



逢甲大學學生報告 ePaper

報告題名：基本遊戲類型架構研究與製作
-以 ActionScript3.0 為例

作者：謝武宏

系級：應用數學所

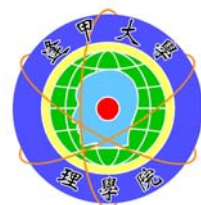
學號：M9618592

開課老師：黃新峰

課程名稱：多媒體技術與應用(二)

開課系所：應用數學所

開課學年： 98 學年度 第 2 學期



中文摘要

遊戲的類型可分成很多種類型，如動作類型、益智類型、冒險類型、戰略類型、RPG 類型、競速類型…等等，但是傳統主要類型為動作類型、冒險類型、益智類型、競速類型等四類。本研究將以 ActionScript3.0 程式語法，探討關於此四種基本類型遊戲的模式架構與製作方法。



關鍵字：互動式遊戲、ActionScript、Flash

目 次

1 緒論.....	3
1.1 研究背景與動機.....	3
1.2 研究目的.....	3
2 文獻探討.....	3
2.1 ActionScript 概述.....	3
2.2 ActionScript3.0 與其它程式語言的差異.....	4
2.3 基本遊戲類型與精神.....	5
2.4 遊戲基本元素.....	5
3. 研究流程與研究方法.....	6
3.1 遊戲模式架構.....	6
3.2 遊戲重點設計.....	8
3.3 問卷建構.....	12
4. 結論與建議.....	14
參考文獻.....	16



壹. 緒論

1. 研究背景與動機

隨著科技發展日新月異，電子遊戲內容也不斷的在改變，從傳統的單機遊戲，如 Family Computer(俗稱紅白機)的「小精靈」或「小蜜蜂」，到個人電腦的線上遊戲，如遊戲橘子公司所開發的「天堂」線上遊戲或「跑跑卡丁車」線上遊戲，其遊戲製作與遊戲結構都有所不同。

在每一個電子遊戲中，遊戲類型結構不一定會相同，有些遊戲偏向益智類型，如猜謎遊戲或是棋藝遊戲等益智遊戲，其遊戲精神在於訓練玩家們的思考能力，也有些遊戲偏向動作類型，如球技競賽遊戲或射擊遊戲，其遊戲精神則是訓練玩家們的反應能力。此外，除了遊戲類型結構不一定相同之外，其遊戲畫面也會有所差異，如傳統的 2D 平面遊戲以及耳熟能詳的 3D 立體遊戲，其遊戲畫面所帶給玩家的視覺感受也各有所不同。

從 1961 年電子遊戲誕生到現在，可以製作遊戲的軟體多不勝數，其中 Adobe Flash CS4 Professional 套裝軟體內建的繪製功能，與其支援的 ActionScript3.0 指令撰寫功能，可以為我們規劃出遊戲的藍圖，來設計出想要製作的遊戲類型，不論是屬於靜態遊戲或是動態遊戲，甚至是利用聲音來進行遊戲，都可以由 Adobe Flash CS4 Professional 套裝軟體製作出來，再藉由製作遊戲的過程與結果，來了解各種互動式遊戲的差異性。

2. 研究目的

市面上互動式遊戲琳瑯滿目，每一種遊戲都有它獨特的特色與精神，且在不同的年齡層中，也各自佔有一片天空，例如 FaceBook 網頁中的開心農場，或是各遊戲公司所代理的線上遊戲，又或者是大家耳熟能詳的多人麻將，彼此之間都有著不同的玩法與趣味。因此，本研究以製作基本類型遊戲的方式，來分析探討遊戲本質應該具備著何種精神與特色，才會使得遊戲是受人喜愛的。希望能藉此研究來幫助以 Flash 作為遊戲開發工具的遊戲人才，更進一步了解廣大民眾對於遊戲的期望與要求。

貳. 文獻探討

1. ActionScript 概述

ActionScript 前身出現於 Flash Player 4 中，用於控制 Flash 內容，並進行簡單的互動性腳本編寫。它並不複雜，其語法和語意與 ECMAScript 也不相同。ActionScript1 是在 Flash 5 的時代誕生，這時的版本就已經具備了 ECMAScript 標準的語法格式和語義解釋。隨著 Flash Player 5/6 的版本更新，越來越多的 ECMA 語法與語義被納入到 ActionScript1 的 API 當中去。但是核心語言的編譯處理及表現方式都是延續了 Flash 5 的 ActionScript1 的標準。甚至於到了後來的 Flash7，它裡面的 ActionScript1 的核心解釋機制仍然與 Flash Player 5 有著很多相同之處。

