

在效能受限平台上開發網路遊戲之效能研究

**Develop game depended on Flash and investigate
how to improve performance on a limited platform**

陳光雄	賴志承
Kuang-Hsiung Chen	Chih-Cheng Lai
khchen@ mail.stut.edu.tw	zisaboy@yahoo.com.tw
0918199954	0910395862

南台科技大學 多媒體與電腦娛樂科學研究所
Department of Multimedia and Entertainment Science,
Southern Taiwan University of Technology

在效能受限平台上開發網路遊戲之效能研究

Develop game depended on Flash and investigate how to improve performance on a limited platform

摘要

隨著網路多媒體的逐漸演進，Macromedia 所開發的 Flash 也漸漸從網頁上的動畫特效走入了遊戲娛樂的開發應用之中。許多利用 Flash actionscript 開發的網路遊戲也如雨後春筍般的蓬勃發展，Macromedia 公司花了大量的時間來致力於 Flash player 播放器達到最小的程度，這樣發布在網路上有大多數人都可以很容易得下載到 Flash player 播放器，但缺點卻是犧牲了 Flash 在程式上的執行效能與圖片的更新速度，這使得 Flash 在跟其他遊戲平台的效能比較上來得遜色許多。因此本論文提供一些在 Flash 遊戲開發上可提升效能的方法及模式，以幫助遊戲設計者提昇其遊戲設計實力。

關鍵字：遊戲、效能、Flash、Actionscript。

Abstract

As the progressing of the multimedia in network by degrees, the Flash which was developed by a company named "Macromedia" has gone into the development from special efficacy of animation in network to entertainment in game step by step. Many network games which were created by using of Flash actionscript has developed vigorously. Macromedia spent much time to devote themselves to smaller their capacity of their program which was named Flash player. Many people can download the Flash player easily by the network. A defect is that they sacrifice the executed efficiency and the update speed of Flash, and this made the efficiency of Flash is much inferior than other developing software in game. Therefore this paper provides some referable frameworks and modes in development of Flash games to improve the performance. To help designers to improve their technology on designing game。

Keywords：game、performance、Flash、Acionscript。

1. 前言與動機

從 1995 年 Macromedia 公司推出的第一套 FutureSplash Animator 開始至 2005 年的 Flash 8 為止，Flash 在互動創作與視覺藝術上帶給全世界廣大的使用者無數的驚奇與讚賞，時至今日，Flash 的多媒體應用除了以往的動畫效果外，也朝著數位娛樂的方向發展。一些精緻的 Flash 遊戲，如：神影無蹤-廖添丁；或者是 Flash 大型多人連線的線上遊戲，如：DoFus-MMORPG，都顯示出 Flash 在互動遊戲世界開發的可能性。

Flash 平台開發遊戲的優點在於它使用向量運算的方式，所以製作出來的遊戲體積會比其他遊戲開發平台小很多，且 Macromedia 公司宣稱全世界 97.3% 的網路瀏覽器都有內建 Flash player 播放器[1]，如此一來對遊戲的散播性有很好的幫助。同時，程式設計師與美術人員可以彼此共享相同文件檔案。最後 Flash 簡單易用，對於初學者而言，如果使用其他程式語言開發遊戲可能會比使用 Flash 所需開發時間還要來得緩慢。不過 Flash 也存在部分缺點，首先它缺乏對三維空間(3D)的支援，且不能很容易的來操作本機系統執行像是瀏覽或是儲存硬碟的工作。但最大的缺點在於它的程式執行效率無法與其他遊戲開發平台比較，為了讓 Flash player 播放器達到最小，而犧牲了 Flash 在程式上的執行速度。當遊戲遊玩類型是在要求執行效率較為嚴苛的情況下時，遊戲的開發架構與寫作方式就顯得格外重要。

基於上述背景，本研究提供一些在 Flash 遊戲效能上應注意的觀念與架構，並解釋不同 Flash 遊戲捲軸的執行原理，讓新進遊戲開發者自行建構知識，培養其遊戲設計概念。本研究章節如以下安排，第二節為圖形優化處理、第三節為程式效能優化、第四節為捲軸引擎、第五節為結論與未來展望、第六節為參考文獻。

2. 圖形優化處理

Flash 遊戲效能最佳化雖然與程式脫離不了關係，但圖形的呈像處理卻也不可或缺。許多有關 Flash 圖形處理的文獻，都曾敘述過如何處理圖形來使得程式執行效能提升。因此本節將介紹各項可以使 Flash 效能提升的圖形處理方式。

