

行動雲端雷電射擊遊戲之實作與分析

劉仲鑫* 張家榮*

*中國文化大學資訊工程學系

摘要

雲端運算，它是一種概念的延伸，透過電腦與網路的結合使提供的服務更加的無遠弗屆，未來在任何地方只需透過電腦與網路，就能完成一般性的工作，然後從此概念延伸出技術，且已成為台灣智慧型產業中的發展重點。然而，行動雲端運算的系統和應用程序比傳統的客戶服務模型的環境中更加複雜。因此，如何在雲端運算架構下增加資料安全性將是首要克服的問題。Android 系統是 Google 研發的手機作業系統。Android 採用了開源的 Linux 作業系統，底層使用了存取速度最快的 C 語言，應用層則透過簡單而強大的 Java 語言。本研究以 Android 系統開發出雷電射擊遊戲然後結合現今的雲端的技術，希望藉由雲端技術以及 Android 遊戲開發系統讓呈現小時候經典的雷電遊戲，此外進行雷電射擊遊戲時，玩家會進行攻城掠地，極力擴展其勢力範圍，因此本研究即進行覆蓋範圍分析。

關鍵詞：雲端運算、Android、雷電射擊遊戲

1. 緒論

1.1 研究動機

雷電射擊遊戲曾是最夯的遊戲，多重的關卡、敵人陣行的變化多端、每一道關卡的最後大魔王、遊戲背景、武器的造型和它的變化量等等...都是吸引玩家去玩的地方，而現今雖然也有很多類似的遊戲出現，還有許多 Android 系列的飛機射擊遊戲出現，但是玩過這麼多飛機射擊遊戲還是覺得以前的雷電遊戲是最好玩的也是最經典的飛機射擊遊戲，就是這點吸引著本研究作為較深入之探討。

1.2 研究目的

結合著現今發展還不錯的雲端技術和網路還有透過現今很紅的 Android 遊戲開發平台系統，試著透過這些開發平台來做出網路型的雷電射擊遊戲，可以讓更多玩家一起參與其中，一起作戰打敗最後的魔王，不管是在手機上或者是電腦中都可以開啟這款遊戲來執行並且使用它，而在路網分析的部分，會將所有有關於玩這款遊戲的年齡層、分布於全球或者台灣的哪些地方，透過雲端網路系統來得知，還有玩家的上線時段以時間多寡。本研究旨在做出屬於自己風格的雷電遊戲。

2. 文獻探討

2.1 Android

使用 Android 開發遊戲之前，掌握遊戲的基礎知識是很重要的。

首先我們要瞭解它的平台架構以及特性，Android 採用了整合性的策略思考，包括底層的 Linux 作業系統，中間層的中間套件和上層的 Java 應用程式。

Android 裡面有幾個比較核心的程式庫，包含 C 程式庫、媒體庫、Surface Manager、LibWebCore、SGL、3D libraries、FreeType、SQLite。[2]

Android 包含了一個核心庫，提供了 Java 程式語言核心庫的大多數功能。每一個 Android 應用程式都在自己的進程中進行，都擁有一個獨立的 Dalvik 虛擬機實例。Dalvik 是針對同時很有效率的執行多個 VMs 來完成的。Dalvik 虛擬機執行.dex 的 Dalvik 可執行檔，該檔針對最小記憶體使用了最佳化。[5]此虛擬機是基於暫存器的，所有的類別都經由 Javac 編譯，然後透過 SDK 中的 DX 工具轉換成.dex 格是在由虛擬機執行。Dalvik 虛擬機依賴於 Linux 一些功能，比如執行緒機制和底層記

記憶體管理機制。

Android 的核心系統服務是仰賴於 Linux 2.6 核心。如安全性、記憶體總管、進程管理、網路協定和驅動模型等，Linux 核心也同時做為硬體和軟體作為堆疊之間的硬體收抽象層。[9]

當我們在開發雷電射擊遊戲時，我們要先給它座標系統，例如：螢幕的左上角我們當作原點(0,0) 向右延伸當作 x 軸而向下延伸當作 y 軸，用此來設定飛機的位置以及螢幕的長跟寬，再來就是視圖邊界用來判斷在遊戲的同時的上下左右的位置才不會導致在遊戲時跑出邊界。

