

國立交通大學

多媒體工程研究所

碩士論文

移植多媒體講解呈現播放器於 Android



The System Porting and Implementation of a Multimedia Lecturing

Presentation Player under the Android Platform

研究生：黃詩婷

指導教授：陳登吉 教授

中華民國九十九年八月

移植多媒體講解呈現播放器於 Android 平台的製作

The System Porting and Implementation of a Multimedia Lecturing Presentation
Player under the Android Platform

研究生：黃詩婷

Student: Shih-Ting Huang

指導教授：陳登吉

Advisor: Dr. Deng-Jyi Chen

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

1896
Master

in

Computer Science

August 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年八月

移植多媒體講解呈現播放器於 Android 平台的製作

學生：黃詩婷

指導教授：陳登吉 博士

國立交通大學多媒體工程研究所碩士班

摘要

隨著科技的進步、無線網路的普及，行動裝置不斷的日新月異，多媒體加值服務越來越多樣化，使的行動裝置漸漸成為生活中不可或缺的工具。電子書閱讀器(eBooks)、平板電腦(Tablets)、智慧型手機(Smart Phones)等行動裝置正經歷著爆炸性的成長，造就了隨處都可上網，處處都可學習的環境。

市面上已有相當熱門的數位內容服務，例如：YouTube、iTunes 等服務，利用了完善的服務架構，讓使用者可以隨時隨地享受豐富的聲光視覺化多媒體服務。本研究探討其服務架構，設計多平台之行動數位學習服務架構，以讓學習者可使用不同的行動裝置，並透過網際網路的結合，達到行動學習(Mobile Learning)的目標。

目前 PC 上已有完善的數位學習資源，故可利用既有的數位教材相關工具，發展多平台之行動數位學習服務。然而，教材錄製工具需要較為強大的運算能力，不適合於行動裝置上使用。故本研究利用 PC 上既有的教材錄製工具錄製教材，將多媒體講解呈現播放器移植於行動裝置，並搭配數位學習部落格，讓學習者可以透過網路隨時隨地瀏覽數位教材。

本研究移植及製作一個多媒體講解呈現播放器於行動裝置，並以 Android 平台之手持裝置 HTC Hero 為例，探討 Android 平台與來源平台的系統資源、程式語法及支援程式庫的差異，實作情境感知內容協調的技術與內容呈現之效能改進，使 PC 平台製作的多媒體講解呈現檔可以於 Android 平台呈現觀看。

The System Porting and Implementation of a Multimedia Lecturing Presentation Player under the Android Platform

Student: *Shih-Ting Huang* Advisor: *Dr. Deng-Jyi Chen*

Department of Multimedia Engineering National Chiao Tung University

Abstract

With the fast steps of Information Technology innovation, network infrastructure became universally available. With these Information Technology implemented mobile devices such as eBooks, Tablets and smartphones, information can now be accessed at anytime, anywhere. The devices we use to gain such access are now tagged indispensable.

At present time, there are already many digital multimedia service providers existing in the market. YouTube and iTunes are two of the many successful stories. With the implementation of the maturely designed architecture, these providers are able to grant their service users to access and enjoy the multimedia namely any where any time through their mobile devices. In this research, we will start with analyzing and discussing the architecture used by these service providers, then continued by the design of the architectures and how they are implemented on the various platforms. At last, how the combination of the two assist scholars and learners to achieve the objective of Mobile Learning through their mobile devices.

As the modern computers are generally featured with matured e-learning systems, these systems can therefore be modified and referenced to assist in the development of e-learning on mobile platforms. However, as the multimedia lecturing creation tools require a great computing capability which no mobile device possesses, we only ported the multimedia lecturing presentation player onto mobile devices. With the e-learning blog, users can have access to the lecturing content through internet.

In this research, we have designed a multimedia lecturing presentation player and implemented to the selected platform, Android and test it on HTC Hero. In this section, we will analyze and discuss the differences of system resource, programming language and the

supportive libraries between the Android platform and the PC platform. We will also talk about the fine-toning processes such as context-aware content adaption mechanism and execution efficiency of the player, in attempt to present the PC platform made multimedia lecturing content on the Android platform through the player in a desired manner.