在“大師談 Flash 遊戲設計與製作”這本書裡[2]，曾提出以下幾點圖形處理的方式來優化遊戲程式效能。

- (1.) 根據特定的圖形需求使用點陣圖和向量圖。
- (2.) 索引型的 GIF 和 PNG 圖檔要在匯入 Flash 前先做最佳化處理。
- (3.) 要用失真的 JPEG 壓縮時，應匯入未經壓縮的 24 位元圖檔，並用 Flash 壓縮(只壓縮一次)。
- (4.) 照片品質的 24 位元 PNG 圖像要用 JPEG 與遮色片壓縮，以取得不同程度的透明度和好的壓縮。
- (5.) 當需要不同程度的透明度時，PNG-8 格式可提供只有幾個顏色或大區域顏色圖像最佳解決方案。
- (6.) 向量圖使用 Flash 最佳化曲線的功能，可有效減少複雜向量圖形。
- (7.) 半透明或平滑邊緣的圖，表現比純色圖差。

- (8.) 儘可能把點陣圖以百分之百的大小放在遊戲裡偶數像素的位置，如果不是放在偶數像素位置，就須進行補差才可顯示在畫面，這會非常消耗 CPU 資源。
- (9.) 執行時動態載入圖形素材可便於管理下載及增加彈性。
- (10.) 儘可能讓圖檔大小和圖形表現最佳化，每一點改善都對遊戲表現有很大的幫助。

表 1、各種圖檔格式特徵圖 (資料來源：大師談 Flash 遊戲設計與製作)

圖檔格式	GIF	PNG-8	JPG	PNG-24
適用	插圖、小圖案、文字。	需要半透明區域的圖形。	照片或色彩柔和轉換、漸層及透明圖像。	照片或色彩柔和轉換、漸層圖像。
壓縮	不失真。圖像是以刪除複製像素資料方式壓縮。減少圖像色彩可提高壓縮比。	不失真。圖像是以刪除複製像素資料方式壓縮。減少圖像色彩可提高壓縮比。	會失真。圖像是以刪除圖像和色彩資料方式壓縮。結果造成畫質損失。	會失真。使用 Flash 的 JPG 資料壓縮。
色彩	索引型。8 位元/256 色或以下。	索引型。8 位元/256 色或以下。	24 位元/數百萬色。	24 位元/數百萬色。
Alpha	每 1 位元/像素可不全透明或全透明。	每調色盤索引有 256 層透明度。	不支援 alpha。	8 位元/256 層透明度。

而在 Macromedia 的行動裝置開發人員中心[3]，也提出他們對於改善圖像來使效能提升的一些看法與建議。

- (1.) 製作動畫時，一開始先用 10 到 15 的每秒畫面更新速率(frame per second, fps)的影格速率，低影格速率可以讓檔案變得較小。在 Flash 裡，每一個元件關鍵影格都會讓 swf 影片再增加 12 個位元組。
- (2.) 簡化動畫，一次不要有太多的項目出現在畫面上。盡量避免像是透明或漸層等會拖慢效能的效果。
- (3.) 避免全景和透明淡入/淡出，靜態背景使用短暫(五個影格)淡出/淡入就夠了。如果可能的話，使用形狀即可，將線條移除掉，包含形狀的外框線。在 Flash 中線條的呈現，會經過較為複雜的處理，因此會拖慢速度。如有必要，可以改用形狀來建立外框線。

在日本則有人提出如何解決在 IE6 瀏覽器上 Flash fps 遲緩的問題[4]。

- (1.) Flash 影片在 Html 上的 fps 表現(圖 1)，會比在本機上執行的效率還要差。主要是受到瀏覽器和電腦的性能影響，因此可以利用更改 Flash 發佈設定裡的視窗模式，如：不透明無視窗、透明無視窗來改善視窗的影格更新速率(圖 2)。

指定なし	不透明	透明
-	value="opaque"	value="transparent"
		
FPS= 25	FPS= 30	FPS= 30

圖 1、Flash 不同視窗模式在 IE6 瀏覽器上的 fps 圖 (資料來源：Next Frame)

	指定なし	不透明 "opaque"	透明 "transparent"
Internet Explorer 6	低下	改善	改善
Netscape 7.1	低下	低下	低下
FireFox 1.0.4	低下	低下	低下
Sleipnir 2.00	低下	改善	改善