2.2 雷電遊戲開發

遊戲機本開發包括圖層物件、視圖物件和 Activity 物件。圖層物件包括圖形的繪製、圖層的寬和高還有移動。視圖物件包括圖形的繪製還有回應鍵盤的動作設定。Activity 物件包括啟動遊戲、暫停遊戲、停止遊戲等等遊戲的流程。

Java ME MIDP 2.0 提供了 API 是專門用來遊戲開發所使用的剛好跟 Android 也是使用 Java 語言，可以減少 Android 的遊戲開發工作量，同時 Java Me 是使用 Image 而 Android 使用 Bitmap 圖片類別兩者是可以替換的，而 Java Me 繪製圖片的類別為 Graphics，Android 則是 Canvas，而 Canvas 可以代替 Graphics，因此算是滿互補的。[7]

Layer 類別是用來定義遊戲的視覺化圖層，所以基本都要有圖層的目前座標、寬和高、是否可以看見。[3]

Sprite 類別是用來檢測圖層碰撞還有提供圖片轉換，是一個基本的動畫圖層類別。[3]

TiledLayer 類別移植，因為在開發遊戲的同時有許多圖案是非常大的，所以他的功能是用來將那些大圖片切割成小小的圖片，才可以有效且方便的載入或是進行操作。[3]

LayerManager 類別移植是用來管理圖層，才能新增多個 Layer 物件，方便集中維護和顯示。

在上述的 TiledLayer 類別移植、LayerManager 類別移植和 Sprite 類別，他們的 Graphics、Image

都必須替換成 Canvas、Bitmap。[3]

在步驟的部分，必須先建立一個 Android 專案，再來新增一個套件 X，將移植好的 Java Me 遊戲 API 複製到套件中，再新建一個新的物件 Y 直接加入一個 Sprite 的 Player 類別，這是代表玩家的圖層。此類別包括了螢幕的寬和高、玩家圖層的寬和高、鍵盤移動方向、圖層的移動速度、圖層是否看的見、圖層是否存活，這些屬性在類別裡的程式碼都必須先設定好再將此類別加入該套件中，如果沒有設定好會出現一些 bug 的情形，因為程式碼很龐大，如果等到他出 bug 再回去修改會花費太多的時間，所以先設定好然後檢查再套入是會比較好的選擇。[8]

定義敵人的圖層物件時，因為定義方法和定義玩家本身的圖層物件類似，只需修改圖層的座標和移動方式和方向就好了，其他部分和定義自己玩家本身的物件都大同小異。

遊戲中最重要的就是用子彈將敵人打落才可以破關，如果沒有子彈，遊戲也不會結束，就只是雙方角色在螢幕上移動而已，所以子彈對這款遊戲來說是很重要的，所以必須設定一個子彈的類別，裡面有子彈的圖層、按發射鍵會有子彈射出的功能、還有玩家以及敵人的存活，玩家如果被子彈打到數下就會因為血量不夠而死亡，而敵人方面也是。

定義遊戲的視圖類別，裡面包括了玩家、敵人、子彈的 Bitmap 和爆炸圖片，還有檢測被子彈打到後是否爆炸，是的話就會顯示爆炸的圖片。還有回應鍵盤事件，就是鍵盤的上下左右功能和射擊功能，隨時刷新螢幕和顯示。這部分的程式碼換是滿重要的，包括了整個遊戲的流程、完整性，還有流暢度。

MainActivity 是定義控制遊戲流程的類別，此類別繼承了 Activity，主要是控制遊戲流程，包括：啟動遊戲、暫停遊戲、停止遊戲。此類別也是個不可或缺的類別。[10]

在遊戲完成之後，最後的步驟就是將此款遊戲上傳是 Android Market，首先我們也要擁有自己 Market 的帳戶，並且需要一張國際信用卡(VISA

或 Master)，這樣就可以開發自己的遊戲上傳至 Market 供客戶下載，也成了一個賺錢的條件。但是最近 Market 已經改成了全部免費的模式，上面的所有東西都是不用錢的可以任意的讓客戶們下載自己所需要的東西，雖然沒有辦法賺錢，遊戲開發者還是會繼續上傳很多遊戲到 Market 裡面提供給客戶下載也可以打響知名度。