誌謝

承蒙指導教授陳登吉老師的細心指導與教誨，這兩年來，不論在研究上或是生活上都給了很大的鼓勵與指引，更時常與我們分享人生大道裡與生活經驗，讓我學到了很多書本上學不到的東西。感謝曾建超、孔崇旭老師耐心與不厭其煩的教導，很感謝有這麼多好老師的幫助，讓我可以順利的完成這兩年的課業及論文的研究，也讓我受益良多。老師，謝謝您。

接著，感謝在交大陪伴我、互相砥礪的同學、學長姐、學弟妹，還有許多鼓勵、幫助我的朋友們。在課業上，有你們不遺餘力的幫助；遇到困難時，也總是有你們安慰與鼓勵。因為有你們，豐富了我的生活。認識你們，是我的榮幸。

最後，感謝我的家人們，因為有你們的支持與栽培，成就了今天的我。謝謝你們讓我在沒有後顧之憂的環境下完成了碩士學位，未來我將會更加努力，以期有好的作為。



目錄

內容

摘要.....	i
誌謝.....	iv
目錄.....	v
表目錄.....	vii
圖目錄.....	viii
一、緒論.....	1
1.1 研究動機與目標.....	1
1.2 研究方法與步驟.....	2
1.3 章節概要.....	3
1.4 研究相關名詞.....	4
二、相關研究與探討.....	5
2.1 Android 平台簡介.....	5
2.2 Linux 平台與 Android 平台的差異.....	6
2.3 Android 相關工具介紹.....	6
2.4 DRM 驗證.....	7
2.5 情境感知內容協調技術.....	7
2.5.1 情境感知改編協調.....	8
2.5.2 內容協調.....	8
2.5.3 內容協調項目.....	9
三、系統分析與架構.....	11
3.1 教材錄製工具比較.....	11
3.2 教材錄製及播放架構.....	12
3.2.1 錄製與播放架構--以 PC 為例.....	12
3.2.2 錄製與播放架構--以 PC 及 Android 手持裝置為例.....	14
3.3 不同平台之播放端比較.....	15
3.4 播放器功能需求分析.....	15
3.5 播放器模組架構分析.....	17
3.6 程式移植策略.....	18
3.6.1 開發工具與語言.....	18
3.5.3 程式移植方式.....	19
3.6 多媒體講解呈現檔.....	21
四、系統實作.....	26
4.1 多媒體講解呈現播放器於手機端的播放流程.....	26

4.1.1 播放本機儲存端講解檔的流程圖	26
4.1.2 播放部落格上的講解檔之流程圖	27
4.1.3 bst 檔播放排程	28
4.1.4 bst 檔下載排程	28
4.2 基本的播放功能實作	29
4.2.1 檔案解析	30
4.2.2 資料解析	30
4.2.3 情境感知內容協調.....	33
4.2.4 內容呈現	34
4.3 DRM 驗證.....	36
4.4 網路結合相關	36
4.4.1 播放器與部落格結合.....	37
4.4.2 下載部落格上的講解檔.....	37
4.5 播放器效能改進	39
4.5.1 畫布更新時間計算方式.....	39
4.5.2 螢光筆混色方法.....	41
4.5.3 聲音檔案播放.....	42
五、系統展示	44
5.1 多媒體講解呈現教材的錄製與發佈.....	44
5.1.1 多媒體講解呈現教材的錄製	44
5.1.2 多媒體講解呈現教材的發佈	45
5.2 Windows 平台上播放	48
5.2.1 播放本機端的講解檔.....	48
5.2.2 播放部落格上的講解檔.....	50
5.3 Android 平台上播放	51
5.3.1 播放本機端的講解檔.....	52
5.3.2 播放部落格上的講解檔.....	54
5.3.3 功能展示	54
六、總結	57
6.1 結論	57
6.2 未來展望	57
參考文獻或資料.....	59

表目錄

表格 1：靜態協調與動態協調優缺點比較.....	9
表格 2：錄製工具測試比較環境.....	11
表格 3：錄製工具設定參數.....	11
表格 4：多媒體講解錄製工具比較.....	12
表格 5：不同平台播放端比較.....	15
表格 6：多媒體講解呈現播放器之功能分析.....	16
表格 7：多媒體講解呈現播放器功能分析比較.....	16
表格 8：開發工具與語言比較.....	19
表格 9：C/C++程式之元件替換方式.....	20
表格 10：Android API 對多媒體檔案的支援.....	20
表格 11：Java 程式之元件替換方式.....	21
表格 12：塗鴉動作實作方式列表.....	35
表格 13：原始畫布更新方式之延遲時間.....	40
表格 14：改進後畫布更新方式之延遲時間.....	41



