

# 致理技術學院

## 資訊網路技術系 實務專題報告

### 互動式教學-無線網路

指導老師：陳文雄

學生：陳彥璋(29534537)

彭文諒(29534511)

林君達(29534541)

吳翊毅(29534542)

中華民國 96 年 12 月

# 專題研究授權書

本授權書所授權之專題研究為

---

共四人,在致理技術學院資訊網路技術系 96 年度完成資網實務專題。

專題名稱: 無線網路互動式教學

本組同學共四人,皆同意著作財產權之論文全文資料,授予教育部指定送繳之圖書館及本組畢業之學校圖書館,為學術研究之目的以各種方法重製,或為上述目的在授權他人以各種方法重製,不限地域與時間,惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。

指導教授簽名:

專題生簽名:

(親筆正楷)

學號:

(務必填寫)

中華民國            年            月            日

## 誌謝

本專題能順利完成要感謝陳文雄老師在專題製作期間之督促與細心的指導,在遭遇瓶頸時.總是幫我們渡過難關解決心中的疑問且熱心提供許多寶貴的意見。

陳文雄老師治學嚴謹和對事物執著的態度是我們將來出社會要學習的榜樣,很感謝老師這段日子對我們關心和關懷,擴展我們的視野,讓我們在學業和為人處世上受益匪淺,這一切都深深影響著我們,也希望這能成為我們成長的動力。

在此也要感謝柯振根主任、資網系辦以及資網系統資源組林源淺老師,提供本專題系統開發時所需的各式軟體、軟體配件、無線網路設備,以及專案成果發表時系統展示所使用之實物投影機,僅此致上最誠摯謝意

## 摘 要

隨著電腦與無線通訊技術的快速發展，行動計算已然成為電腦通訊領域中備受矚目的一環。Mobile Ad-hoc (MANET) 是一種完全經由無線連結的行動節點所建構而成的網路，通常具有動態形成及有限頻寬等特性，網路裡的成員則可能有筆記型電腦、PDA(Personal Digital Assistant)、行動電話等。

正因為無線通訊的市場漸漸在成熟，也宣告了未來無線通訊將是主流，也知道很多人對於無線網路的觀念有點懵懂，也讓我們思考也許可以把我們所學的，盡量以顛覆以往無線網路的形式，用不同的手法去把這些觀念呈現出來，務必要使使用者覺得這不是在學習，而是在玩樂，呈現的手法也務求簡單明瞭，盡量不要長篇大論的文字。

舉個比較實際的例子，以最基本的無線網路通訊原理當作一個基礎，透過這個最基本的原理會衍生出哪些問題？透過FLASH的動畫製作功能，採用創意及簡單抓重點的方式，讓人們了解到學習不只是靠枯燥無味的文字，而是可以活潑愉快的。

# 目 錄

授權書.....	I
誌謝.....	II
摘要.....	III
目錄.....	IV
圖目錄.....	V
表目錄.....	VI
第壹章 緒論.....	1
第一節    互動式教學的重要性.....	1
第二節    研究動機與目的.....	3
第三節    研究範圍與報告書架構.....	4
第貳章 互動式教學理論與技術探討.....	5
第一節 Flash 技術之探討.....	5
第二節 Action Script 的探討.....	18
第參章 互動無線網路教學系統.....	29
第一節 系統架構.....	29
第二節 系統功能簡介.....	30
第三節 系統操作流程.....	32
第四節 系統特色.....	53

第肆章 系統呈現.....	63
第一節 預期效能.....	63
第二節 系統效能.....	64
第伍章 結論.....	65

## 圖目錄

圖 1-1 專題報告架構圖.....	4
圖 3-1 系統架構圖.....	29
圖 3-2 主動畫開端.....	32
圖 3-3 主動畫開端(二).....	33
圖 3-4 主動畫開端(三).....	34
圖 3-5 系統主畫面.....	35
圖 3-6 教學的目錄.....	36
圖 3-7 無線網路基本概念.....	37
圖 3-8 第二章目錄.....	38
圖 3-9 independent 架構.....	39
圖 3-10 infrastructure 架構.....	40
圖 3-11 無線網路訊框.....	41
圖 3-12 第三章目錄.....	42
圖 3-13 無線網路傳輸原理.....	43
圖 3-14 隱藏節點的問題.....	44
圖 3-15 第四章目錄.....	45
圖 3-16 DCF 概念.....	46

圖 3-17 PCF 概念.....	47
圖 3-18 世界地圖.....	48
圖 3-19 世界地圖(二).....	49
圖 3-20 問答題介面.....	50
圖 3-21 問答題介面(二).....	51
圖 3-22 問答題介面(三).....	52
圖 3-23 技術呈現.....	54
圖 3-24 技術呈現(一).....	55
圖 3-25 技術呈現(二).....	56
圖 3-26 技術呈現(三).....	57
圖 3-27 技術呈現(四).....	58
圖 3-28 技術呈現(五).....	59
圖 3-29 技術呈現(六).....	60
圖 3-30 技術呈現(七).....	61
圖 3-31 技術呈現(八).....	62
圖 3-32 技術呈現(九).....	63



# 第壹章 緒論

## 第一節 互動式教學的重要性

人跟人之間要互動是大家都公認的,但是面對冷冰冰的機器,如何達成雙向的教學,可以讓使用者真正達到思考的目的,而不是匆匆瀏覽過,為了這個目的,我們所設計的程式或者是網頁,都期望能夠達到互動兩個字的效果,到底該如何達到互動的效果呢?

我想先決條件是要使用者對我們製作的東西有興趣,如果連看一遍的興趣都沒有,所有在以這為前提的條件之下,我們的專題決定的方向乃採用大量精緻的動畫取代過往純文字式的學習方法,利用影音的內容來達成傳播知識的目的,讓使用者能夠以更輕鬆的態度去面對它,達成輕鬆學習的效果。

語音教學則是我們亟欲希望去做到的功能,以語言英文為例,大家都知道英文裡有 KK 音標,我們希望每點到一個音標,會有它的發音,我們以此為範例,是希望告訴大家,語音功能正是我們想要研發儒何應用在 FLASH 的功能之一。

據說，在牛津大學中，教師的主要任務是擔任學生導師，授課乃屬次要；學生以從導師那裡獲得研習指導為主要任務，上課聽講乃屬次要。正因為自學、獨立思考、全面發展這樣一個與眾不同的人才培養計劃，才使得牛津大學成為培養頂尖人才的搖籃。

由此可以看出所謂的學習並不是在課堂聽課，課後複習，課前預習，就算是學習，最主要的還是需要培養自己獨立思考的能力，這遠比學習來的重要的多，學習對大部分人來說只是在求學階段做的事情，殊不知即使出了社會學習的腳步也不能停，這正是成功的人和平凡老百姓之間的區別，而這區別至關重要的要素，是在於本身有沒有獨立思考的能力，而不是只會照本宣科，這也是我們認為互動式教學最重要的地方，不只把它落實在生活，更希望藉著我們努力開發出的程式，讓更多人領略互動式教學的真諦

## 第二節 研究動機與目的

最主要的研究原因還是適逢無線網路的興起和漸漸成熟的技術，雖然不太可能取代 Wireless Network，但兩者並存絕對是未來的趨勢，如何有善去運用變成我們主要的課題，有鑑於此讓我們對於 Wireless Network 有了無限的遐想，讓我們決定利用互動式教學的概念，把我們知曉的無線網路的知識透過 FLASH 的技術把它展現出來，利用我們對於無線網路的了解，透過動畫的方式生動的呈現在大眾面前，也正好可以給那些想踏入這個領域的人，一條便利的捷徑，初窺無線網路的精隨。

其中有一些傳輸運作的示意圖，我們將透過 FLASH 讓它生動的活起來，增加大家對無線的學習興致，想必會讓很多人感到新鮮。

### 第三節 研究範圍與報告書架構

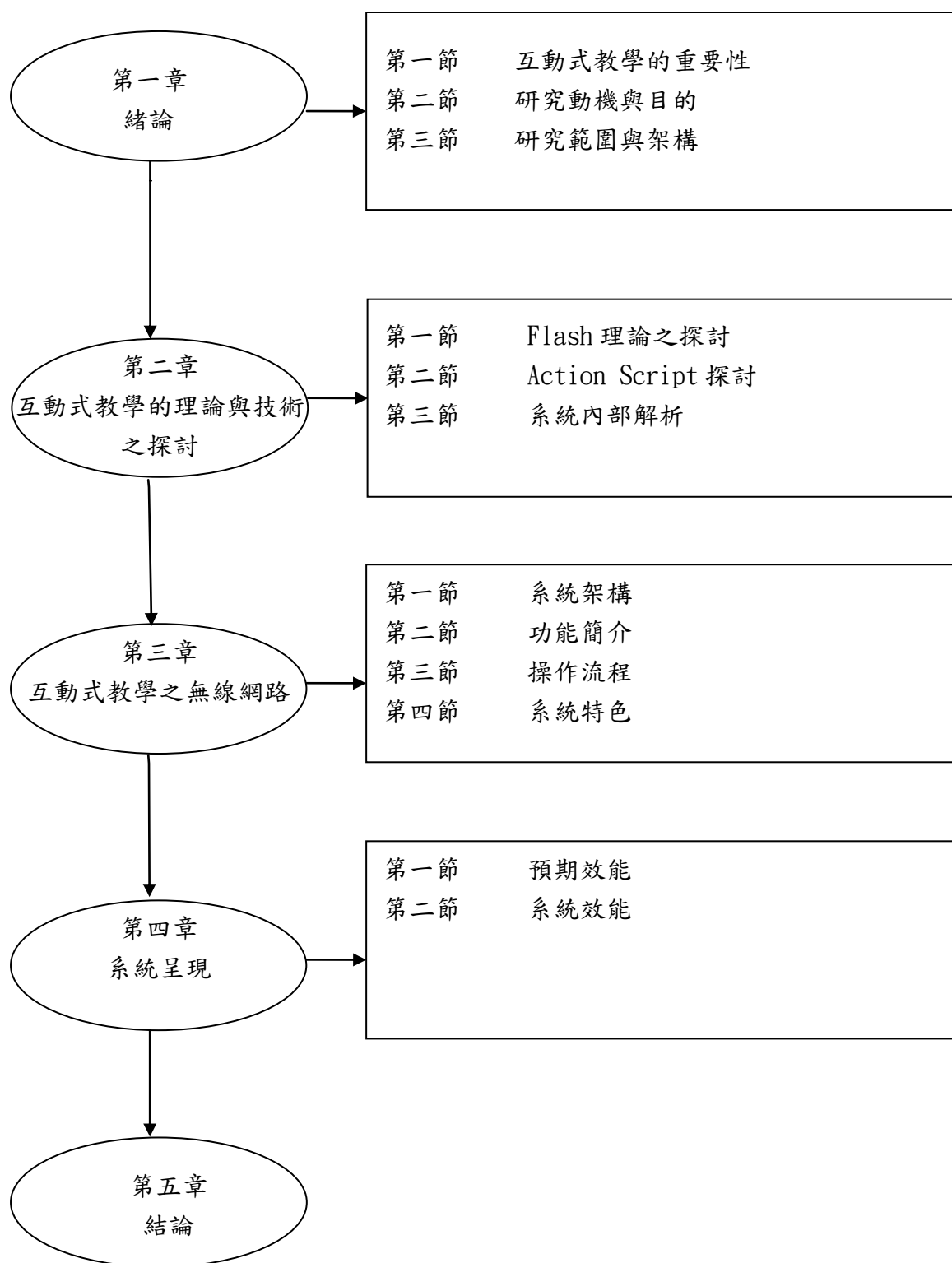


圖 1-1 專題報告架構圖

## 第貳章 互動式教學理論與技術探討

### 第一節 Flash 技術之探討

Flash 是 Macromedia 公司開發的網頁多媒體製作軟體，向量繪圖與動畫編輯功能，簡易地製作連續動畫、互動按鈕、繪圖與音效在您的網頁中使用！可以不需要任何程式腳本即可在網頁中增加互動式多體。