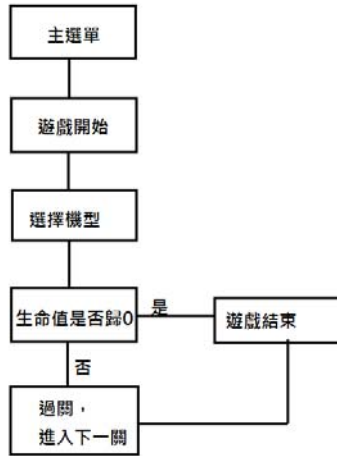


圖 1 遊戲的流程圖

3. UML 模型系統

玩家登入遊戲後，系統會進行驗證接著顯示所有關卡模式，在選擇想要玩的關卡點選進去就開始遊戲了，如果在遊戲中死亡就會直接跳回主畫面，沒有的話就繼續玩直到死亡為止。這些東西都會利用雲端讀取公司的資料，也會將資料傳送到公司等等。

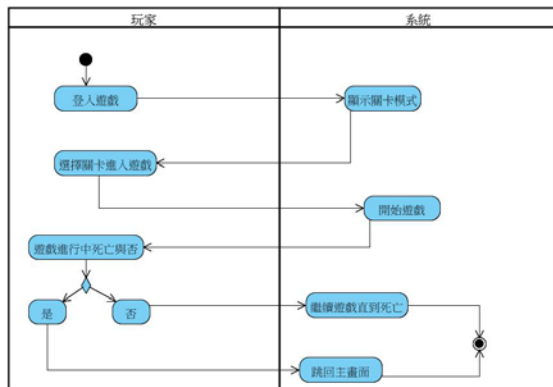


圖 2 玩家活動圖

玩家自登入遊戲後，就可以選擇人物還有想

要玩的機型，接著就可以選擇關卡然後進行遊戲直到死亡。

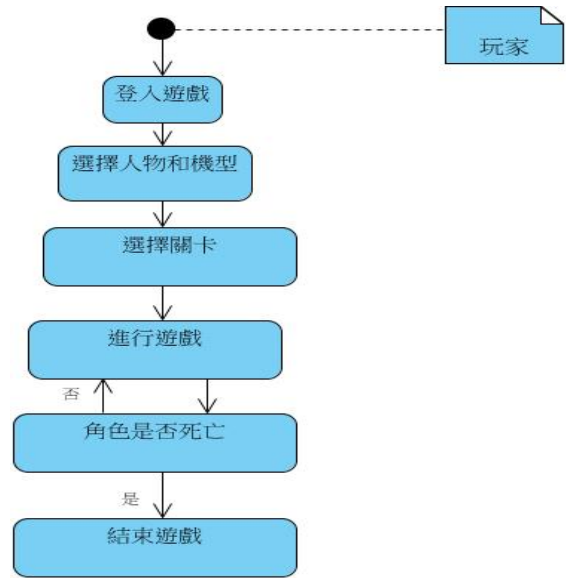


圖 3 玩家的狀態圖

在雲端網站我們還會有介紹遊戲架構、遊戲的基本介紹等等，還有資料存取的地方。

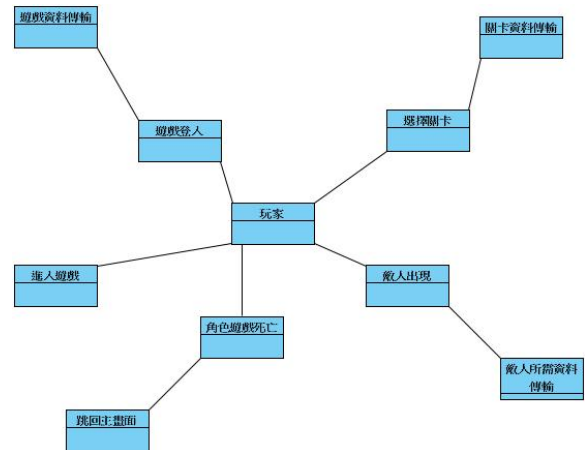


圖 4 玩家的物件圖

玩家可以選擇關卡、登入遊戲、進入遊戲、角色死亡和遇到敵人等等類型

每個資料都包含著公司的整個資料庫等等

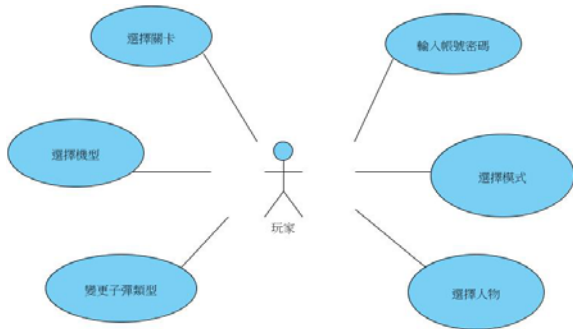


圖 5 玩家的使用案例

玩家要玩這款遊戲必須 要有以下動作，登入遊戲、選擇關卡、選擇模式、人物、機型、變更子彈類型等等，每項功能又包含著子功能

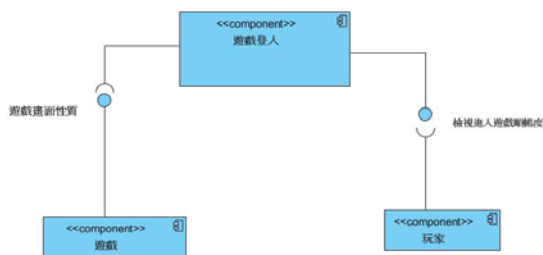


圖 6 玩家的元件圖

遊戲登入要有遊戲和玩家連接，玩家需要輸入資料，資料會傳回總公司，總公司會把玩家所需的資料再回傳給玩家