圖目錄

圖 1：數位內容服務架構.....	1
圖 2：多平台行動數位學習服務架構.....	2
圖 3：Android 平台架構〔7〕〔9〕〔10〕.....	6
圖 4：錄製與播放架構(以 PC 為例).....	13
圖 5：錄製與播放架構(以 PC 及 Android 手持裝置為例).....	14
圖 6：Linux 平台之多媒體講解呈現播放器架構.....	17
圖 7：Android 平台多媒體講解呈現播放器架構.....	17
圖 8：程式移植方式.....	19
圖 9：多媒體講解呈現播放器架構之元件替換.....	21
圖 10：多媒體講解呈現檔架構.....	22
圖 11：bst 檔編碼流程.....	23
圖 12：場景圖資訊.....	24
圖 13：腳本系統之 script 檔內容.....	25
圖 14：本機儲存端講解檔之播放流程.....	26
圖 15：部落格上講解檔之播放流程.....	27
圖 16：bst 檔播放排程.....	28
圖 17：bst 檔播放流程.....	29
圖 18：bst 檔解碼流程.....	30
圖 19：解析 XML 運作流程--寫入資料結構.....	31
圖 20：解析 XML 運作流程--寫入 XML 檔.....	31
圖 21：(a) 課程發佈資訊資料結構 (b) 對應的 bst 頁面列表.....	31
圖 22：場景圖資訊關連圖.....	32
圖 23：腳本系統資訊關連圖.....	33
圖 24：播放資訊協調示意圖.....	33
圖 26：畫布分層更新示意圖.....	34
圖 25：多媒體圖片內容協調示意圖.....	34
圖 27：圖層結合方式.....	36
圖 28：DRM Client 模組運作流程.....	36
圖 29：播放器與部落格結合示意圖.....	37
圖 30：下載排程器運作流程.....	38
圖 31：檔案下載範例示意圖.....	39
圖 32：原始的畫布更新方式--預期結果.....	39
圖 33：原始畫布更新方式--實際結果示意圖.....	40
圖 34：畫布更新方式改進方法.....	40
圖 35：腳本資訊--螢光筆動作.....	41
圖 36：螢光筆混色--原始方法示意圖.....	42

圖 37：螢光筆混色--改進方法示意圖	42
圖 38：聲音播放--原始做法	42
圖 39：聲音播放--改進做法	43
圖 40：教材錄製--匯入投影片	44
圖 41：教材錄製--投影片載入成功	44
圖 42：教材錄製--錄製畫面	45
圖 43：教材發佈--發佈畫面	45
圖 44：發佈教材於部落格--選擇發佈的部落格.....	46
圖 45：發佈教材於部落格--輸入使用者帳號密碼.....	46
圖 46：發佈教材於部落格--填寫發佈資訊	46
圖 47：發佈教材於部落格--選擇面板資訊	47
圖 48：發佈教材於部落格--選擇主題	47
圖 49：發佈教材於本機端--填寫資訊	48
圖 50：發佈教材於本機端--填寫 DRM 資訊	48
圖 51：播放本機端講解檔(Windows 平台)--本機端資料夾	49
圖 52：播放本機端講解檔(Windows 平台)--DRM 驗證.....	49
圖 53：播放本機端講解檔(Windows 平台)--書籤.....	49
圖 54：播放本機端講解檔(Windows 平台)--播放畫面	50
圖 55：播放部落格上的講解檔(Windows 平台)--部落格畫面	50
圖 56：播放部落格上的講解檔(Windows 平台)--講解檔網路上的資訊	51
圖 57：播放講解檔(Android 平台)--開啟播放器	51
圖 58：播放講解檔(Android 平台)--選擇播放本機端/網路端的檔案..	52
圖 59：播放本機端的講解檔(Android 平台)--檔案選擇清單	52
圖 60：播放本機端的講解檔(Android 平台)--DRM 驗證.....	52
圖 61：播放本機端的講解檔(Android 平台)--書籤功能.....	53
圖 62：播放本機端的講解檔(Android 平台)--播放畫面	53
圖 63：播放本機端的講解檔(Android 平台)--塗鴉、滑鼠呈現畫面	53
圖 64：播放本機端的講解檔(Android 平台)--電子白板呈現畫面	54
圖 65：播放部落格上的講解檔(Android 平台)--部落格畫面	54
圖 66：播放部落格上的講解檔(Android 平台)--講解檔網路上的資訊 .	54
圖 67：播放器功能(Android 平台)--功能選單	55
圖 68：播放器功能(Android 平台)--播放控制列.....	55
圖 69：播放器功能(Android 平台)--課程清單	56
圖 70：播放器功能(Android 平台)--畫面比例、方向切換.....	56
圖 71：未來展望示意圖.....	58