互動式的動畫和影音同步效果使網頁繪圖更加生動活潑，使用了 Flash 製作的任何物件，皆可以時間軸與動態路徑（Path-Based）的動畫設計方式，由淺入深，容易學習與使用！Flash 提供了很棒的線上教學。只有以向量為基礎的 Flash 多媒體，才能流暢地呈現在 Internet 上，即使放大縮小也不降低本身品質。

Flash 主要以 ActionScript 程式設計為導向，他能讓使用者快速學到開發互動式教材所需必備的知識與技巧，也可讓使用者快速了解 Actions 各種指令的用法，為將來程式開發設計作準備，奠下良好的基礎。使其對互動式教材的架構與觀念也能更透徹的了解。Flash 是一個高互動性的軟體教材，它讓使用者能以教材帶入 ActionScript 語法與自創函式，且直接切入互動式教材製作主題，使其學習有目的。

(一)Flash 主要有以下幾點特性：

1. 使用向量式圖形技術 來製作動畫，它使檔案容量較小（因為向量圖形是使用數學函數來記錄圖形中的屬性）；而且將向量圖放大或縮小，也不會失真，最重要的是檔案容量也不會改變。
2. 可為網頁配上悅耳動聽的音效，而且是以 MP3 的音樂壓縮格式壓縮，可大幅降低聲音所佔據的檔案容量，也可保有高品質的音質。〔可匯入 WAV (Windows) 的聲音檔〕。
3. 採用 Stream 資料流傳送方式，在檔案下載的同時即可流暢的播放，不須等到資料全部下載完畢才能觀看動畫。
4. 提供 Actions 指令設定環境，可使網頁作到極佳的互動性。
5. 為了讓使用者可以在完成動畫之後，立即看到動畫在網頁中的果，Flash 可直接將動畫出版成網頁，產生 HTML。
6. 具有抗鋸齒的功能，可讓文字或影像的邊緣都非常平滑。

7. 加強與支援點陣式圖形處理 (Enhanced Bitmap Support)，使之可旋轉、拉長等功能。

(二)在本節裡我們把 Flash 技術運用大概分為兩種

### 1. ActionScript 介紹

以 ActionScript 來說，使用 ActionScript 編寫作品時，如果使用正常模式的動作面板，通過功能表和列表選擇選項，可以創造簡單的動作。要想用 ActionScript 編寫功能強大的作品，你就必須深入瞭解和學習 Flash MX 的 ActionScript 語言。像其他語言一樣，ActionScript 也有變數、函數、物件、操作符號、保留關鍵字等語言元素，有它自己的語法規則。ActionScript 允許用戶創造自己的物件和函數。ActionScript 的語法和風格與 javascript 非常相似，但不完全相同。ActionScript 擁有自己的句法和標點符號使用規則，這些規則規定了一些字元和關鍵字的含義，以及它們的書寫順序。例如，在英語中用句號結束一個句子，而在 ActionScript 中則用分號結束一個語句。

(1) ActionScript 的一些基本語法規則，對於比較特別的規則。

a. 點語法

在 ActionScript 中，點(.)被用來指明與某個物件或電影剪輯相關的屬性和方法。它也用標識指向電影剪輯或變數的目標路徑。

點語法運算式由物件或電影剪輯名開始，接著是一個點，最後是要指定的屬性、方法或變數。例如，運算式 ballMC.x 是指電影剪輯實例 ballMC 的\_x 屬性，\_x 電影剪輯屬性指出編輯區中電影剪輯的 X 軸位置。

表達一個物件或電影剪輯的方法遵循相同的模式。例如，ballMC 實例的 play 方法用於移動 ballMC 的時間軸播放，就像下面的語句：`ballMC.play();`

b. 斜線語法

Flash 的以前版本是使用斜線語法指出電影剪輯或變數的目標路徑。Flash 8 播放器仍然支援這種語法，但不推薦使用。在斜線語法中，斜線被用來代替點，用以標明電影剪輯或變數的路徑。要指出一個變數，可以在變數前加上冒號。



(a)我們現在用點語法取代上述斜線語法來表示上面的目標路徑：

`myMovieClip.childMovieClip.myVariable` 斜線語法在 `tellTarget` 動作中使用最為普遍，但這種動作在 Flash MX 中已不再推薦使用，與其使用 `tellTarget` 動作還不如使用 `with` 動作與點語法相容。

### c. 大括弧

ActionScript 語句用大括弧({})分塊，如下面的所示：

```
on(release){  
  
    myDate = new Date();  
  
    currentMonth = myDate.getMonth();  
  
}
```

### d. 分號

ActionScript 語句用分號(;)結束，但如果你省略語句結尾的分號，Flash 仍然可以成功地編譯你的作品。例如，下面的語句用分號結束：

```
column = passedDate.getDay();  
  
row = 0;
```

同樣的語句也可以不寫分號：

```
column = passdDate.getDay()
```

```
row = 0
```

#### e. 圓括號

定義一個函數時，要把參數放在圓括號中：

```
function myFunction (name, age, reader){  
    ...  
}
```

調用一個函數時，也要把要傳遞的參數放在圓括號中：

```
myFunction ("Steve", 10, true);
```

圓括號可以用來改變 ActionScript 的運算優先順序，或使自己編寫的 ActionScript 語句更容易閱讀。

也可以用圓括號來計算點語法點左邊的運算式。例如，在下面的語句中，圓括號使運算式 `new color(this)` 得到計算，並創建了一個新的顏色物件：

```
onClipEvent(enterFrame){
```

```
(new Color(this)).setRGB(0xffffffff);  
}
```

在上例中，如果不使用圓括號，就需要在代碼中增加一個語句來計算它：

```
onClipEvent(enterFrame){  
    myColor = new Color(this);  
    myColor.setRGB(0xffffffff);  
}
```

#### f. 大小寫字母

在 ActionScript 中，只有關鍵字區分大小寫。對於其餘的 ActionScript，可以使用大寫或小寫字母。例如，下面的語句是等價的：

```
cat.hilite = true;
```

```
CAT.hilite = true;
```

但是，遵守一致的大小寫約定是一個好的習慣。這樣，在閱讀 ActionScript 代碼時更易於區分函數和變數的名字。如果在書寫關鍵字時沒有使用正確的大小寫，你的作品腳本將會出現錯誤。

例如下面的兩個語句：

```
setProperty(ball, _xscale, scale);
```

```
setproperty(ball, _xscale, scale);
```

前一句是正確的，後一句中 `property` 中的 `p` 應是大寫而沒有大寫，所以是錯誤的。在動作面板中啟用彩色語法功能時，用正確的大小寫書寫的關鍵字用藍色區別顯示，因而很容易發現關鍵字的拼寫錯誤。

#### g. 註解

需要記住一個動作的作用時，可在動作面板中使用 `comment`(註解) 語句給影格或按鈕動作添加註解。如果你在協作環境中工作或給別人提供範例，添加註解有助於別人對你編寫的作品正確理解。

在動作面板中選擇 `comment` 動作時，字元 “//” 被插入到作品中。如果你創建腳本時加上註解，即使是較複雜的腳作品也易於理解，例如：

```
on(release){  
  
    //建立新的日期物件
```

```
myDate = new Date();

currentMonth=myDate.getMonth();

//把用數位表示的月份轉換為用文字表示的月份

monthName = calcMoth(currentMonth);

year = myDate.getFullYear();

currentDate = myDate.getDat();

}
```

#### h. 關鍵字

ActionScript 保留一些單詞，專用於本語言之中。因此，不能用這些保留字作為變數、函數或標籤的名字。下表列出了

ActionScript 中所有的關鍵字：

break continue delete else

for function if in

new return this typeof

var void while with

這些關鍵字都是小寫形式，不能寫成大寫形式。

## i. 常量

常數有其值永不改變的屬性。常量用大寫字母列於動作工具箱中。例如，常數 BACKSPACE、ENTER、QUOTE、RETURN、SPACE 和 TAB 是 Key 物件的屬性，指鍵盤上的鍵。要測試用戶是否在按 Enter 鍵，可使用下面的語句：

```
if(keycode()= =key.ENTER){  
  
    alert = "準備好了嗎?"  
  
    controlMC.gotoAndStop(5);  
  
    }  
  
emma.shove(hamish, 10);
```

## (2) 作品之流程

Flash 將從第一行語句開始執行 ActionScript 語言，一直按次序執行到最終語句或 ActionScript 指定跳轉到的語句。

將執行次序發送到其他地方而不是下一語句的動作包括：if 語句、do...while 迴圈和 return 動作。

if 語句被稱為條件判斷語句或“邏輯分支”，因為它是基於某些條件的評估判斷來控制作品流程的。

用戶也可以添加 else 語句來創建更加複雜的條件判斷語句，還有迴圈語句將重複執行某一動作若干次，直到滿足某一條件才結束。

### (3) 作品的運行

在編寫作品時，用戶可以使用動作面板，將作品附加給時間軸中的影格，也可以附加給舞台上的按鈕或影片片段。附加給影格的腳本將在播放到達該影格時執行，而附加給按鈕或影片片段的作品則在某一事件發生時執行。所謂的“事件”，可以是電影播放時滑鼠移動、某個按鈕被按下，也可以是某個影片片段被載入等。使用 ActionScript 可以發現操作發生的時間，然後根據該事件執行指定的作品。

附加給按鈕或影片片段的動作被一個特殊的動作包圍，該動作被稱之為 handler(處理程式)，onClipEvent 和 on 動作都被稱之為處理程式，因為它們操控或管理著操作。用戶可以為每個處理程式指定一個或多個操作。當由處理程式指定的事件發生時，影片片段或按鈕動作才執行。如果要在不同的事件發生時執行不同的動作，則可以給一個物件附加多個處理程式。

onClipEvent 動作處理的是影片片段的事件，而 on 動作則處理按鈕的事件。用戶也可以和影片剪輯一起使用 on 動作，創造一個能接受滑鼠事件的影片片段。

## 2. Flash Methods 函數集解釋

Play() 播放動畫

StopPlay() 停止動畫

IsPlaying() 動畫是否正在播放(true, false)