ActionScript2 可以看成是 ActionScript1 的物件導向程式設計包裝版。ActionScript2 的程式碼編寫引入了物件導向程式設計的方式，有良好的型別宣告，而且分離了執行時和編譯時的例外處理。但是，ActionScript2 只是在編譯階段支援這些 OOP 語法，在執行時仍然使用的是 ActionScript1 的模式。因此，ActionScript2 和 ActionScript1 在性能上不會有本質的區別。雖然它們在格式上遵從了 ECMA4 的語言方案，但是並不真正相容 ECMAScript 標準。

Flash Player 和 ActionScript 的開發團隊體認到發展現有引擎，將有著太多限制而無法繼續研發新功能。經過這些可敬的人們長期不斷的努力後，Flash Player 的歷史轉捩點到來了。Flash Player 9 版本在 2006 年首次引入了 ActionScript3 和新一代的 ActionScript 引擎—ActionScript Virtual Machine 2 (AVM2)，AVM2 是一個新的、高度最佳化虛擬機，顯著超越了 AVM1 可能達到的性能。ActionScript3 基本上是 ActionScript 引擎的完全覆寫，在 AVM2 中執行可達到前所未有的高效率和高性能，執行速度最多可比原有 ActionScript1/2 程式碼快 10 倍。

2. ActionScript3.0 與其它程式語言的差異

物件是 ActionScript3 語言的核心要素，也是基本的建構單元。所宣告的每一個變數、所撰寫的每一個函數，以及所建立的每一個類別實體都是物件；可將 ActionScript3 程式視為一組執行工作的物件，這些物件會回應事件，並且彼此進行通訊。熟悉 Java 或 C++物件導向程式設計(OOP)的程式設計人員可能會將物件視為模組，其中包含兩種成員：儲存在成員變數或屬性的資料，以及可透過方法存取的行爲指令。ECMAScript 第 4 版草稿是 ActionScript3 所依據的標準，其定義物件的方式與此類似，但稍有差異。在 ECMAScript 草稿中，物件只是屬性的集合；這些屬性是容器，不但能保存資料，也能保存函數或其它物件。若函數是以這種方式附加到物件上，則稱為方法。

ActionScript 中的類別與 Java 或 C++中的類別之間有一項微妙的差異，就是在 ActionScript 中，類別不僅只是抽象實體。ActionScript 類別是以儲存類別之屬性和方法的「類別物件」來代表，因此能夠運用 Java 和 C++程式設計人員可能會覺得性質很不相同的技巧，例如，在類別或套件的最高階包含陳述式或可執行的程式碼，且每一個 ActionScript 類別都有所謂的「原型物件」。在舊版 ActionScript 中，原型物件連結在一起形成「原型鏈」，而整體做為整個類別繼承階層的基礎；但是在 ActionScript3 中，原型物件在繼承系統中只扮演份量很輕的小角色，原型物件仍然會很有用，若要在類別的所有實體間共享屬性及其值，它可以替代靜態屬性和方法。

3. 基本遊戲類型與精神

動作類型(Action)

動作類遊戲都有移動的物體，它主要為挑戰玩家爭取高分的反應、手眼協調能力、快速思考和時機掌握等能力的類型設計，藉由考驗個人的敏捷與反應，來換取更多遊戲分數或能力見證。

冒險類型(Adventure)

冒險類遊戲裡，玩家在遊戲環境中操控一個角色，其主要目的是通過重重的障礙或敵人，並試圖到達目的地以表示過關，其中「超級瑪莉」就是冒險類遊戲的經典範例。

競速類型(Racing)

很單純地已達到領先對方為目標，從原始講究真實的實感賽車，而後發展至能以賽道上獲得特殊道具以干擾對手的偏益智型趣味賽車，如「瑪莉歐賽車」、「跑跑卡丁車」等遊戲皆屬此類。

益智類型(Puzzle)

益智遊戲也被稱為邏輯遊戲，它可以說是最受一般玩家歡迎的休閒遊戲，主要是挑戰玩家智慧的遊戲，它注重的是玩家的思考與邏輯判斷，在遊戲中不需要以快節奏的方式呈現，而是將重點放在多樣變化的解題路徑上。

4. 遊戲基本元素

無論哪種遊戲類型，一個豐富好玩的遊戲，一定會具備著好玩的遊戲元素，才能讓玩家一再的想要遊玩，而這些遊戲元素也會依據遊戲內容的不同，其包含的元素種類也會有所不同，以下為最基本的遊戲元素。

計分(Score)

