac 各種電腦作業系統下，以網頁開啟 Java Applet，不需進行安裝遊戲的動作。

CYC 遊戲平台上有各種的休閒遊戲，如六子棋、大老二、麻將等。在開發上提供了高階開發環境，實作了三大類型(麻將、棋類、牌類)遊戲的模版，使整個系統擴充性相當好。

CYC 遊戲平台架構[1]可簡化為三個部分: 遊戲資料庫、遊戲伺服器與遊戲客戶端。

CYC 遊戲平台的遊戲資料庫與一般平台相同，在此便不再多提。

CYC 遊戲平台的遊戲伺服器提供了大廳的功能，讓玩家可進行配對遊玩，並且記錄各遊戲桌的資訊與驗證使用者操作。

CYC 遊戲平台的遊戲客戶端負責遊戲畫面的顯現與玩家的操作，同時遊戲指令的運算是在此處理，而遊戲客戶端的階層架構如下圖。

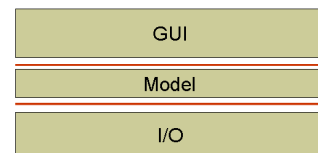


圖 2-1、電腦遊戲客戶端之階層架構圖

- 輸入輸出(I/O):採用 TCP/IP Protocol，以事件驅動(Event Driven)的方式進行收送封包，並且在封包傳送時進行加密，以免被竊聽或假造封包。
- 進行模式(Model):負責大廳、遊戲的指令與訊息處理，進行各種的運算與狀態的改變控制。
- 圖形化操作介面(GUI):進行畫面與各種遊戲特效的顯現，以及使用者的操作遊戲與大廳的介面。

接下來來講解 CYC 進行遊戲的流程。

1. 連線登入至遊戲伺服器後，將會顯現大廳畫面，在這裡畫面分為四個部分:
 - 聊天功能:畫面左下為顯示大廳聊天訊息，以及玩家輸入聊天訊息的部分。
 - 使用者狀態:畫面右下為所有使用者的狀態列表，顯示使用者暱稱、所在桌與分數。使用者的資訊為及時更新，其他玩家進行入桌、離桌、登入、登出等動作，均即時

的由遊戲伺服器傳送過來。

- 圖形化使用者配對介面:位於畫面的左上部分。在此以圖形化的方式顯現各遊戲桌有哪些使用者，並以點選椅子的方式進行開桌、入桌的使用者配對功能，玩家資訊更新也是即時的在此顯現。在此會貼了大量桌椅、玩家角色的圖片，耗費了相當多的記憶體。
- 功能列表與遊戲桌狀態縮圖:在此顯示了各項功能按鈕及遊戲桌狀態。



圖 2-2、CYC 大廳畫面(取自群想公司[7])

2. 在點選入桌或開桌後，將會顯示或調整該遊戲桌的遊戲設定，如遊戲時間、是否啟動特殊規則、是否聊天、是否計分等設定。
3. 遊戲設定完成，便進入遊戲畫面中，左半部為遊戲畫面的操作，右半部則顯示玩家的角色圖示以及聊天介面。



圖 2-3、CYC 六子棋遊戲畫面(取自群想公司[7])

2.2 開發語言

由於先前提到程式維護的問題，我們在選擇開發行動裝置遊戲客戶端的語言時，會希望一份程式盡可能跨多平台。而現今在行動裝置上一般常見的跨平台方案有兩種：

- Java ME[16]
- C/C++搭配 Qt[14]

但上述兩者不見得支援所有行動裝置平台。為了未來的擴充性考量，最後我們選擇了具有較多平台支援的 C/C++。倘若選擇 Java ME，未來想擴展的平台上無 Java Virtual Machine，自行實作太過困難，選擇 C/C++搭配 Qt，該平台上若無 Qt，我們實作所用到的 Qt 部分功能即可。