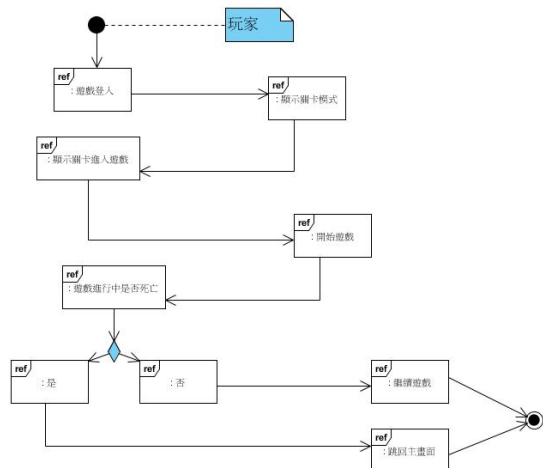


圖 7 玩家的互動圖

玩家從遊戲登入，選擇關卡，再進去遊戲開始一直到死亡後才會結束，這些資料都會直接傳回總公司，總公司也會將玩家接下來所需的資料或者畫面傳回給玩家

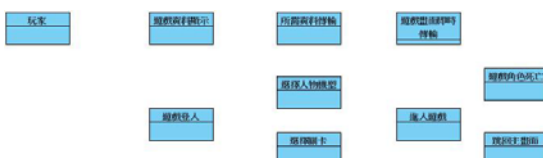


圖 8 玩家的類別圖

玩家可以先登入遊戲然後開始選擇關卡，系統會顯示一些資料以及傳輸

，玩家可以選擇角色機型等等類型進路遊戲，也可以利用雲端來瞭解這款遊戲的資料等等

4. Android 實作

遊戲開始，出現敵機(藍色)還有本身控制的紅色飛機

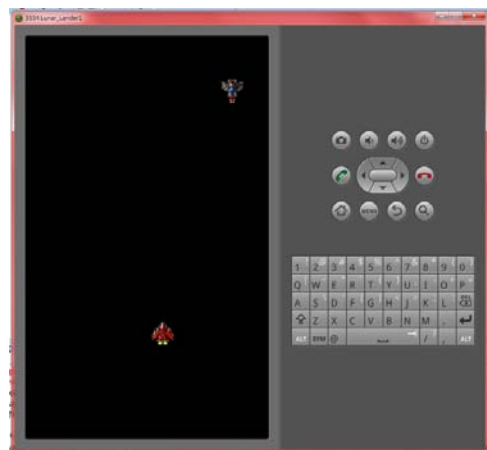


圖 9 遊戲開始

敵機靠近，我方射出子彈，射中敵機，產生爆炸畫面，以及另一隻敵機即將擊中。未擊中敵機，從身邊飛過，我方射出三發子彈，敵機靠近我方本身飛機，我方飛機即將碰撞敵機，碰撞後，我方死亡，我方產生爆炸畫面出現三架敵機，我方射出子彈對付。

敵機兩架遭受我方聚烈攻擊死亡，剩餘一架，敵機大量出現，我方來不及對付我方來不及對付遭受撞擊死亡。

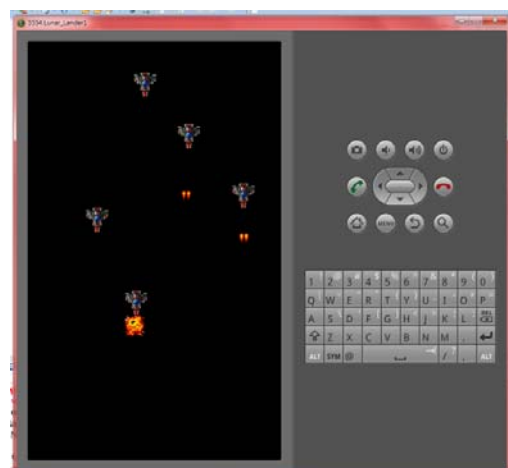


圖 10 遊戲過程

5. 雲端網站

先將 TOMCAT 打開, 設為 Started, 若為 stopped 就沒辦法將雲端網站開啟, 會顯示無法連線或者葉面無法顯示。雲端網站介面, ip 位址為 127.0.0.1, 這是通用的 ip, 任何一台電腦都可使用, 畫面左邊是遊戲介紹, 而右邊則是操作方法。