計分制在遊戲裡是玩家功力的一種評量標準，可以用來與朋友互相比較或競爭，當玩家執行任務中表現好或敢於冒險時，給予分數的獎勵，在遊戲結束後，將顯示分數讓玩家明白實力到達哪一種程度。此外，分數的競爭方式，會更增加遊戲的可玩性，因為畢竟每個人都喜歡當第一名。

難度(Level)

也許製作出來的遊戲讓人覺得平凡無奇，但若是將遊戲難度藉由關卡增加或時間拉長而逐漸升高，便會讓玩家沉醉於其中，並試圖突破關卡，如此一來遊戲的耐玩度就會更加提高。在難度的設定上，必須因循漸進，讓在一開始就將難度調得過高，那麼玩家將會因挫折感而直接放棄遊戲，相對的，若遊戲中每個關卡的難度都過於簡單，那麼遊戲將會失去新鮮感，而不會再次遊玩。

體力/生命(Life)

生命可以控制遊戲的時間，並可以讓玩家有多層的機會進步。遊戲中運用生命的方法很多，像是人物打鬥的體力表，或是一次斃命的死亡次數，這都是提供玩家可以失敗幾次的方法。一般遊戲中給予玩家的生命次數約為三次，這是大家最習慣的次數，也是遊戲時間的理想長度。

遊戲時間長度(Time)

大多數網站遊戲的成人玩家，都是趁著工作中休息片刻的時間來玩玩遊戲，提振一下精神，讓腦筋清醒一下，而青少年玩家則只能在放學回家後的休息時間，短暫地玩個遊戲就必須準備複習課業，因此，若遊戲的時間長度太過於冗長，反而會讓人連想玩的動力都沒有。

音樂/音效(Sound)

一個遊戲中，若沒有的背景音樂或事件觸發的音效，那麼就相當於是一個沒有靈魂的人，雖然遊戲中的各種條件都兼具，卻獨獨缺少了音樂以及音效，反而會讓人無法完全投入於遊戲。此外，音樂與音效在遊戲中的搭配也很重要，相信任何一個玩家都無法想像在驚悚恐怖的遊戲中，卻聽到充滿可愛風格的背景音樂吧。



參. 研究流程與研究方法

1. 遊戲模式架構

貪食蛇-動作類型(Action)

廣泛的動作類型遊戲中，其注重的遊戲精神與目的不一定相同，本研究中，以手眼協調能力為重點，製作廣為人知的「貪食蛇」遊戲，當作動作遊戲類型的研討目標。

在遊戲中，玩家操控一條細長的直線（俗稱蛇或蟲），它會不停地前進，玩家只能操控蛇的頭部往上、下、左、右四種不同方向移動，並試圖去吃到散落在地圖畫面上的食物，而在移動的過程中，還要避免觸碰到自己的身體或者其他障礙物。每次貪食蛇吃掉一件食物，它的身體便增長一些，讓遊戲的難度漸漸變大。

以往貪食蛇遊戲的規格模式，是以方塊類型製作而成，在封閉的小房間中，主角貪食蛇會以格子式的走路方法，四處遊走尋找食物，以增加自己身體的長度而得到分數。本研究中，更換貪食蛇傳統的方塊行走模式，而採用不會停頓的連續行走模式，讓貪食蛇在行走時，不會出現停頓的畫面。

在遊戲的基本元素調配中，將貪食蛇遊戲設計成單一難度的遊戲，探討難度(Level)在遊戲中的重要成份。

飛越大峽谷-冒險類型(Adventure)

突破重重的危險，讓自己生存下來，是冒險類型遊戲最顯著的特性，本研究中，將以此特性製作出一個駕駛飛機在大峽谷中冒險飛行的遊戲，當作冒險遊戲類型的研討目標。

玩家扮演一位飛機駕駛者，在險峻的大峽谷與頻繁的飛彈攻擊中，試圖駕駛飛機穿越過重重的險境，讓自己生存下來。遊戲中的遊戲畫面，以捲軸畫面的方式行進，玩家無法猜測出下一次的大峽谷地形為何，並且將飛彈以鎖定目標的方式直線前進，讓玩家無法安穩的在地圖中飛行。

在遊戲的基本元素調配中，將飛越大峽谷遊戲設計成單一生命的遊戲，探討生命(Life)在遊戲中的重要成份。

雪原賽車-競速類型(Racing)

快速的領先對方，將周邊一切的人、事、物通通拋於腦後，追求極致的速度感，這就是競速類型的主要特色，本研究中，將以突顯速度感的製作方式，來製作探討競速類型的遊戲。

玩家扮演一位試圖超越電腦賽車駕駛者的賽車駕駛者。遊戲中，一開始電腦會以極快的速度超越玩家，玩家必須在充滿水灘或巨大雪球侵襲而來的雪道上，克服萬難的追趕上電腦並且超越過它。遊戲中玩家駕駛過一定的距離後，賽車的速度就會獲得提升，相對的，水灘與雪球出現的次數以及速度，會也因賽車本身速度的加快，而變得頻繁以及加快撞擊的速度。