圖 2、Flash 視窗模式在不同瀏覽器的效能改善圖 (資料來源：Next Frame)

在<<神影無蹤-廖添丁>>網路遊戲創作的論文作者，另一篇文章“Flash 動畫製作技巧”中 [5]，也提出一些對動畫製作的技巧與經驗來幫助系統效能的提升。

- (1) 需注意 Flash 影片畫面更新區域的大小，更新區域的大小與效能成反比。
- (2) 利用皮影戲技法製作人體動作(圖 3)，可依照人體不同動作將肢體動作分開拆解，如：腳部關節拆解成大腿和小腿、關節角度不同而有不同動作圖檔。當動畫執行時，人體關節圖檔可以隨著動作適時增減，以達到增加效能的目的。

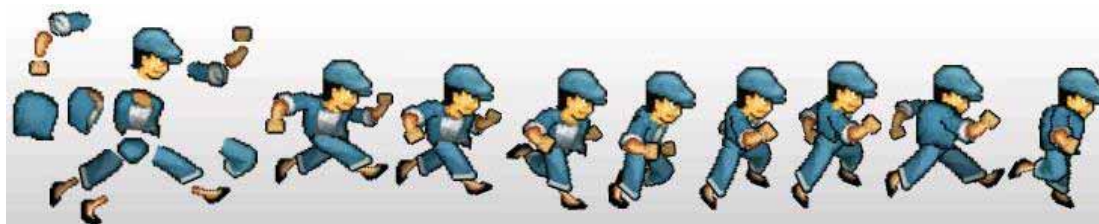


圖 3、皮影戲法人體動作拆解圖 (資料來源：Flash 動畫製作技巧)

3. 程式效能優化

由於 Flash 程式效能無法與其他遊戲開發平台相比，因此如何利用程式語法的優化來發揮程式執行時的最大效能是一門重要的課題。透過不同版本的 Flash player 播放器與迴圈來測試執行指令完成之所需花費時間(表 2)，可以得到下列數據供 Flash 程式設計師做為程式寫作時的效能參考依據[6] [7]。

表 2、迴圈測試執行指令完成所需花費時間。(資料來源：AcionScript performance tests)

Flash Player 版本	Flash Player 7 播放器	Flash Player 8 (beta) 播放器
Flash 軟體版本	Flash MX 2004 使用 Flash Player 7 發佈	Flash MX 2004 使用 Flash Player 7 發佈
所有區域變數使用 var 宣告	29ms	35ms
利用 this 直接參考物件	101ms	95ms
利用區域變數參考物件	235ms	117ms
for in 迴圈方式儲存陣列	184ms	68 ms
for 迴圈方式儲存陣列	67 ms	53 ms
while 迴圈方式儲存陣列	69 ms	55 ms
do while 迴圈方式儲存陣列	64 ms	53 ms
多重定義變數 (a=b=3)	43 ms	39 ms
逐一定義變數 (a=3, b=3)	50 ms	49 ms
使用 if else 判斷條件	85 ms	71 ms
使用 if else if 判斷條件	84 ms	71 ms
使用 switch 判斷條件	84 ms	69 ms
使用長變數名	91 ms	122 ms
使用短變數名	90 ms	120 ms
不宣告陣列長度並利用迴圈動態增加陣列元素	356 ms	379 ms
事先宣告陣列長度並利用迴圈增加陣列元素	185 ms	108 ms
i++和 i--	56 ms	53 ms
i+=1 和 i-=1	66 ms	71 ms

除了討論程式指令的優化，Flash 程式設計師在開發遊戲的架構時，可以注意以下幾點：

- (1.) 在 Flash 裡，onEnterFrame 事件是用來持續地驅動程式運作，其執行頻率和影格播放速率相等，因此如果影格播放速率越高，則需要更多 CPU 資源以處理每個影格的程式碼。所以盡量使用一個 onEnterFrame 事件來控制所有運作，如果分散至各個 MovieClip 將影響效能。舉例來說：在遊戲中，可以用一個 onEnterFrame 事件來驅動所有的敵人，這樣會比每個敵人各自用 onEnterFrame 事件驅動還要好控制效能損耗。
- (2.) 在 Flash 捲軸遊戲裡常會有許多的 MovieClip。假設遊戲區域比實際遊戲螢幕還要大，則螢幕外的 MovieClip 仍會佔用一定數量的系統資源，因此在 object base 的地圖可以使用 removeMovie()或 attachMovie()將遊戲螢幕外不可視範圍的物體做增刪的動作。但在標準的 tile base 的地圖，用 gotoAndStop()置換成空白影格會較快速。
- (3.) 在製作 MovieClip 的特效時，簡單的效果可以用時間軸製作動畫的方式呈現，複雜的效果可以用 Actionscript 程式語法製作，以節省不必要的 CPU 運算。