GotoFrame(frame\_number) 跳轉到某影格(frame\_number+1)

TotalFrames() 獲取動畫總影格數

CurrentFrame() 回傳當前動畫所在影格數-1

Rewind() 使動畫返回第一影格

SetZoomRect(left, top, right, buttom) 放大指定區域

Zoom(percent) 改變動畫大小

Pan(x\_position, y\_position, unit) 使動畫在 x, y 方向上平移

PercentLoaded() 返回動畫被載入的百分比(0-100)

LoadMovie(level\_number, path) 載入動畫

TGotoFrame(movie\_clip, frame\_number) movie\_clip跳轉到指定



幀數

TGotoLabel(movie\_clip, label\_name) movie\_clip 跳轉到指定標

籤

TCurrentFrame(movie\_clip) 回傳 movie\_clip 當前影格-1

TCurrentLabel(movie\_clip) 回傳 movie\_clip 當前標籤

TPlay(movie\_clip) 播放 movie\_clip

TStopPlay(movie\_clip) 停止 movie\_clip 的播放

GetVariable(variable\_name) 獲取變數

SetVariable(variable\_name, value) 變數賦值

TCallFrame(movie\_clip, frame\_number) call 指定影格上的  
action

TCallLabel(movie\_clip, label) call 指定標籤上的 action

TGetProperty(movie\_clip, property) 獲取 movie\_clip 的指定  
屬性

TSetProperty(movie\_clip, property, number) 設置 movie\_clip  
的指定屬性。

## 第二節 Action Script 的探討

### (一) 變數

在 Action Script 中, 變數是一個重要的概念, 當定義一個變數時, 就應該為變數定一個明確的值, 變數的初始化通常發生在動畫的第一影格中, 初始化變數使得當動畫播放時, 對變數值的跟蹤和比較便的較為容易。

Action Script 中的變數可以儲存所有類型的資料包含字串、數值、布林值、物件和影片片段, 當某個變數在一個腳本中被指定值時, 變數包含的資料類型將影響變數值的改變變數的命名。

#### 1. 變數的命名

一個合法的變數必須滿足下列要求。首先它必須是一個識別字, 並要滿足識別字的相應要求。不能是 Action Script 的關鍵字或者一個代表布林值得 true 或 false。在作用範圍內, 它的名字必須是唯一的。

#### 2. 變數的輸入

變數的輸入, 在 Action Script 中, 不需要明確的定義一個變數是要儲存為數字還是字串或者其他的資料類型態, 當變數被指定值

時,FLASH8 會自動確定這個變數的資料類型,這一點和其他程式不同。

例如這個程式:`x=" name"`;這個變數是 String 的資料型態,以後的指定值可能會改變 x 變數的資料類型,例如:語句 `x=12` 會使變數 x 資料類型變為 Number 資料類型,且 FLASH8 程式會對運算子右側的運算式進行求值。

當運算式需要時, Action Script 還會自動變換變數的類型,例如:把一個值傳遞給 trace 動作時, trace 動作會自動將接收到的值轉換成字串,我們可以使用這個動作來除錯腳本,跟蹤變數的值以及跟蹤腳本的分之流向。

在含有運算子的運算式中, Action Script 會根據需要變換資料類型,例如:當「+」運算子被用於字串計算時, Action Script 會自動將另一個資料類型轉換為字串(如果它不是字串)

```
"Next in line,number" +1
```

Action Script 會自動將數位 1,轉換為字串 1 並將其載入到第一個字串的後面形成一個更長的字串「Next in line,number1」

### 3. 變數的宣告

總體變數的宣告,可以使用 set variables 動作或指定運算子,這兩種方法可以達到同樣的目的;局部變數的宣告,則可以在函數內部使

用 var 語句來實現, 局部變數的作用域被限定在所處代碼塊中, 並在塊結束處終結, 未再快的內部被宣告的局部變數將在它們的腳本結束處終結。

#### 4. 變數的作用域

當變數在其作用域中便可以引用, 在 Action Script 中的變數可以被定義為總體變數或局部變數。一個總體變數可以在所有的時間軸中共用, 而一個局部變數僅能在其所屬的代碼塊(即大括弧)內使用。

使用局部變數可以有效的防止名稱的衝突, 因此而被大量的使用, 但在有些情況之下, 則必須使用總體變數, 例如: 如果要在一個代碼塊中引用另一個代碼塊中的變數, 或者要在整個動畫中傳遞某個變數的值時, 總體變數將會發揮巨大的作用。

#### 5. 在腳本中使用變數

要想在腳本中使用變數, 首先必須在腳本中宣告這個變數, 如果使用了沒宣告的變數, 則將會出現錯誤。另外, 還可以在一個腳本中多次改變變數的值, 變數包含的資料類型將對變數何時以及怎樣改變產生影響, 原始的資料類型, 如字串和數字等, 將以值的方式進行傳遞, 也就是說變數的實際內容將被傳遞給變數。

例如: 變數 ting 包含一個基本資料類型的數字 4, 因此這個實際的值數字

4 被傳遞給了 sqr 返回值為

```
Function sqr (x){  
  
Return x * x ;  
  
}
```

```
Var ting = 4 ;
```

```
Var out = sqr(ting);
```

其中, 變數 ting 中的值仍然是 4, 並沒有改變

又例如: 在以下的程式中, x 的值被設置為 1, 然後這個值被賦予給 y, 隨後 x 的值被重新改變為 10, 但此時仍為 1, 因為 y 並不跟蹤 x 的值, 它只是儲存 x 曾經傳遞給它的值。

```
var x =1 ;
```

```
var y =x ;
```

```
var x =10 ;
```

## (二) 物件

一個物件是多個特性的組合, 每個特性都有都有相應的名字和值, 一個特性的值可以是任何一種資料類型, 甚至可以是物件資料類型。這使得我們可以將一個物件放到另外一個物件的內部, 或者說是將它們

進行套入。

我們還可以使用 Action Script 預定定義的物件來存取和操作特定類型的資訊, 例如, 可以透過 Math 物件的方法來對傳遞給它的數字進行數學運算。

例如, 下面的語句使用 Math 物件 sqrt 方法對數位 256 進行平方根運算, 並將運算的結果給變數 squareRoot。

```
squareRoot = Math.sqrt(256) ;
```

另外, 在 Action Script 的影片片段物件具有可以控制舞台上的影片片段實體的方法, 下面語句使用了 play 和 nextFrame 方法

```
mcInstanceName.play();
```

```
mc2InstanceName.nextframe();
```

### (三) 函數

#### 1. 內置函數

所謂內置函數, 是一種語言在內部集成的函數, 已經完成了定義的過程, 當需要傳遞參數呼叫時, 就可以直接使用, 它能夠存取特定的資訊, 以及執行特定的任務, 例如獲得播放影片的 Flash Player 的版本號 (備註) 屬於物件的函數稱為方法, 不屬於物件的函數稱為頂級函數, 可以

在動作面板的函數類別找到它們，每個函數都有自己的特性，且某些函數需要傳遞特定的值，如果傳遞的參數多餘函數的需要，多餘的值將被忽略，如果不傳遞所需的參數，空的參數會被指定 undefined 資料類型，那麼在匯出腳本時，可能會導致出現錯誤，要呼叫函數，必須在播放頭到達那個影格中才可以自定義函數。

## 2. 自定義函數

除了內置函數以外，Flash8，同樣允許我們自己定義函數來滿足需求，和內置函數一樣，自定義函數也可以返回值、傳遞參數，還可以在定義之後被任意呼叫，為了便於理解，使用時不必去關心它的內部是如何運行的，只需要知道它能得到什麼參數，同時又能返回什麼值，就可以放心的使用。(備註)可以在任何需要的位置定義所需要的函數，但要記住，一定要在函數被呼叫之前定義，否則 Flash 腳本會出錯(先定義，後使用)。

一般情況都在影片的第一影格定義函數，這和變數初始化的思想相同，但重新定義函數時，新的定義會取代舊的定義。

下面是一個名為 areaOfGircle 的函數，帶有一個參數 radius，表示圓的半徑，該函數用於求一個圓的面積。

```
Function areaOfGircle(radius){  
  
Return Math.PI * radius * radius ;  
  
}
```

我們可以透過建立函數文字來定義函數，函數文字是一種未命名的函數，它是在運算式中而不是在語句中宣告的，另外，可以使用函數文字定義函數、返回它的值以及將其分配給一個運算式中的變數，如下所示：

```
area = (function(){return.Math.PI * radius * radius;})(5);
```

(備註)我們在 Action Script 編程中應盡量使用局部變數而不是總體變數，這樣做的好處是因為局部變數只存在於函數的內部，可以使函數和函數之間相對獨立，除了參數的傳遞之外，互相之間沒有任何干涉，從而可以放心大膽使用每一個函數，並在函數內部定義任何需要的變數，而不必考慮變數名衝突的問題。

#### (四) 程式控制

##### 1. 目標路徑

動作的名稱和位址被指定以後，才能使用它來控制一個影片片段或者下載一個動畫，這個名稱和定址就稱為目標路徑。



在動作面板中打開全域函數下的影片片段控制, 可以看到各種控制電影的動作, 在 Action Script 中, 上面的動作接收一個或一個以上的目標路徑作為程式運行時需要提供相應的控制參數。

例如, 使用 load Movie 動作需要提供三個參數

```
loadMovie(URL, Location, Variable);
```

其中 URL 參數是想要裝載的影片片段 web 上的定位; Location 是影片片段的目標位置路徑, 即確定影片片段將被下載到什麼位置,

Variable 用於制定使用哪種方法發送與當前要下載的影片片段相關聯的變數, 有 get 和 post 兩種方法。

在 Action Script 中, 透過影片片段的實體名來識別一個影片片段, 例如, 在下面的語句中, 實體名為 film 的影片片段的透明度被設定為 80%

```
Film._alpha = 80 ;
```

## 2. 條件語句

條件語句, 即一個以 if 開始的語句, 用於檢查一個條件的值是 true 還是 false, 如果條件值為 true, 則 Action Script 按順序執行後面的語句; 如果條件值為 false, 則 Action Script 將跳過這個代碼段, 執行

下面的語句。

If 經常和 else 結合使用,用於多重條件的判斷和跳轉執行

例如,下面的程式語句就是用來測試條件和執行相關動作的,如果前面的條件返回值為 false,則執行 else 後的語句。

```
If((password) = 123456 && (address = room)){  
gotoAndStoop( “reject” );  
}  
  
else{  
gotoAndPlay( “film” );  
}
```

### 3. 迴圈語句

在 Action Script 中,可以按照一個指定的次數重覆執行一系列的動作,或者是在一個特定的條件下執行某些動作,在使用 Action Script 編程時,可以使用 while, do…while, for 和 for…in 動作來建立一個迴圈語句。