在遊戲的基本元素調配中，將雪原賽車遊戲設計成單獨追逐電腦的遊戲，探討遊戲時間長度(Time)在遊戲中的重要成份。

微笑迷宮-益智類型(Puzzle)

動一動腦筋，思考著如何通過這個關卡，是益智類型遊戲主要傳達的特色，本研究中，以迷宮當作遊戲藍圖並加上不同障礙陷阱的模式架構，當作探討益智類型遊戲的目標。

將傳統的迷宮遊戲，結合人工智慧的陷阱，就是微笑迷宮的基本結構。遊戲中，玩家扮演一個試圖闖過重重的陷阱，而到達迷宮出口的微笑臉，每個陷阱的特殊功能皆有不同，有的會擊破微笑臉，有的會將微笑臉牽制住，有的會將微笑臉傳送到其他關卡並重新來過，而每個陷阱的觸發時間都有所不同，玩家必須推導出正確的通過時間，才不會掉入陷阱。在遊戲的結尾中，安排了隱藏關卡(Hidden Level)供玩家嘗試。

在遊戲的基本元素調配中，加入所有遊戲基本元素，藉此與上述遊戲比較，完整與不完整的遊戲基本元素所影響遊戲好玩度的差別。

2. 遊戲重點設計

貪食蛇-連續行走模式

傳統貪食蛇遊戲的行走模式是以格子式行走，其作法是將身體的座標點，利用傳遞座標點的方式傳達給下一個身體，以達到行走的效果，如圖 1，但是如此一來在遊戲畫面上的效果，就沒有正常行走時的連續動作，而是以跳格子的方式行進，如圖 2。

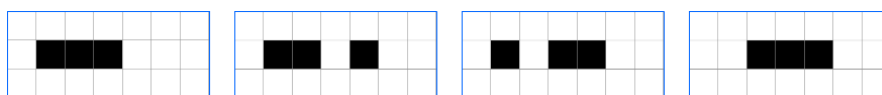


圖 1. 貪食蛇往右移動的行走模式

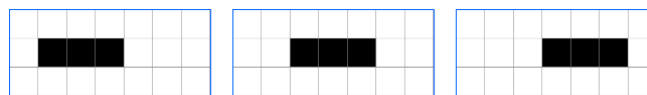


圖 2. 貪食蛇格子式行走模式

本研究中，將傳統貪食蛇的格子式走法，改編成連續動作。首先假設貪食蛇每一個身體都有它專屬的名字 $N(\text{name})$ ，而貪食蛇的頭與第一個身體分別定義為 N_1 、 N_2 ，接著定義 $V(\text{velocity})$ 為貪食蛇的移動速度； $R(\text{rotation})$ 為貪食蛇在遊戲場景中的移動方向； $D(\text{distance})$ 為貪食蛇頭與身體之間的距離差； $M(\text{maximum})$ 為頭與身體之間允許的最遠距離，如圖 3。我們預設了兩種情形：

- (1) 如果 N_1 沒有轉向，則表示 N_1 與 N_2 的移動方向相同，且 $D < M$ ，因此讓每一個身體都以基本速度 V ，往相同的 R 移動，如圖 4。
- (2) 如果 N_1 已經轉向，則表示 N_1 與 N_2 的移動方向不同，且因為移動方向的不同，讓原本在同一直線上的 N_1 中心點與 N_2 中心點，兩者的關係變為三角關係，如圖 5，使得 D 一開始會慢慢地減少，接著會慢慢的增加，如圖 6，當 $D > M$ ，則表示 D 已經超出合理的範圍，將 $R_{n2} = R_{n1}$ ，讓頭與身體往相同方向移動，並將相關位置補正，以完成轉向動作，如圖 7。