2.3 遊戲開發考量

開發一款電腦網路遊戲需要考量的要素，架構、元件的設計以及流程的完善，在徐建智[2]這篇論文已深入探討，並應用於實際的遊戲平台上。而在行動裝置上運行遊戲時，需考量資源與測試的問題，在陳智文的論文[5]有相當多的見解。而在許珉嘉[6]與高偉傑[3]論文中，提出了跨平台開發機制的設計，作為我們在開發行動裝置客戶端時的設計考量。

三、行動遊戲平台移植之設計

本章會先介紹加入行動裝置客戶端後，如何達到快速移植的代理系統(Agent System)設計，以及系統如何運行。接下來介紹行動裝置客戶端如何精簡，並與電腦遊戲客戶端比較。

3.1 代理系統的設計

在先前已提到過，如圖 3-1 的方式，以一般直接將電腦上的遊戲客戶端移植到行動裝置上，會需要修改遊戲伺服器程式，造成系統開發時間長與維護成本高的問題。

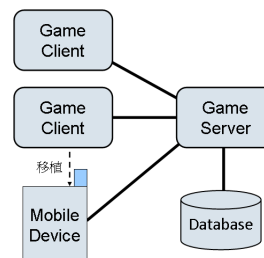


圖 3-1、一般電腦網路遊戲移植至行動裝置的平台架構圖

為了解決需要修改遊戲伺服器的問題，我們設計將原先電腦的遊戲客戶端增加了代理 (Agent) 的功能，替行動裝置的遊戲客戶端連線至遊戲伺服器，充分利用運行已久且穩定的連線機制，而連線的模式將如圖 3-2。

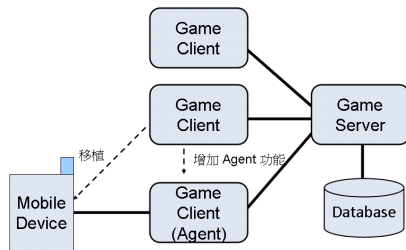


圖 3-2、行動裝置透過增加代理功能的電腦遊戲客戶端連線至遊戲伺服器

我們提出了以代理的模式，在遊戲平台架構中增加了代理伺服器，這部分代理的功能也可考慮實作在遊戲伺服器或行動裝置中，但都會有缺陷，因此我們選擇將代理伺服器獨立出來。

- 在遊戲伺服器實作代理功能: 遊戲伺服器功能非常複雜，對於以穩定營運的遊戲平台而言，修改遊戲伺服器風險相當高，可能會造成伺服器運行時當機，讓電腦與行動裝置玩家均無法遊戲，這種情況是遊戲公司不願意見到的。
- 在行動裝置實作代理功能: 遊戲伺服器沒做修改，因此無法區分行動裝置與電腦客戶端，廣播訊息會一視同仁的傳送至此兩類客戶端，無法達到節省網路頻寬的功能。
- 獨立出代理伺服器: 可不修改遊戲伺服器，且由於代理伺服器由遊戲公司提供，代理伺服器與遊戲伺服器兩者間傳輸不需擔心頻寬問題，也可達到降低傳輸至行動裝置客戶端的網路流量。

依據上述設計，我們預期該連線模式會有下列的優缺點。首先優點如下:

- 開發時間可縮短: 因此設計可不修改遊戲伺服器，而遊戲伺服器與原先電腦上的遊戲客戶端溝通也不需任何修改，僅電腦遊戲客戶端需要修改，增加與行動裝置溝通的

代理功能，既有系統的改動相當少。

- 維護成本降低: 因為增加行動裝置的遊戲客戶端僅與電腦上的遊戲客戶端溝通，其餘機制並不會變動，因此我們只需做行動裝置的遊戲客戶端與電腦上的遊戲客戶端的連線測試即可，不需擔心造成原先電腦上的遊戲客戶端不穩定，可減少測試工作。

缺點如下:

- 增加營運成本: 遊戲公司需要提供代理功能的遊戲客戶端主機，以供行動裝置遊戲客戶端連線，與一般連線模式相比，是額外的支出。

接下來三個小節會分別針對代理系統的架構圖、說明系統如何啟動以及行動裝置如何進行遊戲服務來做詳細的說明。

3.1.1 代理系統架構圖

依循我們先前提到的概念，我們在行動裝置上實作行動裝置客戶端 (CYC Light Client)，而將附加代理功能的電腦遊戲客戶端設計為代理伺服器 (Agent Server)，並可接受多個行動裝置客戶端的連線，產生出對應處理行動裝置客戶端連線的附加代理功能的遊戲客戶端程序 (process)，替行動裝置連線至遊戲伺服器，傳達大廳與遊戲操作。

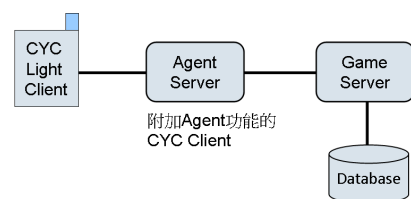


圖 3-3、單一代理伺服器架構圖

為了未來系統的擴充性，我們考慮到了會有同時多台代理伺服器提供行動裝置連線進行服務的情況，因此修改了原先單一代理伺服器的架構圖，提出了主要代理伺服器 (Master Agent Server)，以達到能夠提供多台代理伺服器同時服務的設計。

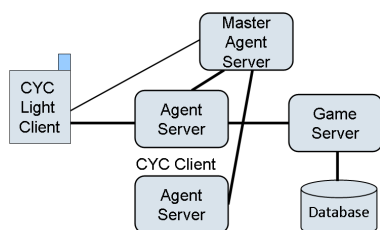


圖 3-4、多台代理伺服器架構圖

- 主要代理伺服器:作為行動裝置客戶端的連線入口，掌握所有可提供服務的代理伺服器的服務 IP、Port、服務的遊戲以及負載狀況，在行動裝置客戶端要進行遊戲服務時，挑選負載較低的代理伺服器為行動裝置客戶端進行服務。
- 代理伺服器:由原先電腦上的遊戲客戶端所修改，增加了代理的功能，以命令模式(Console Mode)執行，替行動裝置客戶端連線至遊戲伺服器，進行遊戲與大廳操作與訊息的轉傳處理，
- 行動裝置客戶端:為了行動裝置實作的精簡客戶端，可進行大廳、遊戲的操作。
- 遊戲伺服器:與原先 CYC 平台遊戲伺服器完全相同，沒做任何改變。
- 遊戲資料庫:與原先 CYC 平台的遊戲資料庫相同，沒做任何改變。

3.1.2 代理系統的啟動

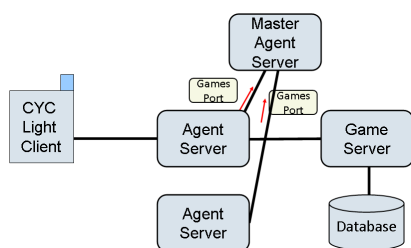


圖 3-5、代理系統啟動圖

如圖 3-5 在啟動主要代理伺服器後，各台代理伺服器連線至主要代理伺服器，傳輸各台代理伺服器提供服務的遊戲種類以及供行動裝置客戶端連線的 Port 至主要代理伺服器。