第一關 zac 魔王介紹, 裡面有介紹魔王怎麼產生的, 是來自於哪裡哪個國度, 還有他們每個魔王的口頭禪。



圖 11 第一關 zac 魔王介紹

第二關卡力斯魔王介紹, 裡面有介紹魔王怎麼產生的, 是來自於哪裡哪個國度, 還有他們每個魔王的口頭禪。



圖 12 第二關卡力斯魔王介紹

第三關弗力貝爾魔王介紹, 裡面有介紹魔王怎麼產生的, 是來自於哪裡哪個國度, 還有他們每個魔王的口頭禪。



圖 13 第三關弗力貝爾魔王介紹

介紹我們的子彈類型, 有包括”巴爾幹砲、雷射、紫色閃電、小型核彈、追蹤飛彈以及炸彈等等”。



圖 14 子彈類型

核彈、追蹤彈類型介紹, 介紹子彈的圖案還有它的功用及效果, 炸彈類型介紹, 介紹子彈的圖案還有它的功用及效果。光速砲型介紹, 介紹子彈的圖案還有它的功用及效果。



圖 15 炸彈類型

6. 覆蓋範圍分析

進行雷電射擊遊戲時, 玩家會進行攻城掠地, 極力擴展其勢力範圍, 因此本研究即進行覆蓋範圍分析。

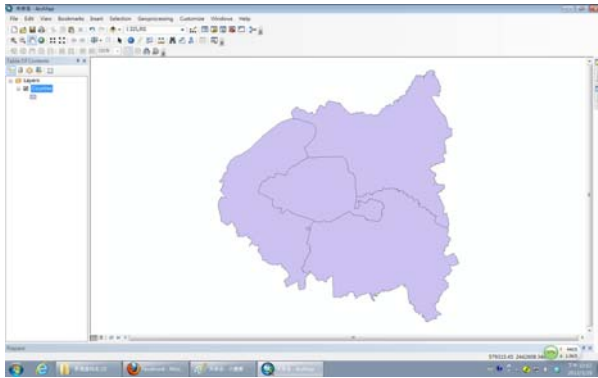


圖 16 開啟圖資

模擬環境

節點散布範圍：1000 公尺 * 1000 公尺

節點分佈模式：隨機亂數產生的節點

節點傳輸半徑 R：180 公尺

節點數：探討節點數分別為 50 點、100 點、200 點、300 點時的情形

要求覆蓋率：分別為 50%、70%、90%、95%、100%時的情形

將資料匯入後會出現一個數據，將那數據點右鍵後會出現面積數據

面積數據出現

Dominating set Algorithm 做部分節點覆蓋

選擇 CH 找出 Dominating set 的演算法:

//自己愈自己半徑 R 內的節點，選一個半徑 R 內的節點數最多的為 CH

For i=0,...,size-1,do

設自己(節點 i 為 max)

For j=0,...,size-1,do

//與自己鄰近節點 j 比較，如果 j 半徑 R 內的節點數比 max 多則把 j 設為 max 值(max=j)。

//如果半徑 R 內的節點數相同時，則以選為 CH 的 node 設為 max 值=最後所得的 max 值(就是節點 max)，挑選出為 CH 程式，跑完可得一 Dominating set，Dominating set 是一組可達 100%