由於每一個身體本身都以 V 來進行移動動作，而不是以記錄上一個身體座標點的方式移動，因此貪食蛇移動時不會停頓，而是連續行走動作。

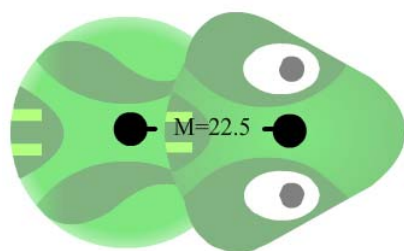


圖 3. 貪食蛇的頭與身體的最遠距離

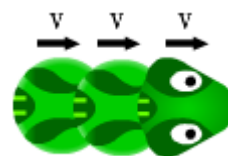


圖 4. 貪食蛇未轉向的移動情況

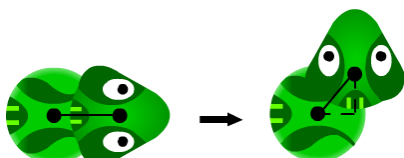


圖 5. 中心點由線性關係轉為三角關係

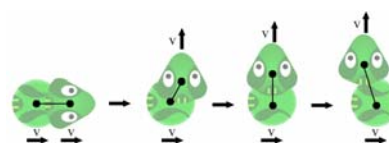


圖 6. D 由小變大

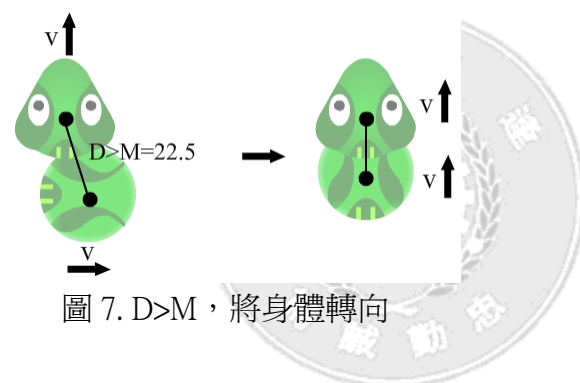


圖 7. $D > M$ ，將身體轉向

飛越大峽谷-隨機地圖選取

經典的戰機遊戲「雷電」，如圖 8，其遊戲畫面是以捲軸方式慢慢呈現出來，而本遊戲的畫面呈現方式，也是以捲軸方式呈現。由於本遊戲設定飛機在沒有墜毀的情況下，可以一直飛行以累積分數，所以遊戲地圖必須不斷的更新，才不會讓玩家在相同地圖上重覆遊玩而感到厭煩。由於遊戲地圖必須以隨機的方式呈現給玩家，因此在一開始遊戲地圖設計上，就無法以人工編排的方式，事先將地圖編排而成，而且這樣的方式既耗時耗力，也無法讓玩家開啓遊戲後，會有遇到不同遊戲地圖的新鮮感。



圖 8. 「雷電」遊戲畫面

在地圖編排上，定義了 8 個地圖(map)為一個集合 S(set)

$S=[map1, map2, map3, map4, map5, map6, map7, map8]$

當玩家開啓遊戲後，定義 N(number)為地圖編號，利用虛擬亂數值

$Math.random()$ 來決定 N 取得的數字，也就是 $N= Math.random()$ ，再由 N 來決定抓取哪一張地圖，我們給定了 4 種步驟:

- (1) 抓取隨機數字， $N= Math.random()$ 。
- (2) 將 N 視為 S 集合的元素編號，以決定抓取的地圖，如 $N=3$ 時，決定抓取「map3」。
- (3) 將抓取過的地圖刪去，以避免重覆抓取相同的地圖，如「map3」被抓取過，則將「map3」在 S 集合中移除。

$S=[map1, map2, map4, map5, map6, map7, map8]$

- (4) 當抓取過第一張地圖後，開始重覆(1)、(2)、(3)步驟，並將最初刪去的地圖重新補上，讓地圖永遠都有 7 種選擇，並且不會選取到已選過的地圖，假設第一次抓取到的 $N=3$ ，並將「map3」放置在遊戲場景中，再從 S 集合中移除

$S=[map1, map2, map4, map5, map6, map7, map8]$

而第二次抓取到的 $N=5$ ，並將「map6」放置在遊戲場景中，再從 S 集合中移除

$S=[map1, map2, map4, map5, map7, map8]$

接著在將原本刪去的「map3」放入 S 集合裡，供第三次地圖選擇

$S=[map1, map2, map3, map4, map5, map7, map8]$

重覆(1)、(2)、(3)、(4)步驟，就可以無限抓取出新的隨機地圖。

雪原賽車-3D 視覺模式

競速類型的遊戲畫面，有 2D 平面畫面，也有 3D 立體畫面，兩種遊戲畫面帶給玩家們的視覺感受也會有所不同。由於 Flash 軟體沒有直接支援 3D 畫面的功能，因此在本研究中，將利用程式與向量圖的繪製，來製作出 3D 立體效果。

當我們在看遠處的物體時，因為距離的關係會覺得物體比實際高度小許多，而拉近距離後，物體又會變大許多，因此我們利用遠近的視覺反應，來讓原本只有 2D 平面的遊戲畫面產生 3D 視覺效果。