在啟動完後，主要代理伺服器與各台代理伺服器仍保持連線，各台代理伺服器定期回報負載，以供主要代理伺服器在為行動裝置客戶端挑選代理伺服器服務的選擇時做參考依據。

3.1.3 行動裝置客戶端進行遊戲服務流程

接下來本小節將詳細說明行動裝置客戶端如何進行遊戲服務的流程，流程圖如下圖 3-6。

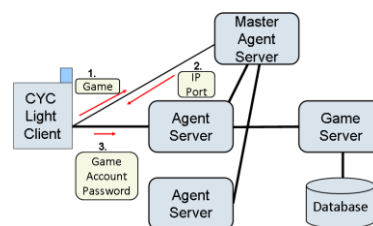


圖 3-6、行動裝置客戶端進行遊戲服務流程

首先行動裝置客戶端連線到主要代理伺服器，並送出要進行的遊戲名稱。

主要代理伺服器從目前正在服務的代理伺服器中找尋有服務該遊戲的代理伺服器，挑選負載較低者，將其供行動裝置客戶端連線的 IP 與 Port 傳送至行動裝置客戶端。

行動裝置客戶端收到目標代理伺服器的 IP、Port 之後，切斷與主要代理伺服器的連線，而後連線至目標代理伺服器，傳送要進行的遊戲名稱以及玩家的帳號密碼，透過代理伺服器登入至遊戲伺服器，進行遊戲的服務。

3.2 行動裝置客戶端與代理伺服器設計

原先電腦客戶端內部設計如圖 3-7，而行動裝置客戶端與代理伺服器內部設計與電腦遊戲客戶端設計圖如下圖 3-8。

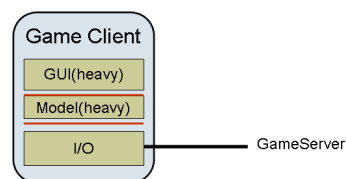


圖 3-7、電腦遊戲客戶端設計圖

電腦客戶端相較於行動裝置客戶端，圖形化操作介面、模式設計都是重量級，功能與程式碼數量都較多。

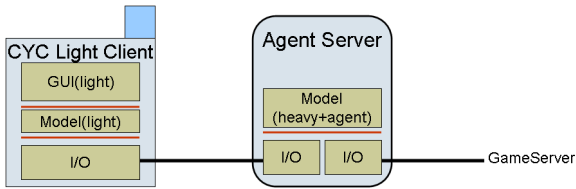


圖 3-8、行動裝置客戶端與代理伺服器設計圖

代理伺服器:

- 圖形化操作介面:代理伺服器以命令模式運行，因此不需要圖形化操作介面模組。
- 進行模式:功能為電腦上遊戲客戶端的重量模式加上代理的功能。
- 輸入輸出:模組分成跟遊戲伺服器與跟行動裝置客戶端溝通的輸入輸出模組，行為模式與電腦上的遊戲客戶端輸入輸出模組完全相同。

行動裝置客戶端:

- 圖形化操作介面:模組作精簡化，大量減少貼圖與特效，降低 CPU、記憶體的使用，使得圖形化操作介面顯示適用於行動裝置。
- 進行模式:經過精簡化後，處理的指令與訊息較為精簡。許多資訊僅在代理伺服器儲存及處理，盡可能減輕行動裝置客戶端的網路傳輸量以及運算負擔。
- 輸入輸出:模組與代理伺服器上輸入輸出模組溝通，模式相同。

3.2.1 針對行動裝置資源限制設計

為了降低行動裝置資源使用的負擔，我們主要針對三個方面進行精簡:

- 圖形化操作介面簡化:減少可精簡的貼圖，例如使用者的紙娃娃(Avatar)與大廳的桌椅等圖，可大幅減少元件數，降低記憶體的使用量與 CPU 的運算量，同時使圖形化操

作介面顯示符合行動裝置螢幕大小。

- 精簡大廳通訊量:精簡大廳傳輸的訊息量，只將與行動裝置使用者以及對手的傳輸至行動裝置，不需要掌控完整的大廳狀態，其餘資訊僅在代理伺服器處理即可，不將訊息傳輸至行動裝置。同時去除了原先電腦遊戲客戶端上大廳訊息的即時更新，可大幅降低訊息量。由行動裝置傳送到代理伺服器的指令也較為精簡，由代理伺服器補齊為完整指令送出至遊戲伺服器。
- 刪除聊天系統:因應行動裝置打字不便以及螢幕過小，便選擇刪除了聊天系統。