一、緒論

1.1 研究動機與目標

隨著資訊科技的發達，網路基礎設備漸漸的普及化，造就了隨處都可上網，處處都可學習的環境。目前，電子書閱讀器(eBooks)、平板電腦(Tablets)、智慧型手機(Smart phones)等行動裝置正經歷著爆炸性的成長，不少廠商紛紛投入行動裝置的開發與設計，而多數行動裝置更具備了無線網路傳輸與多媒體播放的功能，行動裝置漸漸成為生活中不可或缺的工具。

目前市面上有一些相當熱門的數位內容服務，例如：YouTube、iTunes、facebook 等服務，讓使用者可以隨時隨地享受豐富的聲光視覺化多媒體服務，而其服務架構可整理如下圖 1。其中，YouTube 為目前全球最受歡迎的線上影片社群，使用者可使用工具編輯製作影片，並搭配行動裝置隨時隨地使用 YouTube 提供的服務。而 iTunes 提供影音播放及管理功能，並與底層的裝置結合，讓使用者可以直接透過 iPhone 或 iPod touch 等裝置使用 iTunes 提供的服務，並可連結到 iTunes store 享受更多元的服務。



圖 1：數位內容服務架構

本研究利用上述成功案例之服務架構，提供多平台的行動數位學習服務，讓學習者可以使用不同的行動裝置進行數位學習，並透過網際網路的結合，達到行動學習(Mobile Learning)的目標。

就數位學習而言，目前在 PC 上已有相當完善的資源。講解者只需載入製作好的多媒體簡報檔，搭配講解者的聲音與動作，便可製作一份吸引學習者觀看學習的多媒體講解呈現教材。然而，教材錄製工具需較為強大的運算能力，不適合於行動裝置上使用，故本研究利用 PC 上既有的教材錄製工具，再搭配數位學習部落格，讓講解者可以將製作好的教材傳送到數位學習部落格。並將多媒體講解呈現播放器移植於行動裝置，讓學習者可藉由不同的行動裝置隨時隨地瀏覽數位學習部落格，下載觀看教材。

本研究移植及製作一個多媒體講解呈現播放器於行動裝置，並選擇 2007 年底 Google 推出之 Android 平台做為移植的目標平台，其為一開放性、高移植性的平台，目前已廣泛被各式各樣的裝置所使用。而另一位實驗室同學，王尹暄選擇 WinCE 6.0 平台作為移植平台，以達到多平台之行動數位學習之目標。而在移植的過程中，探討來源平台與目標平台之系統資源、程式語法及支援的函式庫差異，並實作情境感知內容協調與內容呈現之效能改進，以呈現與 PC 端一致的講解內容。

1.2 研究方法與步驟

本研究利用圖 1 的架構，設計出多平台之行動數位學習之架構，如圖 2 所示：



圖 2：多平台行動數位學習服務架構

- 教材錄製工具維持在 Windows 平台
教材錄製工具需較強大的運算能力，不適合使用資源較為不足之行動裝置。
故講解者可使用原有的教材錄製工具錄製教材。
- 移植多媒體講解呈現播放器於行動裝置
學習者可透過 Android、WinCE 6.0 平台之行動裝置進行學習。
- 結合數位學習部落格
講解者可將教材分享於數位學習部落格，學習者可以隨時隨地觀看線上的教材，而達到無所不在的學習。

為了要讓 Android 行動裝置的使用者亦能觀看 Windows 錄製之講解檔，故需將多媒體講解呈現播放移植於 Android 平台，步驟大致如下：

- 了解來源端平台播放器之行為模式及播放器功能
藉由分析講解檔錄製端與播放端之架構和講解檔之架構，以了解播放器與其它相關平台(例如：數位學習部落格)之結合方式，及播放器之功能模組。
- 了解來源平台與目標平台之差異
包含了硬體資源的差異、系統支援差異等。
- 設計目標平台播放器之行為模式
藉由了解來源端平台播放器之行為模式及來源平台與目標平台之差異，設計符合目標平台播放器之行為模式。
- 擬定移植策略
分析來源平台與目標平台支援的程式語言、函式庫之差異，擬定程式移植的策略，並尋找函式庫、資料型態等替換方式。
- 系統實作
依照移植策略移植各個模組，盡量延用修改原本的程式碼，如無法延用修改，則盡量採用相同架構重新開發。
- 總結論文結論及未來發展方向