首先定義物體座標的線性方程式 L(linear)當作物體的行進軌道，以及物體的實際大小比率 S(scale)與透明度 A(alpha)。假設遊戲場景如同平面座標圖，並以定義出物體在平面座標上的線性方程式 L，如圖 9，而當物體在遠處時，物體與實際大小的比率會較小，並且比較模糊，若物體在近處時，則物體與實際大小的比率較為一致，並且比較清晰可見，如圖 10

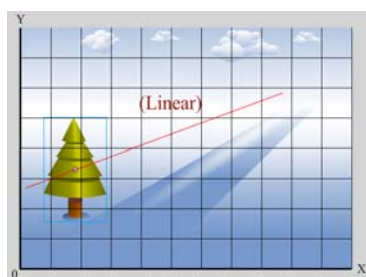


圖 9. 遊戲場景平面座標圖



圖 10. 物體遠近的視覺效果

因此我們在物體的 S 與 A 的設定上，以物體在平面座標中距離原點的遠近而定，若物體距離原點較近，則將平面狀態轉換成空間狀態來看的話，物體則是距離駕駛者較近，相反地，若物體距離原點較遠，則將平面狀態轉換成空間狀態來看的話，物體則是距離駕駛者較遠，因此以距離駕駛者最近物體的 S 與 A 當作初始值比率 100% 來看，距離駕駛者越遠的物體，其 S 與 A 的比率值就越小，如圖 11，以物體的遠近來控制其大小比率與透明度，進而製作出 3D 空間的視覺效果。

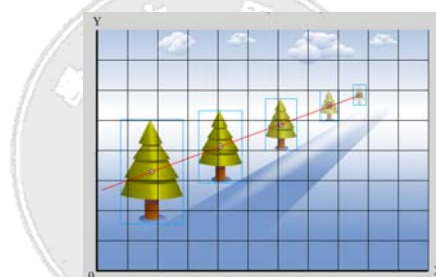


圖 11. 物體在遊戲場景中的比率

微笑迷宮-時間軸設定

將原本單純的障礙擺設，加上了觸發時間控制，就形成了讓玩家傷透腦筋的麻煩陷阱。以視覺來說，有形的事物容易被預測出來，如障礙物的大小或位置，但無形的事物卻是令人捉摸不定，本研究中，將有形的障礙物加上了無形的隨機觸發時間，讓玩家無法預測出下一次陷阱的觸發時間，若玩家遇上了兩種以上的陷阱，其隨機觸發時間各有不同，更會讓玩家在原地躊躇不前，思考著該何時前進才不會誤觸陷阱。

首先將陷阱的隨機觸發時間分為三個部份，第一個部份為 I(initiation time)，用來表示陷阱下一次觸發的當下時間點，第二個部份為 D(duration)，用來表示陷阱觸發後的執行所需時間，第三部份為 R(random duration)，用來表示陷阱距離下一次再次觸發的隨機所需時間。

假設遊戲本身時間為 G(gameTimer)，我們將陷阱隨機觸發時間分為四種情況：

- (1) 計算 I 值， $I=D+R$ ，如圖 12，其中每種陷阱所提供的 D 時間長度不一，而遊戲每次提供給陷阱的 R 值也不盡相同。
- (2) 讓陷阱在 D 所提供的時間內執行完成。
- (3) 當陷阱執行結束後，給定 R 時間長度，讓陷阱在這段時間內冷卻不觸發。
- (4) 當 G 大於 I 值後，如圖 13，重覆(1)、(2)、(3)步驟。

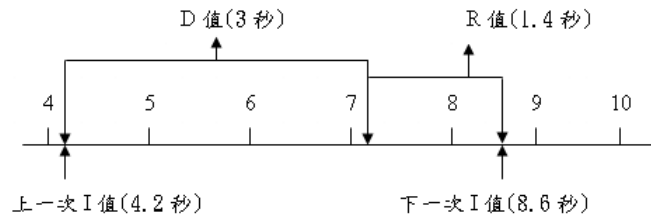


圖 12. I 值計算的時間軸表示

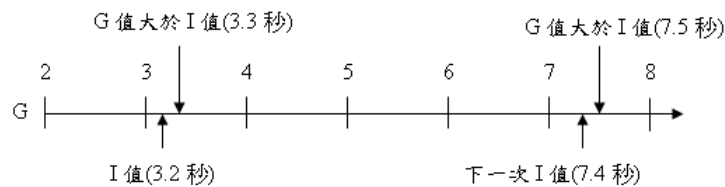


圖 13. G 值與 I 值的時間軸關係

3. 問卷建構

爲了了解現在大學生對於遊戲基本元素的需求，我們以逢甲大學應用數學系大學部學生爲主要問卷調查對象，將問卷項目分爲三大部份，分別爲「問題」、「建議」與「其他意見」，總共有 15 個項目，如圖 14。