3.2.2 行動裝置啟動流程

1. 首先玩家下載行動裝置客戶端的程式至行動裝置中，在執行後會顯示遊戲的選單，讓玩家選擇要進行的遊戲，例如六子棋或兩人大老二。
2. 在決定要進行的遊戲後，會進入到遊戲模式的選擇，可選擇電腦的對戰與網路對戰的功能。其中網路對戰為本系統重點。
3. 點選網路模式後，將會顯示登入畫面，使用者在此輸入帳號與密碼，其中帳號為自動記憶，密碼則可自行選擇是否記憶。
4. 成功登入至遊戲伺服器後，接著顯示大廳畫面，在此可選擇新開遊戲桌或加入待配對的遊戲桌。其中待配對遊戲桌的資訊是以被動更新的方式，當使用者點選重整時才進行更新。



圖 3-9、行動裝置大廳畫面(圖片來源自群想公司[7])

5. 在點選入桌後，進行該桌遊戲的遊戲設

定，如同在電腦客戶端的遊戲設定。

6. 完成遊戲設定入桌後，即可進入遊戲畫面，在此僅可進行遊戲，無法進行聊天，也不顯示玩家的角色圖示。



圖 3-10、行動裝置遊戲畫面(圖片來源自群想公司[7])

3.2.4 精簡大廳訊息範例

在這一節將會介紹介紹，如何降低大廳訊息傳輸量，並以各種例子說明。

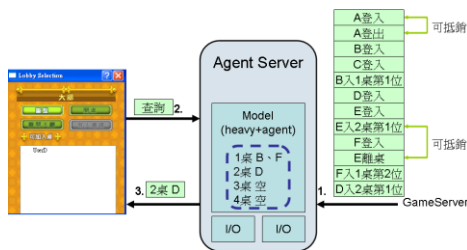


圖 3-11、大廳訊息精簡範例 1:找尋待配對的遊戲桌-簡易範例

所有線上使用者的登入、登出、入桌、離桌訊息都即時的傳送至代理伺服器，在代理伺服器掌握了所有線上使用者的最新所在桌資訊。

圖 3-11 為一簡易的狀態快照，在此時行動裝置客戶端向代理伺服器要求取得尚待配對遊戲桌資訊，代理伺服器從遊戲桌中找出符合條件的遊戲桌，僅將這部分資訊傳遞至行動裝置客戶端。採用這種模式來找尋待配對遊戲桌，以此簡易的範例，即可大量的降低傳輸的訊息量。

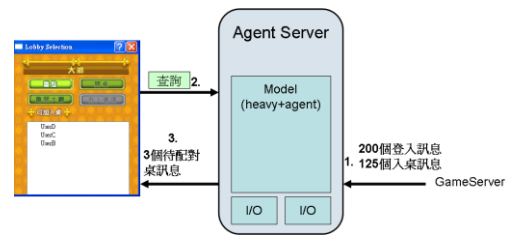


圖 3-12、大廳訊息精簡範例 2:找尋待配對的遊戲桌-實際範例

圖 3-12 中的範例，同樣是使用者在找尋待配對遊戲桌，而訊息數以接近實際玩家人數估算後，共有 325 個訊息傳送至代理伺服器，而僅 3 個訊息至行動裝置客戶端，由此可發現減少了相當多的訊息量，依此種代理模式，由代理伺服器做資料的過濾精簡，可取得相當好的效果。