1.3 章節概要

第一章，緒論，由研究動機與目標引導出研究方法。

第二章，相關研究與探討，介紹相關的研究議題及技術。

第三章，系統分析與架構，分析 Android 平台與來源平台的系統資源、程式庫差異和播放器模組，擬定程式移植的策略。

第四章，系統實作，介紹各個功能的實作方法及運作方式。

第五章，系統展示，呈現播放器在 Android 平台上的功能範例。

第六章，總結，結論與未來展望。

1.4 研究相關名詞

- **數位學習 (Electronic Learning; E-Learning) [1]**
指學習者應用數位媒介學習的過程，其中數位媒介包括網際網路、企業網路、電腦、衛星廣播、錄音帶、錄影帶、互動式電視及光碟等。應用的範圍包括網路化學習、電腦化學習、虛擬教室及數位合作。
- **行動學習 (Mobile Learning; M-Learning) [2] [3]**
透過行動運算裝置來進行學習，行動運算裝置包含了智慧型手機(Smart Phone)、個人數位助理(PDA)等。通常是在數位學習中加入行動的元素，並透過無線網路，讓使用者可以隨時隨地進行學習活動。
- **程式移植(Program Porting) [4] [21]**
將來源平台上的程式移植到目標平台，讓來源平台上的程式，在目標平台上可以正常執行。通常，來源平台與目標平台可使用的資源不同，支援的程式語言與函式庫也不盡相同，故須將程式做轉換或尋找替代解決方案。
- **情境感知內容協調 (Context-aware Content Adaptation) [5]**
內容協調是將原始的內容經過協調轉換到適合目標平台觀看之內容，並達到最佳觀看的品質。而情境感知內容協調為在協調的過程中，加入情境(Context)作為考慮的因素，通常這些情境為：目標載具的能力、網路狀況和使用者的偏好等。
- **數位著作權管理(Digital Rights Management; DRM) [6]**
DRM為用來保護數位出版品的一項技術，透過此數位權限管理，用來控制哪些使用者擁有這些數位出版品的讀取、存取等權限。

二、相關研究與探討

2.1 Android平台簡介 [7] [8] [9]

Android 平台為 2007 年底由 Google 推出的作業平台，在推出的同時組成了開放手持設備聯盟(Open Handset Alliance; OHA)，包含許多的硬體製造商、軟體開發商及電信公司等，皆加入此聯盟致力於 Android 平台的發展。

Android 平台為開放式之平台，讓程式開發人員、使用者可以自由地在此平台上開發、安裝軟體。此平台並免費提供給手機製造商及軟體開發商使用，不收取授權費用，降低了手機製造的成本。

由於 Android 為高移植性、低硬體需求之平台，已廣泛移植到各種不同的裝置，目前應用的領域包含了智慧型手機、數位家庭及嵌入式系統等。其中，智慧型手機為最大的應用領域，包括 HTC、Google、MOTO、SAMSUNG、Sony Ericsson、Acer 等品牌都推出了 Android 平台之智慧型手機，Android 平台手機的市占率逐年攀升。

此外，在 Android 平台上可輕易地與既有的 Google Service 做結合，而未來將與雲端技術最結合，必會使應用服務的型態產生很大的改革。

Android 平台架構

Android 平台架構採用分層設計，將不同階層的相依性降低，方便於獨立開發與除錯等目的。以下為各層的介紹：[7][9]

■ Applications

此層包含了重要的一些核心應用程式，像是email client, calendar, maps, browser等。這些應用程式都是由Java語言開發設計的。

■ Application Framework

此層提供了豐富多元的Framework API供應用程式使用，讓應用程式可以很輕易與底層的硬體做溝通，並可以與網路上的資源做結合。

■ Libraries

包含了一系列的C/C++函式庫，主要是供Android System使用的。

■ Android Runtime

使用自行研發Android Runtime來執行程式，以減少記憶體的使用為其設計目標，讓系統在執行Android應用程式時以最佳化的方式執行。

■ Linux Kernel

Android平台採用修改過後的Linux 2.6為其核心系統。此層提供了核心系統服務，並包含軟體與硬體之間溝通的驅動程式。



圖 3：Android 平台架構 [7][9][10]