在 Action Script 中,可以使用 for 語句使程式按照指定的次數迴圈,在一些特殊場合中,該功能具有很重要的意義。在很多情況之

下,需要預設一個計數初值來控制迴圈的次數,透過宣告一個變數,並編寫一條每當迴圈被執行一次就能對變數進行自增或自減運算的語句的方法,便可以實現一個對迴圈的控制,以達到我們所需求的。

在一個完整的 for 語句中,計數值和計數值的自增或自減運算的語句是其中的一部份,例如下面的語句。

```
For(i=5;i>0;i--){  
  
MyMovieClip.duplicateMovieClip(“newMovieClip” +i, i);  
  
}
```

條件迴圈語句在編程過程中,有時需要再滿足一個條件的情況下實現一個迴圈,這時需要用到 while 語句或 do...while 語句。

在 Action Script 中,迴圈語句 while 用於對一個運算式求值,如果運算式的值是 true,那麼迴圈體中的代碼將被執行。

當迴圈中的所有語句都被執行以後,運算式將按照程式要求再次進行求值,就這樣反覆進行直到運算式的值變為 false,才跳出本次迴圈,執行迴圈後的語句。

例如,在下面的程式與劇中,動作被重覆執行了五次

```
I=5  
  
While(I>0){
```

```
MyMovieClip.duplicateMovieClip( "newMovieClip" +I, I);  
  
i--;  
  
}
```

另外,也可以使用 do...while 語句建立同樣的迴圈,在一個 do...while 語句中,運算式是在代碼塊的底部進行求值運算的,因此,這個迴圈至少會被執行一次。

例如,將上面的那段語句用 do...while 編寫如下

```
I=5  
  
do{  
  
MyMovieClip.duplicateMovieClip( "newMovieClip" +I, I);  
  
i--;  
  
}while(I>0)
```

# 第參章 互動無線網路教學系統

## 第一節 系統架構

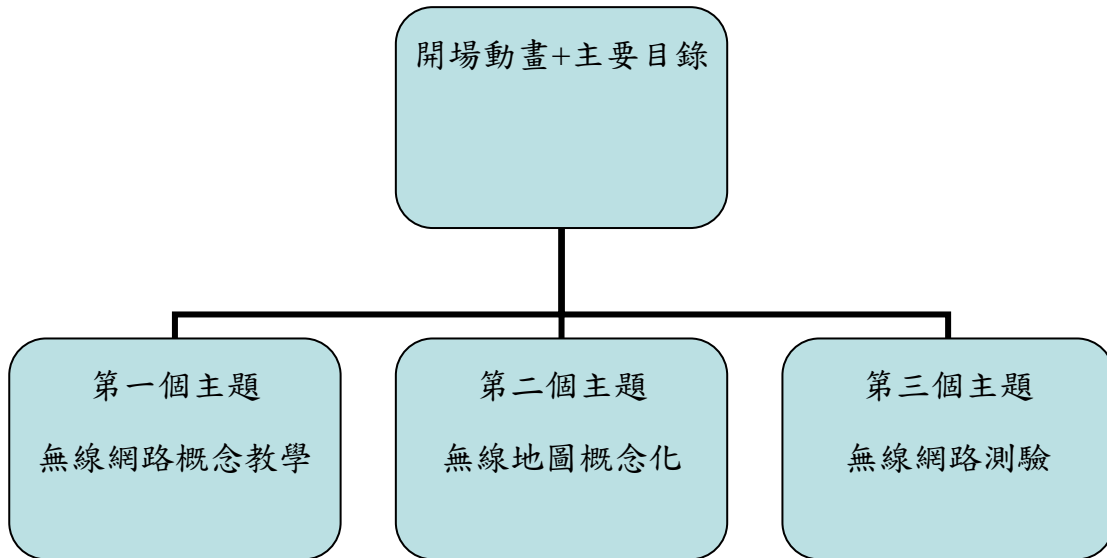


圖 3-1 系統架構圖

## 第二節 系統功能簡介

(一) 第一個主題無線網路概念教學裡面包含了七個無線網路的重要概念。

1. 概念講述了什麼是無線網路？以簡單的動畫型式，清楚的表達出無線網路最基本的概念，以圖形式的概念，讓使用者可以對其印象深刻以達成教學的目的。
2. 概念則是無線網路的基本架構，以動畫表示出這些基本架構的形式和分別，來幫助使用者建立觀念。
3. 概念則是無線網路的傳輸訊框，利用簡單的出場動畫，分別標示各個訊框的功能。
4. 概念則是隱藏節點的問題，針對無線網路傳輸可能會有的盲點，用動畫的形式提出說明。
5. 概念則是針對無線隱藏節點的問題，提出解決之道，利用怎樣的機制，才可徹底解決此問題。
6. 概念則是何謂 DCF？介紹無線網路傳輸的時候，工作站之間是用什麼樣的方式去存取媒介。
7. 概念則是相對的概念 PCF，跟 DCF 一樣是在講述工作站存取媒介的方式跟 DCF 有所區別。

(二) 第二個主題則是把無線網路的概念用地圖的形式,把它具體呈現出來,也許看了第一個主題總是覺得對於無線網路的概念有點懵懂,仿佛只是兩台電腦之間再傳送而已,我們針對於此,提出用世界的觀念去看無線網路這個技術,所以利用世界地圖,在各個國家做了啟動點,一但滑鼠碰到啟動點,則會連線世界各個角落,徹底達成無線網路無國界。

(三) 第三個主題則是針對之前的內容,設計了一些測驗題目,需要使用者動手把答案輸入,來測驗使用者之前的學習的效果。

### 第三節 系統操作流程

(一) 無線網路概念教學：

1. 一開始的主動畫的開端。(參考圖 3-2)



圖 3-2



2. 隨著時間跑動背景的人物和文字會不斷。(參考圖 3-3)

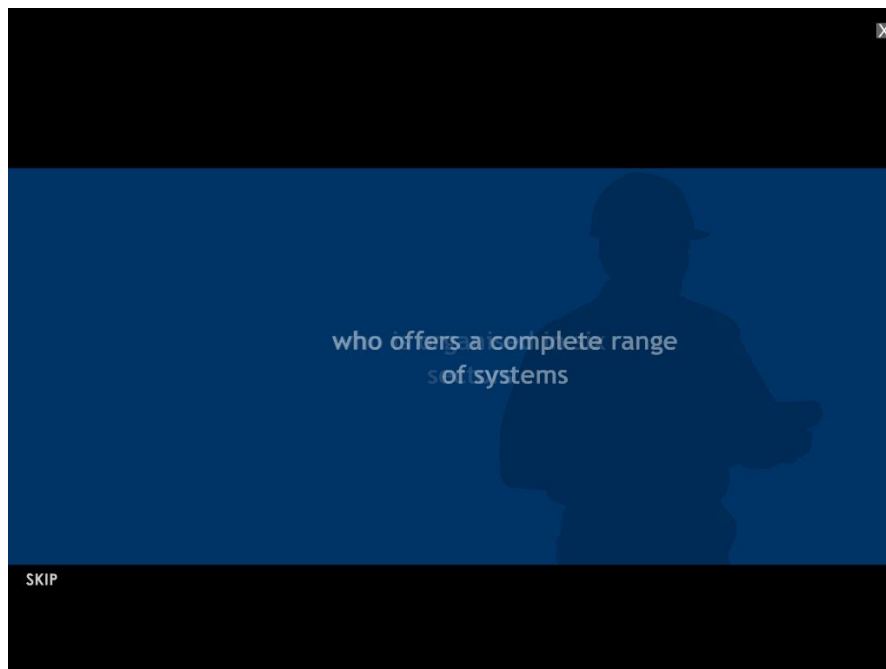


圖 3-3

3. 這動畫就包含了網路無國界的概念。(參考圖 3-4)



圖 3-4

4. 接著來到我們的主畫面。(參考圖 3-5)

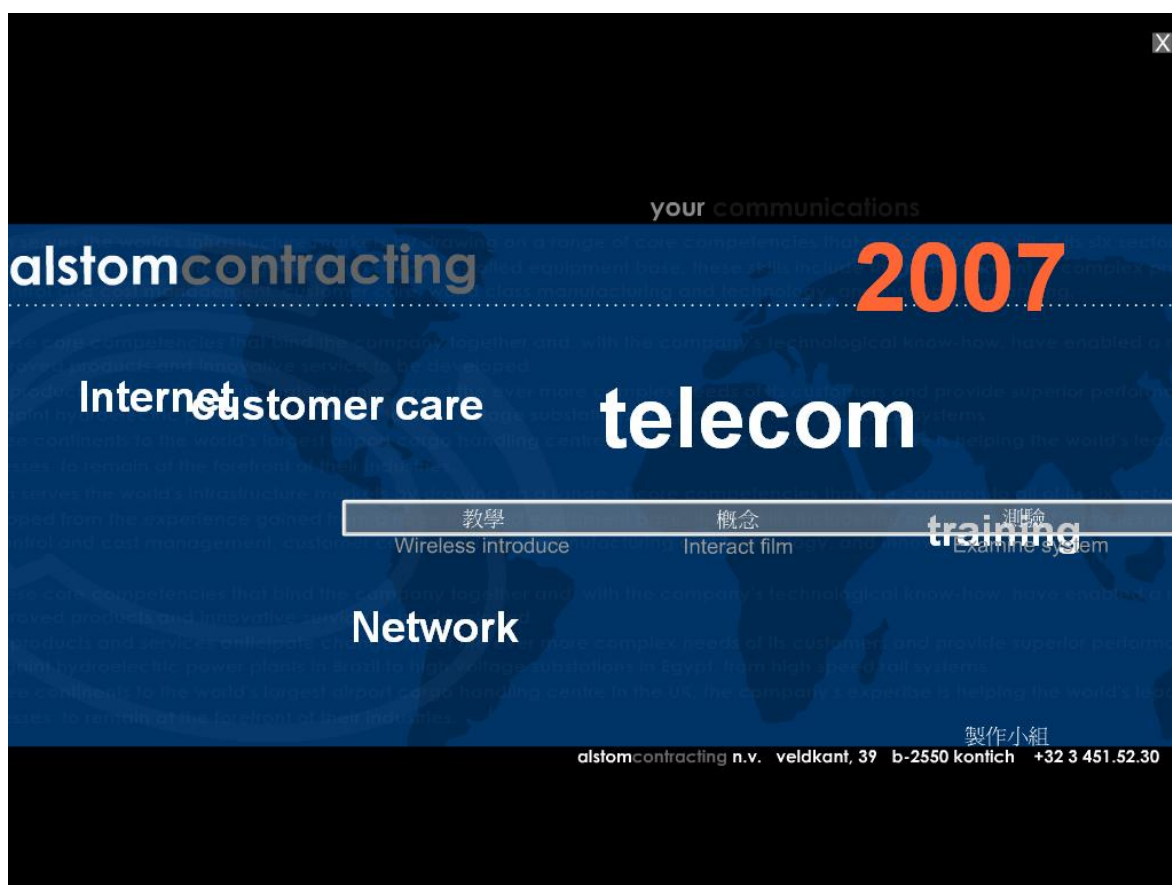


圖 3-5

5. 選擇教學進去後，會看到四個章節。（參考圖 3-6）



圖 3-6

6. 第一章講述無線網路的基本概念, 右下角有個動畫。(參考圖 3-7)



圖 3-7

7. 選擇第二章, 裡面有兩小結節。(參考圖 3-8)



圖 3-8

8. 這是 independent 的架構。(參考圖 3-9)

alstomcontract 2007

- Independent和infrastructure的架構
- Frame Control Field

在Independent中，工作站彼此可以直接通訊，兩者間的距離必須可以在直接通訊的範圍內。

The diagram illustrates an Ad Hoc Network. It features three stations: Station STA1 (top), Station STA2 (bottom right), and Station STA3 (bottom left). Each station is represented by a laptop icon with a wireless antenna. They are all enclosed within a dashed green circle labeled 'Ad Hoc Network'. Lines connect Station STA1 to Station STA2, Station STA1 to Station STA3, and Station STA2 to Station STA3, forming a triangular mesh. Each station also has a small antenna icon above it, indicating wireless communication capability.