四、實作及相關問題探討

本研究將原先電腦上的遊戲客戶端增加了代理的功能，並在行動裝置上實作了下列的部分：

- 遊戲平台:包含了大廳，以及各式選單、網路傳輸加解密等底層的實作、整個程式流程控制為遊戲平台所涵蓋。
- 六子棋遊戲。
- 大老二遊戲。

接下來各個小節將會系統記憶體、CPU、網路、畫面資源的使用來分析探討，以及審視最後完成系統所減少的程式碼數量比例。由於本研究是使用群想 CYC 平台的實驗性系統，並非實際正在營運的系統，因此在各項實驗評測與實際系統會有差距，但已可顯現本研究之成效。

4.1 遊戲程式記憶體使用量

本實驗在觀察電腦客戶端以及行動裝置客戶端在實際進行遊戲時，記憶體的消耗狀況。

表 4-1、執行遊戲記憶體使用量比較(單位為 MB)

	電腦	行動裝置
大老二	超過 100	9.67
六子棋	超過 100	10.12

表 4-1 為電腦與行動裝置在執行遊戲時，記憶體的使用量。由表中可發現，在電腦上運行遊戲，需消耗超過 100 MB 的記憶體，已超過一般高階手機擁有的記憶體量。最後系統在行動裝置上實際運行，所消耗的記憶體大約 10MB 左右，是可接受的。

4.2 代理伺服器記憶體使用量

當行動裝置客戶端連線至代理伺服器進行遊戲服務時，代理伺服器需產生附加代理功能的遊戲客戶端程序，來服務該客戶端。本實驗即是統計每增加一個行動裝置進行服務時，代理伺服器的記憶體使用狀況。

表 4-2、代理伺服器記憶體消耗量

	記憶體用量(MB)
代理伺服器初始時	26.31
1 個大老二遊戲行動裝置連線	29.23(增加 2.92)
1 個六子棋遊戲行動裝置連線	31.60(增加 5.29)

由表 4-2 中可觀察出，每增加一個行動裝置的連線，消耗記憶體不超過 6MB。

4.3 代理伺服器 CPU 使用量

我們將模擬 10 台行動裝置進行大廳的操作以及遊戲的操作，觀察代理伺服器上 CPU 負載量的變化，作為效能的評測與估計系統能負擔行動裝置連線數的上限。

首先我們模擬 10 台行動裝置連線後，每 1 秒不停發送開桌離桌的指令，每 30 秒記錄一次負載，共記錄 20 次，統計出平均的 CPU 負載與出現過的最高 CPU 負載。本實驗在兩台記憶體充足的主機上運行代理伺服器做統計，由於兩台主機 CPU 核心數不同，因此在統計時僅以一顆核心為基準。

表 4-3、模擬 10 台行動裝置連線開桌離桌，在 Intel Q6600 CPU 主機上負載

僅以一顆核心為基準	平均負載	尖峰負載
10 個大老二遊戲行動裝置連線	51.2%	68.0%
10 個六子棋遊戲行動裝置連線	70.4%	84.0%

接著我們模擬 10 台行動裝置連線後，每秒發出下棋或出牌的動作，每 30 秒記錄一次負載，共記錄 20 次，統計出平均的 CPU 負載與出現過的最高 CPU 負載。如同在統計開桌離桌的負載量一樣，我們在統計時僅以一顆核心為基準。

表 4-4、模擬 10 台行動裝置連線遊戲操作，在 Intel Q6600 CPU 主機上負載

僅以一顆核心為基準	平均負載	尖峰負載
10 個大老二遊戲行動裝置連線	1.3%	4.0%
10 個六子棋遊戲行動裝置連線	2.4%	8.0%

接著我們統計了目前實際運作的系統一個遊戲中，每秒所有使用者的動作次數與其比例，統計時該遊戲的使用者人數約為 200 人。

表 4-5、玩家動作次數與比例統計

	遊戲動作	開桌離桌動作
次數及比例	46.0 (99.7%)	0.31 (0.3%)

根據這三項代理伺服器資源使用的實驗與玩家動作之比例，我們可知代理伺服器每增加一個行動裝置連線：

- 最多消耗 6MB 的記憶體。
- 以單一核心服務，0.3% 的玩家進行大廳操作，99.7% 玩家進行遊戲動作估計，以較高的平均、尖峰負載估計，平均負載為 0.260%；尖峰負載 0.823%。