2.2 Linux 平台與 Android 平台的差異

Linux 桌面系統有兩大主流，一是 GNOME，另一是 KDE。Android 平台雖然是採用 Linux 2.6 為其核心系統，但許多部分採用不同的解決方案，如圖 3 黃色框起來的部分即為與 GNOME 桌面系統相異處。[9][10]

舉例來說，GNOME 桌面系統是以 GTK+[11] 這套圖形介面函式庫來開發，並可使用 glibc、gstreamer 等函式庫。但在 Android 平台上，GTK+、glibc 都被取代了。因此，須找尋其它的替代方式。

Android 使用 Bionic Libc 作為系統的 C Library，而不使用 glibc。與 glibc 相比，Bionic Libc 以下的特點：[9]

1. 採用 BSD License，而不是 glibc 的 GPL License。
2. 大小只有大約 200k，比 glibc 差不多小一半，且比 glibc 更快。
3. 實現了一個更小、更快的 pthread。
4. 提供了一些 Android 所需要的重要函數，例如:LOGI。

Bionic Libc 的大小較小、執行效率較高，比 glibc 更適合使用於嵌入式 Linux 系統。然而，卻移除了一些『C++標準函式庫』，使程式的移植及開發變得更為困難。

2.3 Android 相關工具介紹

■ STLport [12]

STLport為一套ANSI C++標準函式庫實現，可於多種平台上使用，並免費提供程式開發人員使用。

■ libutils [9]

libutils是Android的底層函式庫，是以C++實現的。大部分的Android底層函式庫都是基於libutils開發的，故其相當重要。這個函式庫主要分為兩部分，一部分為底層的工具，另一部分為實現Process溝通的機制。

■ zlib [9][13]

zlib是提供資料壓縮、解壓縮用的函式庫，並為使用zlib授權的自由軟體。此函式庫普遍為許多軟體所使用，像是Linux使用它做為網路協定的壓縮、檔案系統的壓縮以及開機時解壓縮的核心。

■ Libxml2 [14]

Libxml2為一個C語言的XML文件解析器，其為Gnome project的一部分，而且為基於MIT License的免費開源軟體。除了C語言的版本外，還支援了C++、PHP等語言的連結。它還擁有跨平台的特性，可在Linux, Unix, Windows, MacOS等平台上運行。

■ JNI [9][15]

Java Native Interface是Java平台上的一種標準介面規範，在Android平台上，可以利用它做為Java application與native code溝通的介面。

2.4 DRM 驗證

DRM(Digital Rights Managment) [6] 為用來保護數位出版品的一項技術，透過此數位權限管理，可以用來控制哪些使用者擁有這些數位出版品的讀取、存取等權限。

一個典型的DRM系統的架構[6]為使用者可以由Content Server下載Content或是將Content分享給其它人，但是如果Content是受到DRM保護的話，則必須要連接到License Server取得License才可行使其擁有的權限。而License Server則是負責產生、管理權限，限制哪些使用者可以擁有此份Content的哪些權限。而呈現Content的Devices則必須支援DRM系統，與DRM系統進行溝通並擁有解釋License的能力。

在本研究中將License Server稱之為DRM Server。我們把用來將Content解密的金鑰放在DRM Server上，並設定權限規則。而Devices播放端要取得解密的金鑰前，則需與DRM Server進行溝通並進行驗證，通過驗證取得金鑰後才可以觀看Content。

2.5 情境感知內容協調技術

多媒體的內容協調技術一直在電腦科學領域中被研究探討，並經常應用於異質性行動裝置的多媒體內容呈現。本研究為了將一般電腦上的多媒體講解教材呈現於手持行動裝置，故須先探討異質性平台間的多媒體內容協調技術，以求得良好的呈現品質。

本章節會先介紹何謂以內容感知為基礎的轉換，再介紹內容協調(Content adaptation)的種類。

2.5.1 情境感知改編協調

情境(Context) [16] [17] [18] [19] 是一可以用來描述任何物件的表徵資訊，此物件可以為人物、地點或任何東西，只要這些物件與使用者、應用程式之間有所關聯且互相牽引著。

一個系統如果為情境感知(Context-aware) [16][19]，意指有能力使用或擷取情境(Context)資訊，並提供相對應的資訊或服務給使用者。

舉例來說，一份多媒體內容可能在不同的裝置上觀看。因為螢幕大小的不同，必須要經過特別處理才可以正常觀看(例如：縮小畫面)。或者是CPU、記憶體或codec不足，造成某些特定格式的檔案不能播放，而必須尋找其它的解決方式(例如：檔案格式的轉換)。