Station STA3

Station STA2

Ad Hoc Network

○ Back

圖 3-9

9. 講述 infrastructure 的架構。(參考圖 3-10)

alstomcontract 2007

- Independent和infrastructure的架構
- Frame Control Field

在infrastructure如果工作站要跟其他工作站進行通訊，必經先將訊框傳給AP，再由AP把訊框傳給其他工作站。



The diagram illustrates an infrastructure network with two access points, AP-A and AP-B. AP-A is represented by a yellow box with the number '2' and a red antenna icon. It is connected to two yellow boxes, each also containing the number '2' and a red antenna icon. AP-B is represented by a blue box with the letter 'C' and a blue antenna icon. It is connected to two blue boxes, each also containing the letter 'C' and a blue antenna icon. The AP-A and its associated stations are enclosed in a red dashed oval, while AP-B and its associated stations are enclosed in a blue dashed oval. The entire diagram is set against a dark blue background with a white border.

○ Back

圖 3-10



10. 講述無線網路的訊框。(參考圖 3-11)

alstomcontract 2007

- Independent和infrastructure的架構
- Frame Control Field

Bits:  
2 4 1 1 1 1 1 1 1 1 1

Type:由兩位元組成,用來指定所使用的訊框類型,  
Type=00此訊框為管理訊框,  
Type=01此訊框為控制訊框,  
Type=10此訊框為資料訊框

○ Back

圖 3-11

11. 第三章進去後, 有兩小節。(參考圖 3-12)



圖 3-12

12. 講述無線網路的傳輸原理。(參考圖 3-13)



圖 3-13

13. 講述隱藏節點的問題。(參考圖 3-14)

alstomcontract 2007

- 無線網路的傳輸原理
- 隱藏節點問題

**隱藏節點問題**  
如下圖所示節點二可以直接跟節點一和三通訊,不過某些因素導致節點一和三無法直接通訊(這跟中間有沒有障礙物的關係不大,節點一和三之間可能是因為距離過遠,無法收到對方的無線電波)

從節點一的角度看,節點三屬於隱藏節點,如果使用簡單的祈禱送達協定,節點一和三有可能在同一時間傳送資料,這會造成節點二無從辨識任何資料,此外節點一和三將無從得知錯誤發生,因為只有節點二才知道有碰撞發生



○ Back

圖 3-14

14. 進入第四章後, 有兩小節。(參考圖 3-15)



圖 3-15

15. 講述 DCF 的概念。(參考圖 3-16)

alstomcontracting 2007

- 什麼是DCF?
- 什麼是PCF?

什麼是DCF?  
即分散式協調功能,是標準CSMA/CA存取機制的基礎,在傳送資料之前,它會先檢查無線鏈是否處於淨空狀態,爲了避免碰撞發生,當某個傳送者佔據頻道時,工作站會隨機爲每個選框選定一段延後時間,在某些情況之下,DCF可利用CTS/RTS淨空技術,進一步減少碰撞發生的可能

致於它的相關原理就是當訊框傳送完成並且經過一段DIFS時間,工作站便會試圖傳送之前擁塞的之後所緊接!

○ Back

圖 3-16

16. 講述 PCF 的概念。(參考圖 3-17)

alstomcontract 2007

- 什麼是DCF?
- 什麼是PCF?

什麼是PCF?  
即中樞協調功能,可讓無線網路提供較為公平的媒介存取機制,有點類似權杖式的存取機制,由基地台掌控權杖

如果需要用到免競爭傳輸,便可運用PCF,然而免競爭服務並非隨時供應,其服務時間的長短,交由中樞協調單元負責仲裁,也只有中樞協調單元可以存取媒介,和它連線的工作站,只有在中樞協調單元允許之下,才可以傳送資料,其概念是由權杖式的網路協定所衍生出來,在中樞協調單元裡是用輪詢取代權杖

其概念為當基地台掌管

○ Back

圖 3-17

(二)無線地圖概念化(參考圖 3-18)

1. 一開始世界地圖上有九個啟動點：



圖 3-18



2. 一經使用者以滑鼠碰觸會連到其他啟動點(參考圖 3-19)

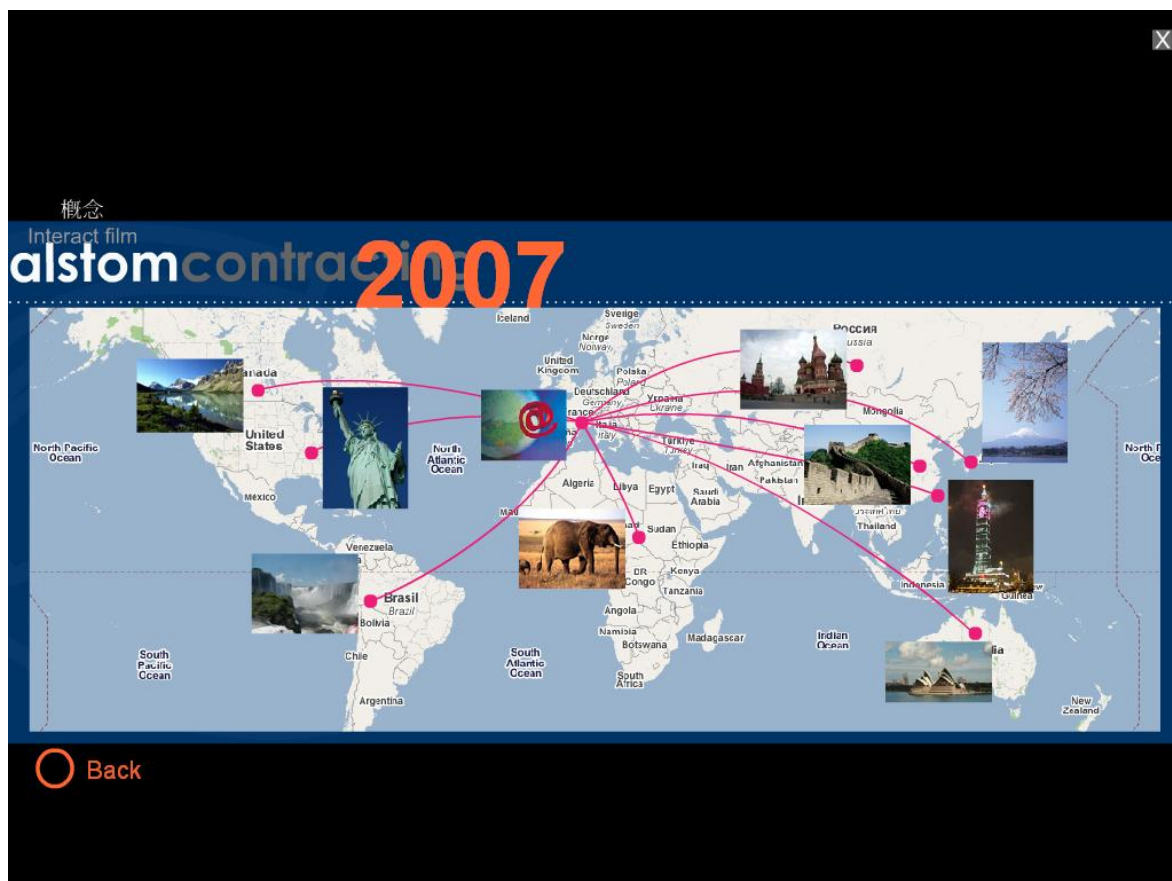


圖 3-19

### (三) 無線測驗測驗

1. 使用者要使用鍵盤輸入答案(參考圖 3-20)



圖 3-20

2. 答錯的話會顯示還有一次的機會(參考圖 3-21)



圖 3-21

3. 再錯的話會自動顯示出答案(參考圖 3-22)



圖 3-22

## 第四節 系統特色與技術

### (一) 特色

其實總結裡面的內容，幾乎沒有無線網路的相關圖片在裡面，我們總是期望能用比較卡通化的方式，讓使用者可以印象比較深刻，而不是把課本裡的東西，一樣搬上檯面，如此一來這個專題的課本只是一個會動一個不會而已，跟我們所期望會有很大的落差，也許裡面的內容讓懂無線網路的人來看可能會覺得很可笑，可是我們深信讓第一次接觸無線的人來說，他們會以開心的心情找到學習的動力。