- (4.) 將 Flash 影格更新速率調整至最大值 120 fps，並撰寫 fps 偵測程式來監測遊戲畫面更新速率，以進行程式效能調整。設為 120 fps 的原因，是由於如果玩家本身電腦硬體效能很好，將導致無法看出遊戲中 fps 的劇烈變化。
- (5.) 將大量的 Flash 內容切割成幾個小型的 swf 檔案，並視情況需要動態載入，以節省檔案大小與玩家等待下載時間。
- (6.) Flash 內容製作完成時，可以使用 Flash 的檢視選項中的頻寬設定。頻寬設定提供一個顯示每個影格的資料傳送量。因此如果發現某個影格包含過多資訊，導致幾乎無法播放影片時，表示可能必須調整該影格部分內容以使影片順利執行[8]。

除了 Flash 程式的效能優化之外，記憶體空間的有效使用，也是另一個值得探討的議題，如：避免使用巢狀函式，以節省記憶體空間[9]。這在 Flash 開發到手機平台上時會顯得格外重要。且有一項觀念需要釐清，檔案大小並不等於使用的記憶體大小。檔案大小固然是重要的評估依據，不過不代表等於播放影片時所需的記憶體。例如：音效或圖片素材在製作時都會壓縮在 Flash 的 swf 檔，但當播放時，就必須還原成原來的大小。而其他像是 MovieClip 或程式碼，也都是在執行播放時比儲存在影片時須更多的記憶體。且 swf 影片執行時，可以建立更多物件，每次新增物件時都需要更多的記憶體，但檔案大小卻不會因此而改變[3]。

4. 捲軸引擎

在二維空間(2D)的遊戲世界裡，捲軸技術是很常使用的技巧。簡單來說，當玩家在遊戲世界裡控制角色向右持續移動時，事實上我們並沒有移動角色，而為了讓角色看起來像在向右移動，我們將遊戲世界中的其他圖像向左移動。因此一個捲軸引擎的好壞，對於遊戲的整體效能與流暢度是非常的重要。以下將介紹幾種在 Flash 遊戲捲軸引擎的做法。

在<<神影無蹤-廖添丁>>網路遊戲創作的論文裡[10]，提到將每一張地圖由許多小平台組成，而非一張張大型圖檔(圖 4)。使得圖形檔案可以重複被利用，而不致於使玩家下載太多時間，且能夠提升系統遊戲執行時的效能。基於此系統架構，每次場景捲動時，都將會有數十個至數百個圖片須處理移動動作，如不作前置處理則系統效能將嚴重不足。因此在場景移動前，會先做一次位置判斷，將螢幕可視範圍外的圖片暫時移除，只顯示並處理螢幕範圍中的所有圖片，以提升系統效能。



圖 4、場景構成示意圖 (資料來源：<<神影無蹤-廖添丁>>網路遊戲創作)

而國外的遊戲教學文章，則提供了一種叫做“gotoAndStop”的捲軸製作方式[11]，其原理是遊戲地圖由許多大小相同的方塊所組成。執行時只把可見部分的方塊放在螢幕上，當它們移出可見區域時，將相同的方塊移動到另一端，重新命名，並且再次使用相同的 MovieClip。

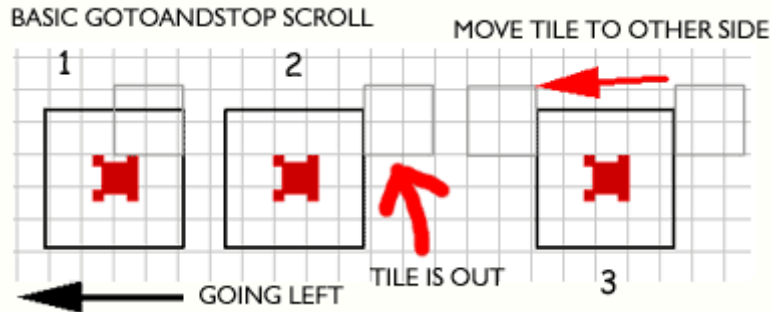


圖 5、gotoAndStop 捲軸原理示意圖 (資料來源：TILE BASED GAMES)