這些影響多媒體播放的品質，影響使用者觀看、操作應用程式等因素，都可稱為情境資訊。而一個系統有能力去處理這些情境資訊並提供對應的處理，可稱之為情境感知。

2.5.2 內容協調

內容協調方式 [16] [21]

■ 靜態協調(static)

事先針對各種異質平台協調轉換各種出適合的版本內容，並將各種版本內容儲存於資料庫中。當實際使用於異質平台時，再從資料庫當中挑選出適合使用的版本內容使用。故需先了解可能使用的異質平台之類型，並須額外的儲存空間。

■ 動態協調(dynamic)

當實際需要使用時才做協調轉換的動作，不需浪費的儲存空間，但使用者需等待轉換時間。

	靜態協調	動態協調
優點	呈現速度比較快	呈現速度比較慢
缺點	須額外的儲存空間 目標平台固定	不需額外的儲存空間 目標平台較有彈性

表格 1：靜態協調與動態協調優缺點比較

內容協調運作架構

一般的改編協調工作可能是由客戶端、伺服器端、代理伺服器或混和的方式實作。不同的設計有其優缺點，可依造系統的環境等因素做選擇。以下介紹不同的改編協調運作架構：〔16〕〔22〕

- 伺服器為基礎的改編協調(Server-based adaptation)
由媒體伺服器負責處理分析所有情境資訊，並選擇適當的轉換方式。通常會提供介面讓內容的製作者事先瀏覽協調完的效果。優點為可以使用動態協調或靜態協調以達到最好的協調效果。缺點為伺服器為了要實作協調的工作，而增加伺服器的負擔。
- 客戶端為基礎的改編協調(Client-based adaptation)
客戶端依賴裝置本身的支援能力，轉換協調出最適合此裝置呈現的內容。優點為伺服器不需要了解客戶端的支援能力。缺點為客戶端需有強大的運算能力，否則一些複雜性高的改編協調可能會無法執行或有呈現延遲的現象。
- 代理伺服器為基礎的改編協調(Proxy-based adaptation)
客戶端透過代理伺服器進行連線，代理伺服器會依照伺服器回傳的資訊決定改編協調的方式，將內容改編協調後的內容回傳給客戶端。其優點為不需要增加伺服器的負擔，容易與既有系統做結合。缺點為內容的製作者較難控制轉換出的效果。

由於本研究的重點為移植多媒體講解呈現播放器於手持裝置，故目前僅採用客戶端為基礎的改編協調。然而手持裝置的CPU能力不足，一些複雜的改編協調不適合執行於手持裝置（例如：影片檔案的轉換），故目前尚未實作。

2.5.3 內容協調項目

異質平台間的內容協調可以分為兩類，多媒體內容協調及畫面協調。〔20〕〔21〕

- 多媒體內容協調
由於不同的平台對多媒體檔案的支援可能有所不同，故需對多媒體內容做轉換。可分為解析度轉換和格式轉換。

- ◆ 解析度轉換：將圖片或影片的解析度轉換到符合目標平台的解析度，通常採用等比例縮放。
- ◆ 格式轉換：將目標平台不支援的檔案格式作轉換，轉換成目標平台可支援的格式。

■ 畫面協調

由於異質平台的畫面呈現大小可能不同，使用者習慣也可能不同，故必須針對硬體規格及使用情境對畫面做協調的動作。一般分為以下三種：

- ◆ 座標轉換：將來源平台的座標系統轉換為目標平台的座標系統。轉換的方式採取等比例縮小的方式。

舉例來說，來源平台為一般的桌上型電腦，其螢幕解析度為1024*768；而目標平台是HTC Hero，其螢幕解析度為480*320。則

來源平台的二維座標系統為 $\{X, Y \mid 0 < X < 720, 0 < Y < 540\}$ ，
目標平台的二維座標系統為 $\{X', Y' \mid 0 < X' < 480, 0 < Y' < 320\}$ ，
則轉換的公式為： $X' = X * (480 / 720)$ ， $Y' = Y * (320 / 540)$ 。

- ◆ 畫面分頁：將來源平台較大的頁面分成多個小頁面呈現於目標平台，故目標平台的使用者需藉由翻頁的方式瀏覽。
- ◆ 畫面移動：將來源平台的內容部分地呈現於目標平台，而目標平台的使用者需經由拖拉的方式觀看其它未呈現的部分。