以此實驗結果估計，在 Agent Server 記憶體充足的情況下，主機一顆核心約可服務 385 條行動裝置連線，而假若同時上線人數有 10000 人，有 5% 是行動裝置玩家時，僅需 2 台單核心的 Agent Server 即可。

4.4 網路測試

由於行動裝置的操作，都要透過代理伺服器轉送至遊戲伺服器，因此與代理伺服器傳遞訊息的延遲時間，會影響遊戲操作的流暢度。本實驗在行動裝置上嘗試了 Wi-Fi 連線以及透過 GPRS 連線上網，測試封包由行動裝置與代理伺服器往返的時間。

表 4-6、封包由行動裝置至代理伺服器往返一次的時間

	時間(ms)
Wi-Fi	87.7
GPRS	768.7

在表 4-6 中顯示了封包從行動裝置送出至代理伺服器接收到後，代理伺服器再送封包回去，直到行動裝置收到的平均時間，由此可知每一次操作，大約一秒內可反應。

接下來測試則是要驗證我們的設計是否可降低網路的傳輸量，因此我們先模擬了一個簡易的操作範例，並統計了經過這些操作後，行動裝置客戶端、代理伺服器以及遊戲伺服器三者之間的網路傳輸量。用此實驗來驗證我們的設計是否取得成效。

首先我們驗證先前圖 3-11 的精簡大廳資訊簡易範例，代理伺服器與遊戲伺服器間的傳輸量為 2563Bytes，而行動裝置客戶端與代理伺服器間的傳輸量為 744Bytes，在這個簡易的範例中，便可將網路傳輸量降低為 1/3。

接著我們以先前統計的傳輸量，以及接近實際運行系統的人數快照，圖 3-12 的範例來預估傳輸量的縮減程度。由計算後可得出在接近實際運作狀況下，代理伺服器與遊戲伺服器間的傳輸量為 42612Bytes，而代理伺服器與行動裝置客戶端的傳輸量為 894Bytes，傳輸量可降低為約 1/40，網路傳輸量降低相當多。

4.5 系統程式碼分析

在本系統完成之後，我們統計了電腦客戶端以及行動裝置客戶端的各項功能程式碼數量，來觀察以此設計方式移植開發出行動裝置客戶端，是否可取得減輕工作量的成效。

在單一遊戲六子棋方面，我們由電腦客戶端移植至行動裝置客戶端的程式碼縮減為原先的 16.6%，整個遊戲系統的程式碼量也縮減為原先的 15.2%。

除了觀察行動裝置客戶端的完成所需程式碼數量縮減外，將電腦客戶端修改為代理伺服器也是本移植系統的關鍵，因此這部分修改的程式碼數量也要列入工作量，經過統計後，我們僅需修改電腦客戶端 2.2% 的程式碼即可。

五、結論與未來發展

在本論文當中，我們設計了一個將電腦網路遊戲快速移植到行動裝置的方法，同時實作在 CYC 平台實驗系統的移植上。這個方法有以下特點：

- 不需修改既有系統的遊戲伺服器:依照本論文的移植方式，可完全不修改既有系統的遊戲伺服器，就可達到行動裝置連線進行遊戲的功能。
- 既有系統只有電腦遊戲客戶端做小幅修改:將電腦遊戲客戶端新增代理功能，只需要修改 2.2% 的程式碼即可達成代理功能。
- 行動裝置資源消耗可接受:在實際將系統在行動裝置上執行後，速度尚可接受，記憶體消耗仍在允許範圍內。

開發一套將電腦網路遊戲移植到行動裝置的方法，其實仍有許多地方要考慮。本論文至此提出了一個基礎的解決方案，未來仍有些課題可以繼續深入探討。以下列出一些思考方向，希望能夠針對這些項目加以改進，進而完善我們的系統。