特色一：基礎觀念明確，徹底把無線的概念呈現出來。

特色二：整體的質感跟由FLASH做的網站相似，可以增加使用者觀看的興趣。

特色三：把網路無國界的觀念具體呈現出來。

特色四：無線網路的測驗可以幫助使用者做一個簡單的學習效果評比。

## (二) 技術

### 1. 主畫面

(1) 下列程式碼讓使用者按下按鈕後，會跳到指定的場景

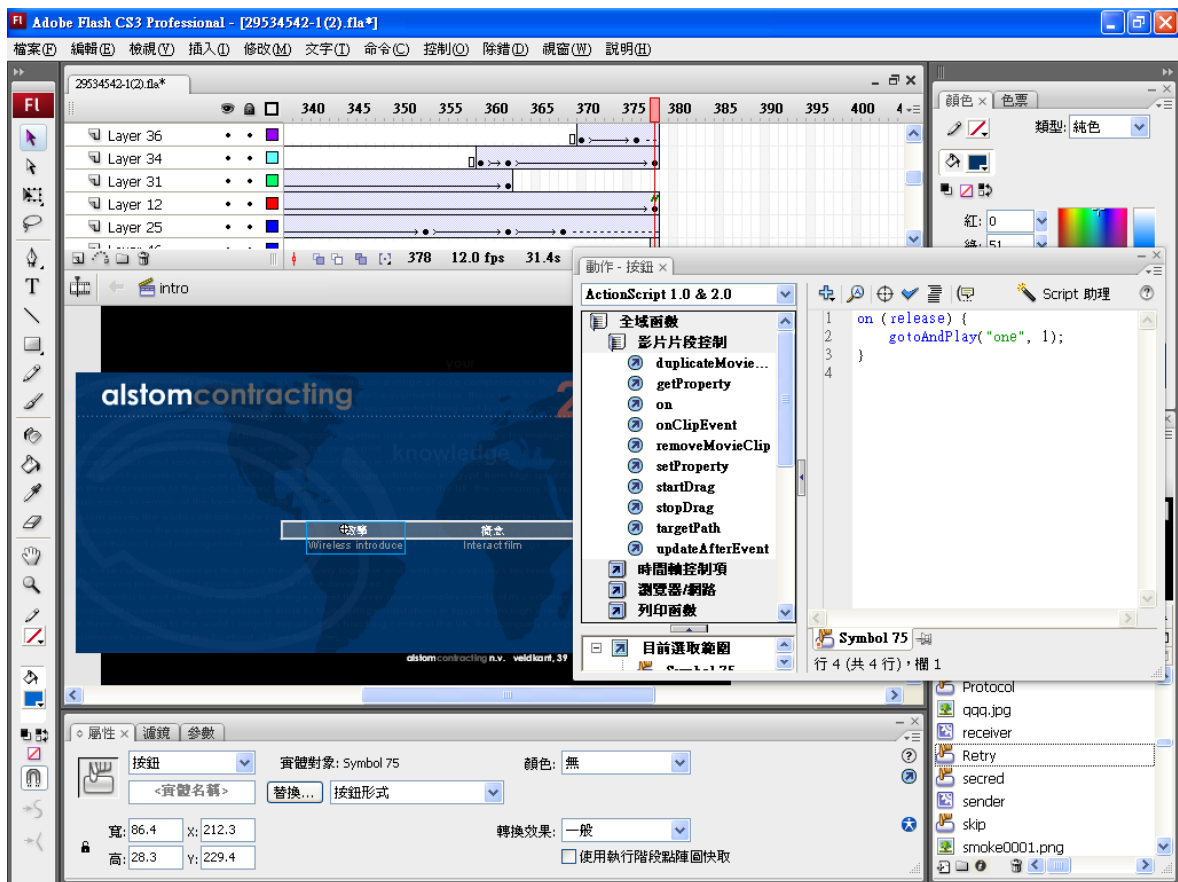


圖 3-23

(2)主畫面的閃爍文字,可以利用左邊的影片片段完成,再放到場景裡

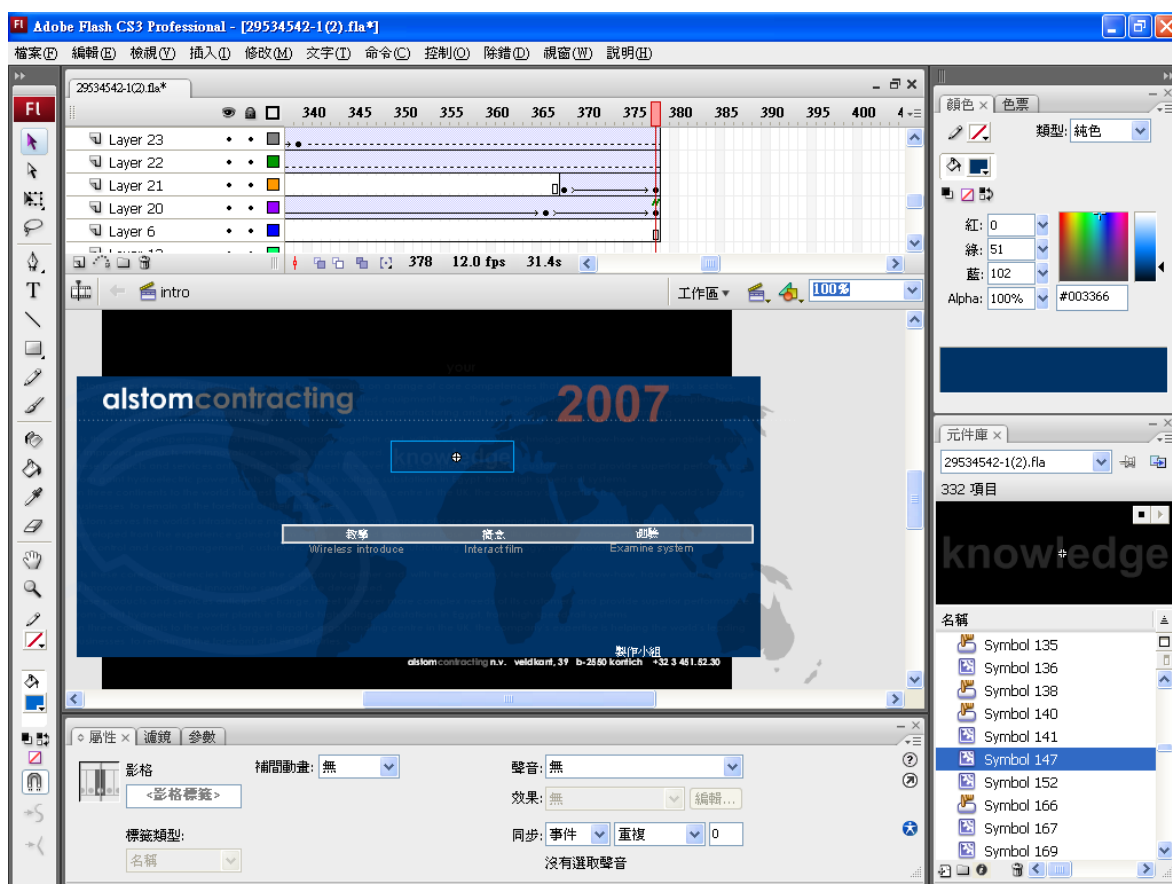


圖 3-24

## 2. 無線網路概念教學

(1) 下列程式碼動作，當使用者點選第一章後，第一章的動畫元件會播放，其他停止

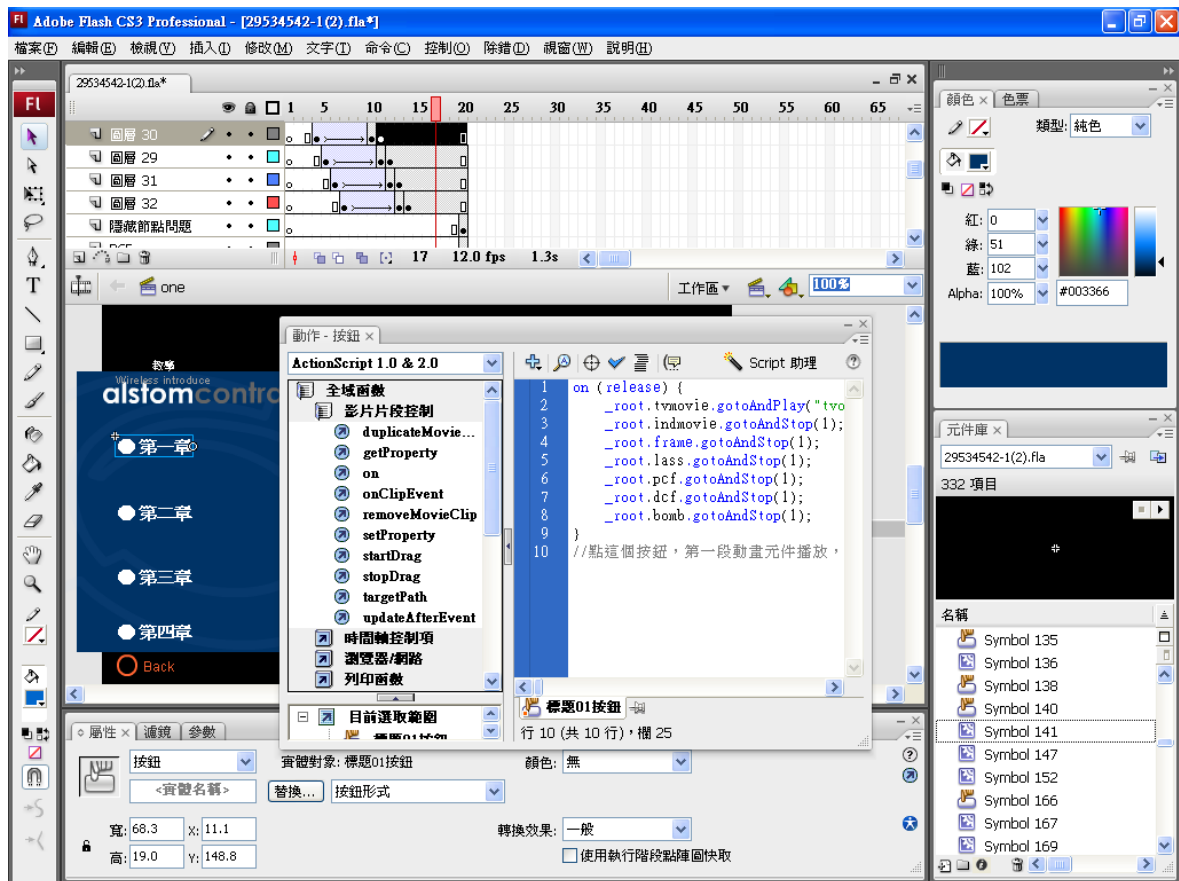


圖 3-25



(2) 第一章的影片片段內部影格

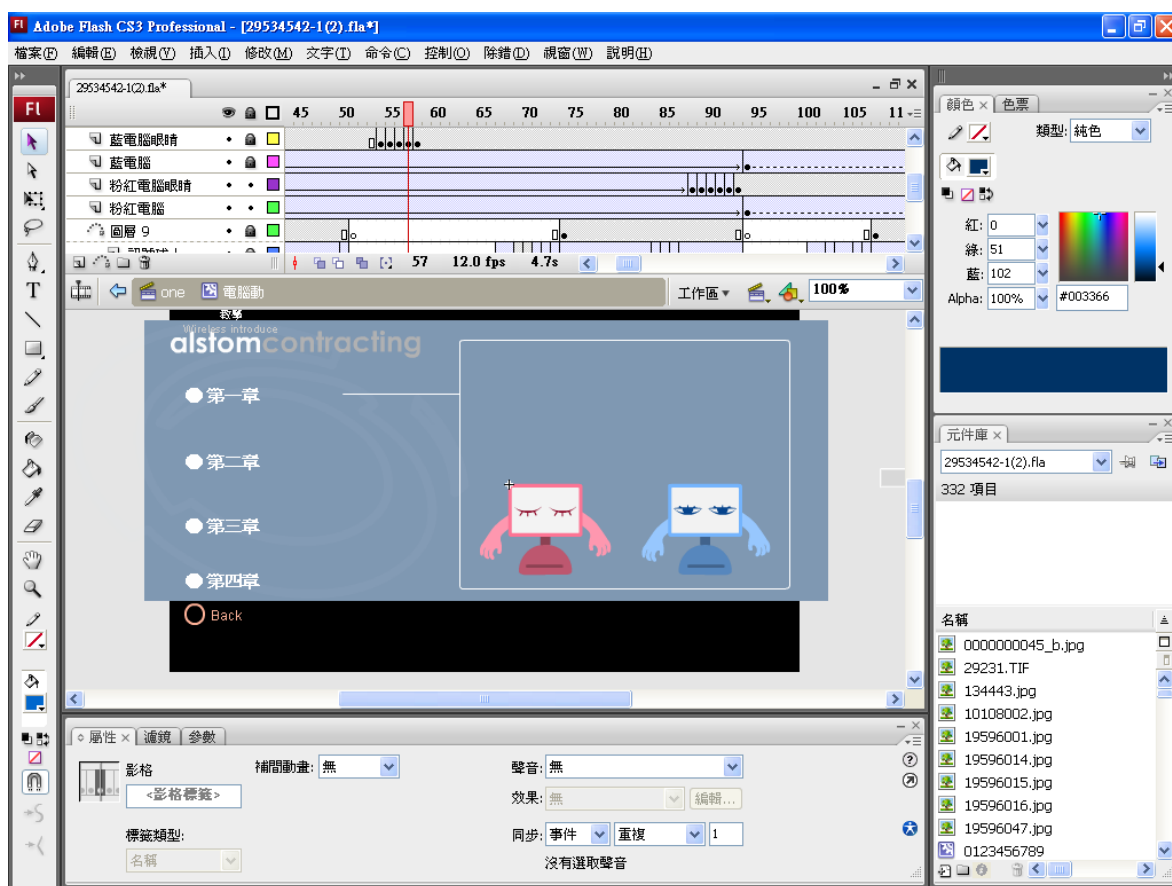


圖 3-26

### 3. 無線地圖概念化

(1) 當影格跑完後，要點選測試場景，地圖影片才會呈現

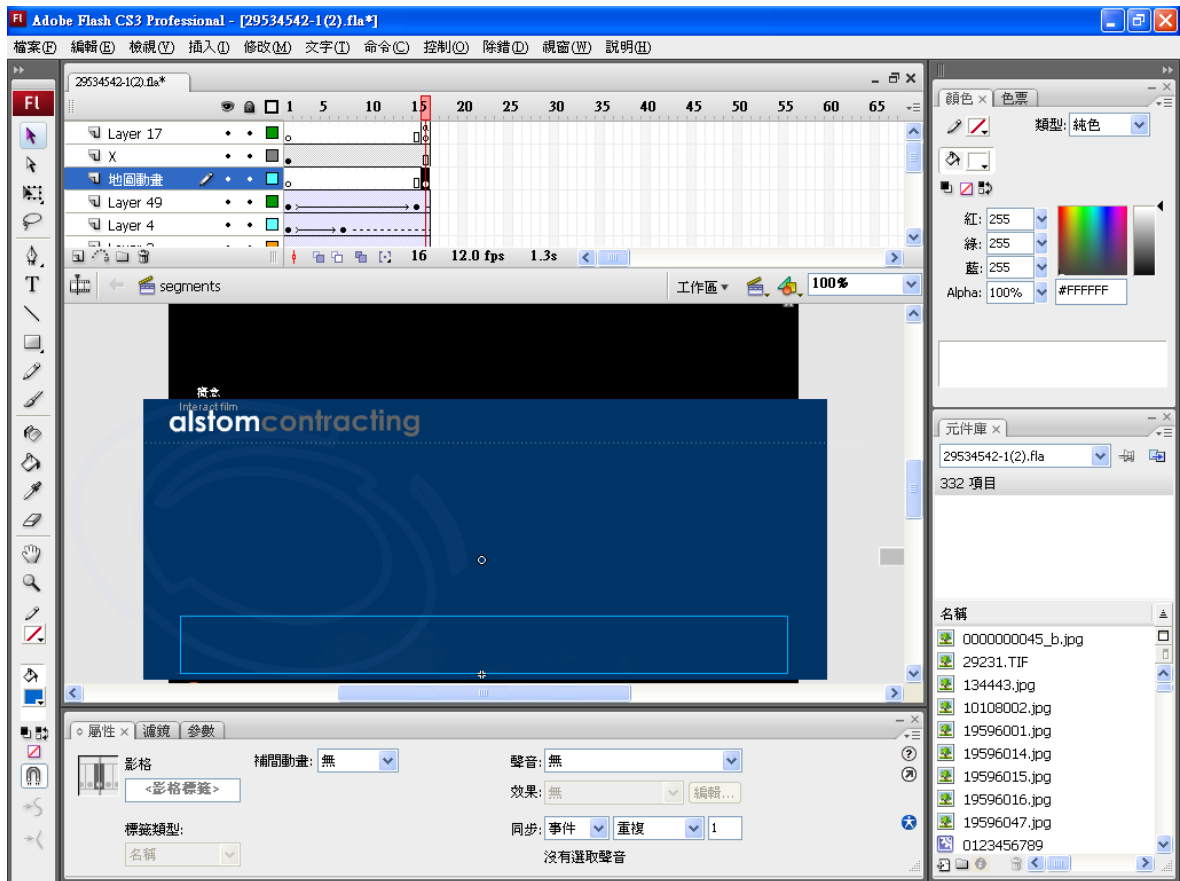


圖 3-27

## (2)地圖影片片段的內建影格

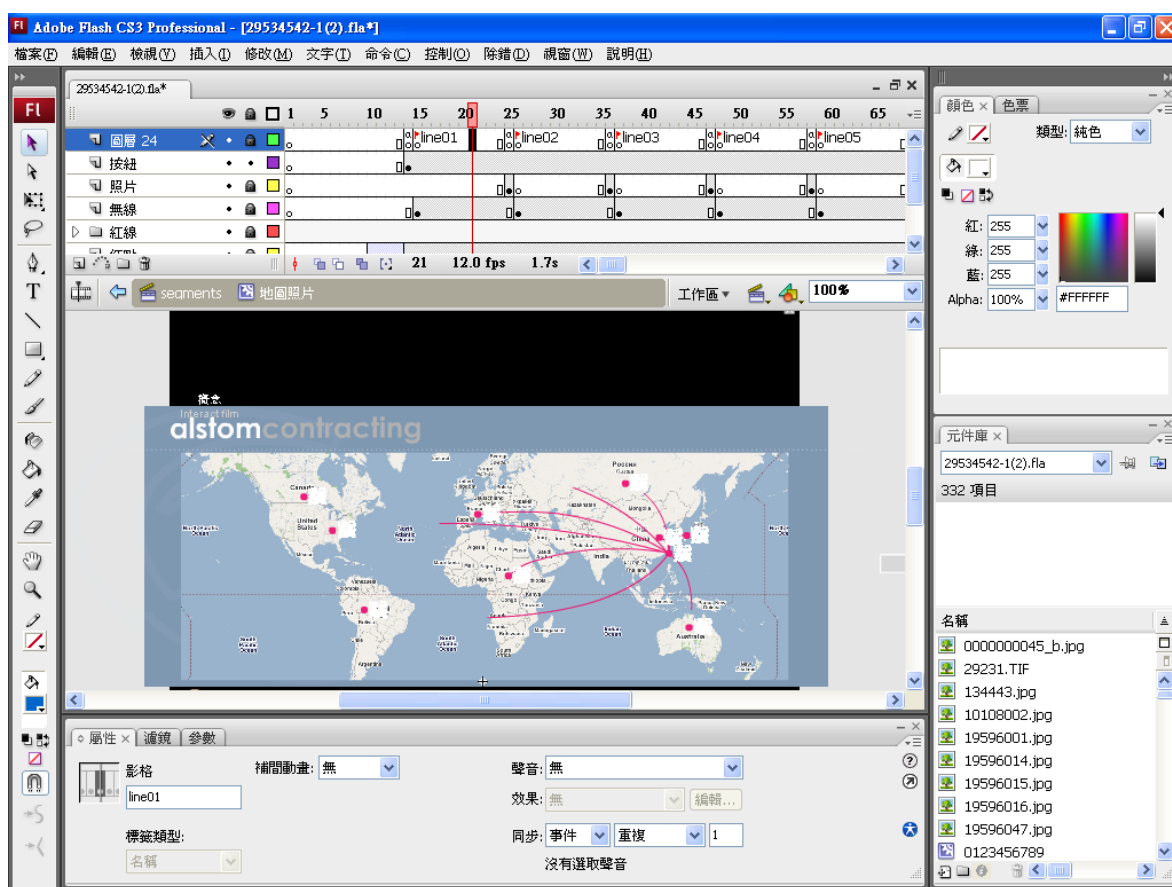


圖 3-28