如圖 5 所示，當方塊從右邊可見區域消失時，立刻將方塊移動到左邊即將進入可見區域的位置。同時間要重新命名 MovieClip，而在新位置的方塊可能要顯示不同的圖片，因此需要使用 gotoAndStop() 指令去移動到正確圖片的影格。這也是為什麼捲軸引擎要被稱做是“gotoAndStop”的原因。由於地圖只需完整載入一次地圖方塊，就不需對 MovieClip 進行增刪的動作，因此在遊戲效能上會有所提升。

除了上述兩種捲動方式，也有文章提出用“super tiles”的方式來創建遊戲地圖[12]。遊戲世界是由許多與遊戲螢幕大小相同的區域切割而成，地圖方塊可由不同方塊組成任意形狀(圖 6)。當螢幕移動與其他區域接觸時，利用螢幕左上角判斷哪些區域是目前可視，並動態建立目前可視區域內所有的方塊列表。但在可視區域內還不代表可以在螢幕上立即顯示方塊，除非可視區域內的方塊進入螢幕範圍。當進入螢幕可視範圍時，依照方塊列表建立地圖方塊實體。最後當移出螢幕範圍時，利用旗標判斷來決定是否刪除方塊。此種方式除了效能不差，最大優點在於遊戲捲軸可以建立任意形狀大小的地圖方塊。



圖 6、super tiles 捲軸原理示意圖 (資料來源：Tile based/object based Flash Tutorial)

5. 結論與未來展望

Macromedia 公司所開發的 Flash 已從早期的網頁動畫製作軟體，漸漸走向了數位生活應用與娛樂科技的領域。在今日不論是手機介面、數位隨身聽、數位錄放影機、掌上型遊樂器，我們都可以看到 Flash 在其中的身影與潛力。因此本論文以娛樂的角度切入研究面，分析探討 Flash 在圖形處理與程式設計時，該如何應用才能使 Flash 在效能有限的平台上，發揮效能達到最佳化。並藉著介紹不同的捲軸引擎與其原理架構，期望培養遊戲設計人才。幫助遊戲開發者除了在遊戲效能及畫面流暢度上有所提升外，並能以此研究作為未來遊戲捲軸引擎更進一步的研究基礎。而本研究部分也可以應用於 Flash 手機遊戲開發上的效能提升，使 Flash 能在多媒體上有更多樣化的發展。

6. 參考文獻

- [1] Macromedia Produces(2005)。網址：http://www.macromedia.com/software/player_census)
- [2] Makar and Winiarczyk(民 92)，大師談Flash遊戲設計與製作，台北：上奇科技。
- [3] Andreas Heim(2005)，使用 Flash Lite 將 Flash 動畫移植至行動電話的最佳實習。網址：http://www.macromedia.com/tw/devnet/devices/articles/optimizing_anim.html)
- [4] Next Frame，Flash 再生速度的改善。網址：<http://nextframe.jp/fpsCheck.htm>)
- [5] 林秉舒(民 94)，Flash 動畫製作技巧。網址：<http://www.masolin.com/mmug/200507/>)
- [6] Oddhammer，AcionScript performance tests。網址：<http://www.oddhammer.com>)
- [7] 閃吧(民 93)，Actionscript 優化教程。網址：<http://www2.flash8.net/teach/1714.htm>)
- [8] Jen Dehaan(2004)，Macromedia Flash MX 2004：training from the source，Macromedia。
- [9] Timothee Groleau(2003)，Scope Chain and Memory waste in Flash MX。網址：http://timotheegroleau.com/Flash/articles/scope_chain.htm)
- [10] 林秉舒(民 93)，<<神影無蹤-廖添丁>>網路遊戲創作。國立台灣藝術大學多媒體動畫藝術學系碩士論文。
- [11] Tony(2003)，TILE BASED GAMES。網址：<http://www.tonypa.pri.ee/tbw/start.html>)
- [12] Strille(2004)，Tile based/object based Flash Tutorial。網址：http://www.strille.net/tutorials/part1_scrolling.php)