擴充可支援的行動裝置平台:本系統藉由 Qt 的幫助，可在多種行動裝置平台上運行。但近來 iPhone 與 Android 越來越盛行，Qt 無法在這兩套系統上運行。希望未來移植的系統能夠支援更多的平台。

提升代理伺服器的效能:代理伺服器在服務行動裝置連線時，CPU 負擔不低，成為服務大量行動裝置連線的瓶頸。目前代理伺服器服務行動裝置時，是以一個完整的電腦遊戲客戶端

附加代理功能服務，未來可精簡或讓多個行動裝置連線共用電腦遊戲客戶端來降低系統負擔。

目前我們已使用本論文的方法將群想 CYC 實驗系統移植至行動裝置上，並設計了幾個範例遊戲。相信依照此模式，也可應用到其他遊戲平台的移植。

誌謝

感謝行政院國家科學委員會(National Science Council)，(97-2622-E-009-003-CC3)與(98-2622-E-009-001-CC3)之計畫與群想網路科技股份有限公司，對本研究提供相關支持，包含遊戲實驗性系統，遊戲畫面的轉載、使用。

參考文獻

- [1] 汪益賢, CYC Platform, 群想網路科技股份有限公司, 內部文件, 2004.
- [2] 徐健智, 「網際網路上桌上型遊戲之發展平臺」, 國立交通大學資訊工程學系, 碩士論文, June 2000.
- [3] 高偉傑, 跨平台手機遊戲開發框架在 Symbian 手機之研究, 國立交通大學資訊工程學系, 碩士論文, July 2008.
- [4] 陳凌彬, 「網際網路遊戲發展平臺之研究」, 國立交通大學資訊工程學系, 碩士論文, June 2001.
- [5] 陳智文, 「手機遊戲發展平臺」, 國立交通大學資訊工程學系, 碩士論文, June 2004.
- [6] 許珉嘉, 跨平台手機遊戲開發框架在 WinCE 手機之研究, 國立交通大學資訊工程學系, 碩士論文, July 2008.
- [7] 群想網路科技股份有限公司, CYC Platform, available from <http://cycgame.com/>, 2009.
- [8] 經濟部工業局, 2007 台灣數位內容產業年鑒, 2008.
- [9] 億普媒體股份有限公司, ePrice 比價王手機館 <http://www.eprice.com.tw/mobile/>, 2009
- [10] Yuvraj Agarwal, Ranveer Chandra, Alec Wolman, Paramvir Bahl, Kevin Chin, Rajesh Gupta, “Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones”, 2007.
- [11] Jacco Bikker, “Mobilecore: A Cross-Platform Framework For ARM-Based Mobile Games”, Gamasutra, 2003.
- [12] Andreas Janecek, Helmut Hlavacs; “Programming Interactive Real-Time Games over WLAN for Pocket PCs with J2ME and .NET CF”; ACM Press New York, NY, USA, 2005.
- [13] Microsoft Corp, “Windows Embedded”, available from <http://www.microsoft.com/taiwan/windows/embedded/default.aspx>, 2009.
- [14] Nokia Corp, “Qt”, available from <http://www.qtsoftware.com/>, 2009.
- [15] J. Ousterhout, “Why threads are a bad idea, invited talk” at the 1996 USENIX Technical Conference, San Diego, CA, USA, 1996, see also <http://home.pacbell.net/ouster/threads.pdf> (last access: May 2005).
- [16] Sun Microsystems, “Java 2 Platform, Micro Edition(J2ME)”, available from <http://java.sun.com/javame/index.jsp>, 2004.
- [17] Symbian Corp, “Symbian OS”, available from <http://www.symbian.com/>, 2009.
- [18] Linus Torvalds, “Linux”, available from <http://www.linux.com/>, 2009.