## 4. 測驗系統

### (1) 程式碼定義答案格和答題數

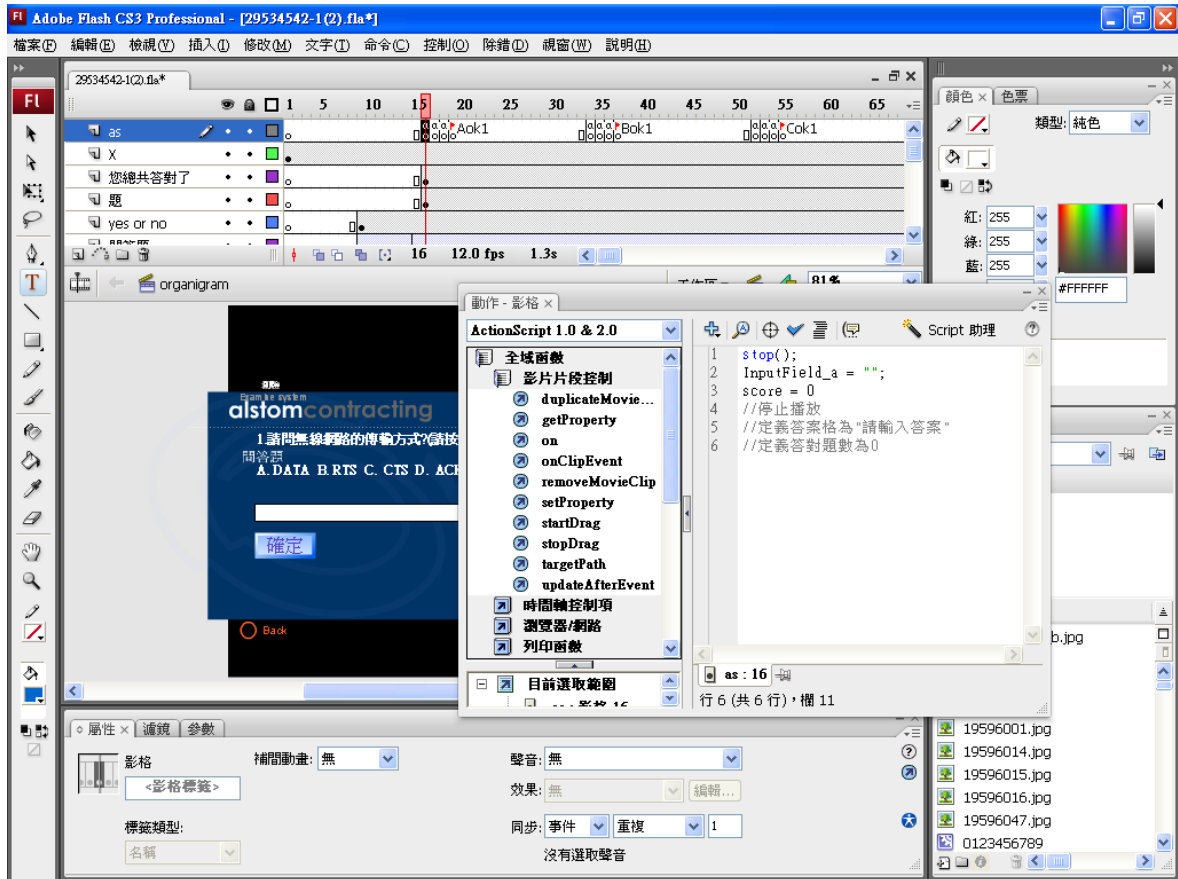


圖 3-29

## (2) 輸入錯誤時會顯現的文字

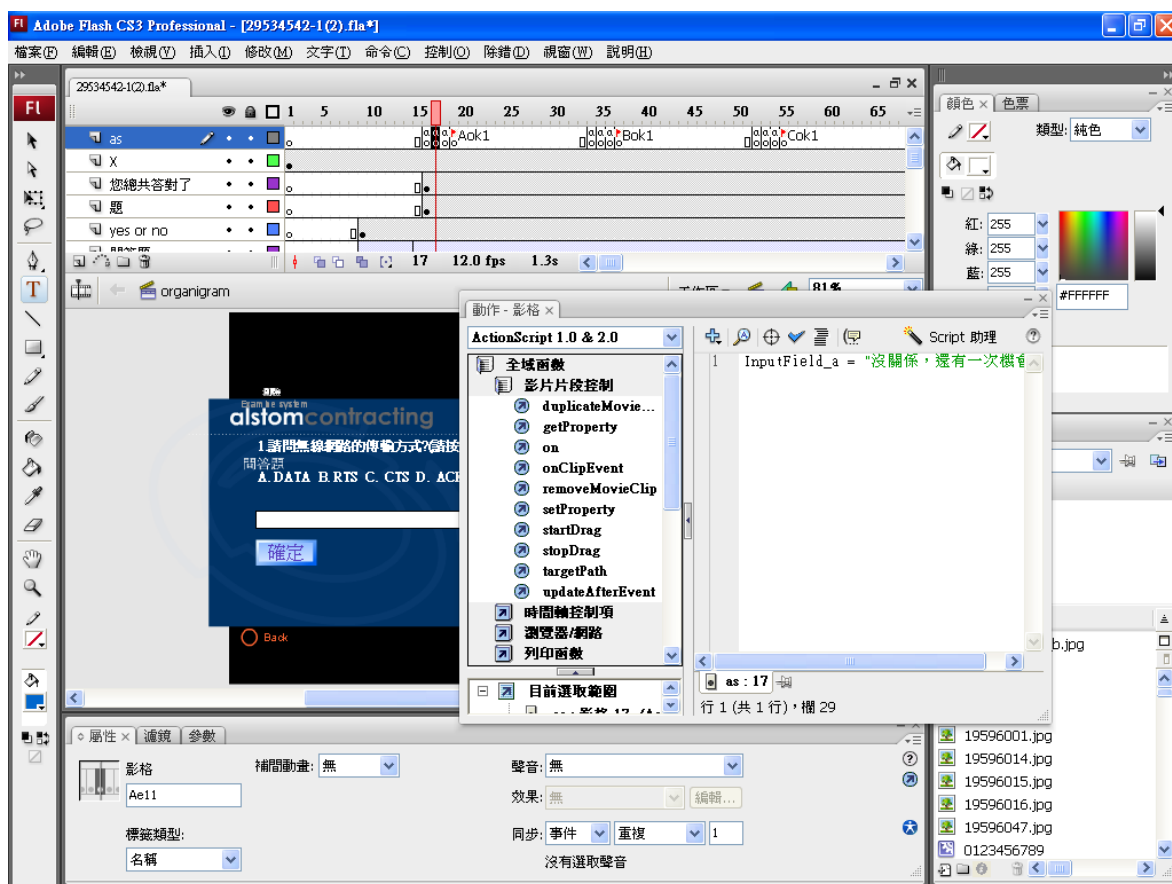


圖 3-30

(三)再輸入錯誤時會跑出答案

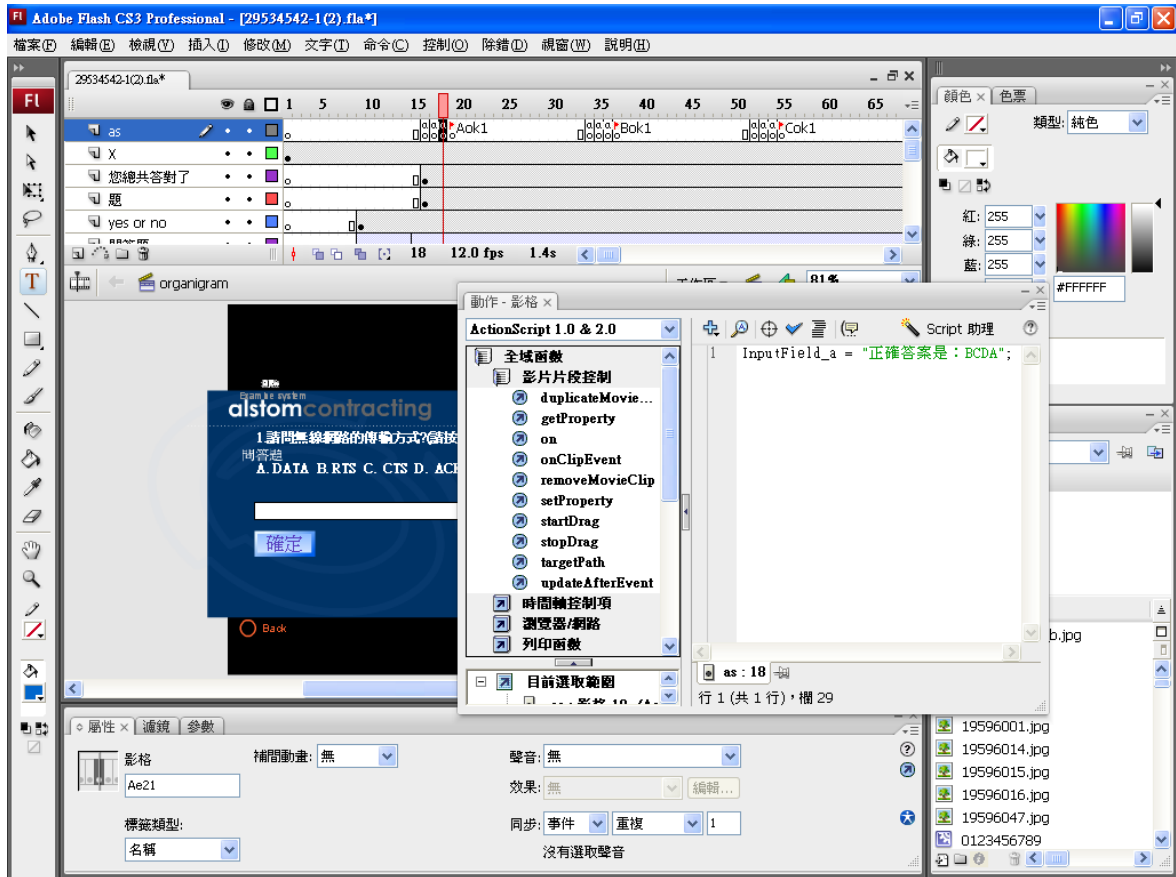


圖 3-31

## 第肆章 系統呈現

### 第一節 預期效能

其實這次的專題研究,是因為我們在瀏覽國外的 FLASH 做的網站時,有看到有人也是以 FLASH 選定一個主題,然後製作,它的內容跟整體規劃,包含一些特殊的聲光效果,五花八門的唯美動畫,都讓我們大受感動,也讓我們有想要效法的決心,對於我們來說,要製作出像樣的動畫,甚至是勉強可以讓使用者掏出時間稍微看一下的 FLASH,都讓我們花費很多心力,甚至是整體的規劃和架構,也是從這些國外的網站得來靈感,只是比起它們,我們所呈現出來的畢竟還是不夠唯美,聲光效果也不夠華麗,都讓我們的成品大打折扣,也是美中不足的地方。

總節來說,我們的內容不夠生動活潑,幸好也不至於淪落到死氣沉沉的刻板印象,簡言之,還有繼續朝更高目標攀爬的潛力,也期許我們能再以這個成品去做擴充,至少能達到跟 FLASH 網站所呈現出來的質感和功能,都是我們要在更努力該去突破的點。

## 第二節 系統效能

以整個架構來說,最讓我們感到較自豪的,大概是無線概念的世界地圖吧,想想當初為了這個構想,著實花了不少時間,而畫面上九個啟動點,每一個都可跟其它的啟動點作連結,連結成功時,會顯現此國家最具代表性的圖片。