在「問題」的部份中，主要針對遊戲在製作上是否有程式漏洞、畫面不協調、音樂以及音效不符合遊戲內容、電腦操作時是否流暢等問題，而在「建議」的部份，則是針對受訪對象所喜歡的遊戲風格爲何，而在「其他意見」的部份，則是讓受訪對象補充上述問題項目所沒有的意見，更能加強改進往後遊戲在製作上的品質。

遊戲名稱					
題號	問題	很同意	同意	不同意	很不同意
1.	整體遊戲的畫面設計精緻				
2.	遊戲的玩法淺顯易懂				
3.	遊戲執行時，電腦運作十分流暢				
4.	主角操作方法簡單易懂，且操作時不會停頓				
5.	本遊戲對我而言是有趣的				
6.	遊戲中的音樂與音效符合遊戲性質				
7.	遊戲進行時，沒有重大的錯誤或 Bug				
8.	本遊戲難度多樣化				
9.	遊戲中的體力/生命符合需求				
	建議	是		否	
10.	我希望遊戲畫面是 3D 立體畫面				
11.	我希望遊戲進行的節奏緊湊				
12.	我希望遊戲多人競賽				
13.	我希望遊戲風格為可愛風格				
14.	我希望遊戲有許多障礙，以阻礙玩家過關				
15.	我希望遊戲難度越難越好				
16.	我覺得遊戲整體分數為 (0-10)				
其他意見:					

圖14. 問卷格式

貪食蛇-難度(Level)分析

在問卷調查中，對於遊戲難度是否多樣化的問題部份，有 74%的比例表示必須要有難度的變化，明白顯示出貪食蛇遊戲在難度的調配中，單一的遊戲速度、沒有障礙物、遊戲過於單調平凡等問題，讓玩家玩過幾次遊戲後，失去遊戲的新鮮感。

在「建議」與「其他意見」中，在「我希望遊戲多人競賽」(94%)、「我希望遊戲風格為可愛風格」(77%)、「我希望遊戲有許多障礙，以阻礙玩家過關」(94%)等意見部份中，其比例高於 75%以上，明顯看出玩家對於此類型遊戲，希望以多人比較的形式在可愛風格的遊戲畫面中進行，並且喜歡在遊戲中挑戰障礙難度，讓遊戲在進行中多點變化。

飛越大峽谷-體力/生命(Life)分析

在問卷調查中，對於遊戲生命是否符合需求的問題部份，有 81%的比例表示不符合需求，明白顯示出玩家在難度較高的遊戲中，希望能有多次嘗試的機會，以避免因為爭取不到較高的分數而有挫折感。

在「建議」與「其他意見」中，在「我希望遊戲多人競賽」(90%)的意見部份中，其比例高於 75%以上，明顯看出玩家對於此類型遊戲，希望以多人比較的形式進行，而在其他意見部份中，其比例較沒有明顯的差異。

雪原賽車-遊戲時間長度(Time)分析

在問卷調查中，對於遊戲是否有趣的問題部份，有 79%的比例表示不有趣，明白顯示出玩家在沒有明顯目標的單機遊戲中，長時間重覆相同動作與畫面會失去對遊戲的耐心與新鮮感。

在「建議」與「其他意見」中，在「我希望遊戲畫面是 3D 立體畫面」(75%)、「我希望遊戲多人競賽」(96%)、「我希望遊戲有許多障礙，以阻礙玩家過關」(75%)等意見部份中，其比例高於 75%以上，明顯看出玩家對於此類型遊戲，希望在 3D 視覺畫面中以多人比較的形式進行，並且希望有障礙物以及道具在遊戲中產生，以增加遊戲好玩度。

微笑迷宮-遊戲基本元素分析

在問卷調查中，在這問題項目滿意的程度，皆大於 80%以上，其中在遊戲是否有趣的部份更高達 90%，明顯看出擁有完整遊戲基本元素的遊戲與其它不完整遊戲基本元素的遊戲，其遊戲有趣程度有著明顯的差異。

在「建議」與「其他意見」中，在「我希望遊戲風格為可愛風格」(93%)的意見部份中，其比例高於 75%以上，明顯看出玩家對於此類型遊戲，希望在遊戲畫面中，呈現出較為可愛風格的畫風，而在其他意見部份中，其比例較沒有明顯的差異。