覆蓋的點集合。

利用 Dominating set 演算法，來做節點部份覆蓋的問題，節點先執行過 Dominating set 演算法後，利用所挑的 Dominating set 再來挑戰限制的點數以求最大覆蓋率。

表 1 Dominating set 演算法之覆蓋率 50%

演算法 節點數	覆蓋率50%	
	所需 節點	實際 覆蓋率
50	3.6	54.8
100	4.1	54
200	4.1	52.75
300	4.1	52.233

覆蓋率 70%、90%、95%、100%之計算亦比照本方式，此處省略。

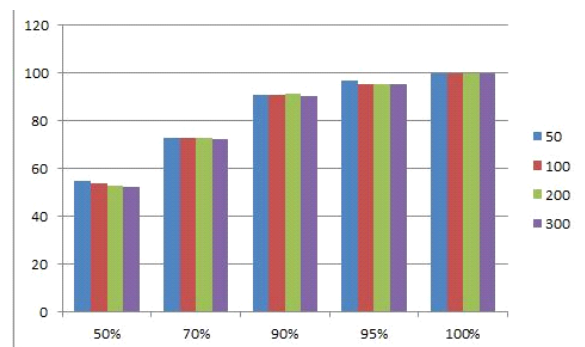


圖 17 Dominating set 演算法之覆蓋率比較

6. 結論

藉由這次的研究，讓我們更加了解，原來從小到大陪著我們成長的遊戲其實並不是那麼容易完成的，期間需要經過無數次的修改及調校，讓遊戲的完整性更高，並達到消費者所需要的那種觀感及感受，才能讓大家一而再，再而三的玩下去，並在完成遊戲之後還需要做出雲端網站以及許多分析等等的計算，並且透過演算法來完成最後的覆蓋率計算，用以達到最後的結果。讓這款遊戲會隨著科技的進步跟著進步，好的遊戲就不會被淘汰。

參考文獻

- [1] 陳泓叡-"行動雲端應用程式中運算及儲存資源運用之探討", 2012
- [2] 林敬硯-"行動雲端個人應用服務之研究", 2012
- [3] 郭宏志-"深入淺出Android實作之無線點餐系統、多媒體、遊戲、GPS與網路應用程式開發", 2009
- [4] 魯旺-"應用同形保密技術於社群適地性行動雲端服務 之研究", 2008
- [5] 黃志安-"台灣行動雲端服務的個資安全探討", 2010
- [6] 顏春煌-"行動與無線通訊，金和資訊股份有限公司", 2005
- [7] 楊文誌-"Android程式設計與應用，旗標出版股份有限公司", 2009
- [8] 楊文誌-"Google Android2程式設計與應用", 2010
- [9] 孫宏明-"Android 2.X手機程式設計入門、應用到精通，基峰資訊 "，2011
- [10] 林城- "Google Android 2.X應用程式開發實戰，基峰資訊"，2010