以台灣為例,當從其它國家連到台灣時,會顯現 101 的圖片,而當你從台灣連出去時,可以看到日本的櫻花,美國紐約自由女神等等,利用簡單的表示法,把無線網路的概念徹底具體化出來,藉由圖像,把我們的觀念深深印在使用者的腦海中。



## 第五章 結論

由於無線網路漸漸成為未來趨勢，很多人對於無線網路的概念也都處於懵懵懂懂的階段，又覺得課本的東西太多太雜，對只想稍微接觸又不想深入的人來說，無法有效的認識無線網路，我們有了把無線的東西活生生的用活潑的手法呈現給這些人，讓只想接觸的人也能有個基本的認知。

所以我們採用 FLASH 系統來作開發，而且整體的規劃，包含架構內容，我們都希望能讓使用者有在瀏覽網站的感受，甚至利用 ActionScript 程式來讓我們的內容變的比較生動，也讓使用者能藉由生動的影片去記憶裡面的內容，達成教學相長學習效果。

對於我們成果的展現，只能說現階段來講，不論是內容還是整體的質感我們都還有上升的空間，也希望使用者能在裡面瀏覽學習之後，能夠對於無線網路有了進一步的認識，也希望各位針對裡面的缺失，能多給我們批評和指教。

## 參考文獻

2. 趙英傑著，2005年，FLASH8 專業版進階應用，第一版，上奇科技
3. 呂聰賢著，2007年，FLASH 創意動畫 FUN 手做，第一版，松崗出版社
4. 吳佳諺著，2006年，FLASH8 徹底研究，第一版，國立成功大學
5. <http://www.dynamicfactory.com/>，Pay Pal VERIFIED