肆. 結論與建議

透過實際遊戲的製作過程與經驗，本研究提供以下幾點建議做為未來遊戲開發時的效能參考：

難度要適中並且有變化

當一個遊戲沒有變化，只是單純的讓玩家重覆做相同動作時，玩家對於遊戲的耐玩度將會大幅下降，因為每個玩家都是為了追求遊戲的刺激性或新奇性，當遊戲無法帶給玩家更新的感受時，就代表遊戲即將結束，以上市遊戲來說，不管是 Single User Game(單機遊戲)或是 Online Game(線上遊戲)，其遊戲內容會不斷的更新，並加強一定的難度，讓玩家隨時保有新鮮感。

給予玩家多次嘗試的機會

如果玩家不小心誤入了陷阱而損失生命，卻因此遊戲結束而必須重頭來過，那麼玩家對於遊戲的接受度，就像玩家在遊戲中的角色生命一樣，如此的短暫，因此讓玩家有多次嘗試的機會，一步步了解遊戲中的優點，而不是因為沒有多次嘗試的機會，讓玩家無法嘗試到遊戲製作者想要帶給玩家的驚喜與優點。

掌握遊戲時間

掌控好時間，可以讓玩家在面對挑戰時，感受到時間緊迫的刺激感，讓遊戲整體加分，相對地，若沒有掌控好時間，則會讓玩家產生設計不良的感覺，例如一個簡單的任務，遊戲製作者卻給予過長的過關時間，或是一個艱難的任務，只給予少許的時間，這都會讓玩家對遊戲產生反感甚至於拒絕再玩。

給予玩家適當的獎勵

當玩家辛辛苦苦過完一個任務後，卻沒有得到應有的分數獎勵，這會讓玩家有白忙一場的感覺，因此當玩家過完一個任務後，必須給予玩家適當的獎勵，這獎勵可以是分數也可以是寶物，只要是能激勵人心的獎勵即可，而在獎勵的調配上，必須與關卡難度一致，讓玩家依循著關卡難度的提升，而得到更好的獎勵，以避免玩家一開始就得到很好的獎勵，或是辛辛苦苦過完任務所得到的獎勵卻是差強人意，這些情況都會減少玩家對於遊戲的支持度。

遊戲在很多人的生活中，已經是密不可分，除了喜愛玩遊戲的玩家，也有部份玩家朝著製作遊戲的方向努力，因此，本研究以製作基本類型遊戲的方式切入研究面，分析探討基本類型遊戲中，所不可缺少的遊戲基本元素 以及讓遊戲失去可玩性的致命缺點為何，幫助遊戲開發者除了在遊戲製作的技術與想法上有所提升外，也能以此研究來做為未來遊戲開發的重點心得，使遊戲開發者能在製作遊戲的方向上，更能得心應手。

本研究相關作品皆已發佈到 <http://140.134.140.68/games> 網站。

參考文獻

- [1] Stefan M. Grünvogel , [volume 5 issue 1 october 2005](#) , Formal Models and Game Design , Game Studies
(<http://www.gamestudies.org/0501/gruenvogel/>)
- [2] Andrew Rollings & Ernest Adams , 2003 , 大師談遊戲設計 , 臺北:上奇科技股份有限公司
- [3] Jobe Makar & Ben Winiarczyk , 2004 , 大師談 Flash 遊戲設計與製作 , 臺北:上奇科技股份有限公司
- [4] 孫穎 , 2008 , Flash ActionScript 3 殿堂之路 , 臺北:松崗電腦圖書資料股份有限公司。
- [5] Dante Treglia , 2005 , 遊戲程式設計精華 III , 臺北:碁峰資訊股份有限公司
- [6] Andrew Kirmse , 2006 , 遊戲程式設計精華 IV , 臺北:碁峰資訊股份有限公司
- [7] 葉思義&宋昀璐 , 2004 , 數位遊戲設計:遊戲設計知識全領域 , 臺北:碁峰資訊股份有限公司
- [8] Keith Peters , 2009 , AdvancED ActionScript 3.0 Animation , friendsof
- [9] Keith Peters , 2009 , Foundation ActionScript 3.0 Animation Making Things Move , friendsof
- [10] Rex van der Spuy , 2009 , Foundation Game Design with Flash , friendsof
- [11] Gary Rosenzweig , 2008 , ActionScript 3.0 Game Programming University , Que Publishing
- [12] 黃新峰&陳子建&葉宇霖 , 2009 , Flash ActionScript 3.0 程式設計 , 臺北:松崗電腦圖書資料股份有限公司。
- [13] 黃新峰&陳子建&葉宇霖&張益銘 , 2009 , Flash ActionScript 3.0 範例應用 20 例 , 臺北:松崗電腦圖書資料股份有限公司。