本研究目前僅採用座標轉換的方式，而未來可以加入畫面移動的方式，讓使用者可自由地將畫面放大、縮小選擇觀看範圍。

三、系統分析與架構

首先，分析 Windows 平台上已有的教材錄製工具，比較其功能及製作的多媒體講解呈現檔，選擇適合移植的多媒體講解呈現播放器與搭配的數位學習部落格。

再來，分析 Windows 平台上多媒體講解呈現教材的錄製端與播放端，了解講解者如何使用教材錄製工具錄製、發佈教材；學習者如何瀏覽選擇、播放教材。再以 Android 手持裝置為例，設計出適合行動裝置之播放流程，並分析播放端的功能需求。

接著，分析不同播放端之差異，了解移植需注意的事項。最後，了解多媒體講解呈現播放器的架構，分析程式庫差異，擬定程式移植的策略及移植的方法，並藉由分析多媒體講解呈現教材的檔案架構，了解實作方法。

3.1 教材錄製工具比較

本研究先分析比較 Windows 平台上已有的教材錄製工具，再藉此選擇適合移植的多媒體講解呈現播放器與搭配的數位學習部落格。挑選三種不同技術開發的教材錄製工具〔24〕〔25〕〔26〕，比較其功能及製作的多媒體講解呈現教材，並參考〔20〕的測試方式，重新針對最新版本的教材錄製工具做測試。
錄製環境：

記憶體	1.99 GB
CPU	Intel Core 2 CPU · 1.66 GHz · 1.67GHz
作業系統	Windows XP Professional · Service Pack 3
螢幕解析度	1024x768

表格 2：錄製工具測試比較環境

錄製的投影片皆為同一份投影片(350KB)，並設定以下的錄製參數：

錄製參數	錄製工具	智勝國際 講解手 3.3	台灣數位學習科技 PowerCam5	訊連科技 串流大師4.0
投影片頁數		30	30	30
預計錄製時間		5:00	5:00	5:00
音訊壓縮設定		32Kbps, Stereo	32Kbps, Mono	32Kbps, Stereo

表格 3：錄製工具設定參數

測試結果：

工具 項目	智勝國際 講解手 3.3	台灣數位學習科技 PowerCam5	訊連科技 串流大師4.0
錄製方式	物件化 (投影片、繪圖、影音)	影像錄製 (Video Recording)	以DHTML的方式儲存 (投影片、繪圖、影音)
支援匯入 PowerPoint	有	有	有
預估一個小時後呈現 檔案壓縮後大小	約略49.8 MB	約略88.42 MB	約略205.2MB
網路分享	有	有	有
部落格結合	有	無	無
DRM結合	有	無	無

表格 4：多媒體講解錄製工具比較

依照上表的分析的結果，三種錄製工具都有支援常見的 PowerPoint 檔案格式。但進一步測試，發現智勝國際講解手 3.3 錄製出來的檔案大小最小，而行動裝置設備之容量較小，無線網路的傳輸速度最慢，較為適合使用在行動裝置上。而無線網路傳輸的風險較高，多媒體講解呈現教材如與 DRM 結合，可降低檔案傳輸的風險。再來，雖然三種教材錄製工具皆有網路分享機制，但只有智勝國際講解手 3.3 有跟部落格、DRM 結合，可降低軟體的複雜度，使用者可以輕易地享有這些功能。

因此，本研究採用『智勝國際講解手』做為教材的錄製工具、『智勝國際講解手播放器』做為移植播放器的來源，並搭配『樣板國度部落格』做為數位教材之取得來源平台。

實驗室學長姐已將 Windows 平台的『智勝國際講解手播放器』移植於採用 Linux 平台 [21][23]，而 Android 平台使用修改過的 Linux 2.6 為其核心系統，考慮到平台的相似性，故選擇將 Linux 平台上的講解手播放器移植於 Android 平台。

3.2 教材錄製及播放架構

本小節將分析 PC 上教材錄製端與播放端的系統架構，並設計出適合 Android 平台之手持裝置使用的錄製與播放架構

3.2.1 錄製與播放架構--以 PC 為例

本研究使用的『智勝國際講解手』錄製工具除了錄製多媒體講解教材之外，還搭配了數位學習部落格上傳分享教材，與 DRM Server 的檔案保護機制。

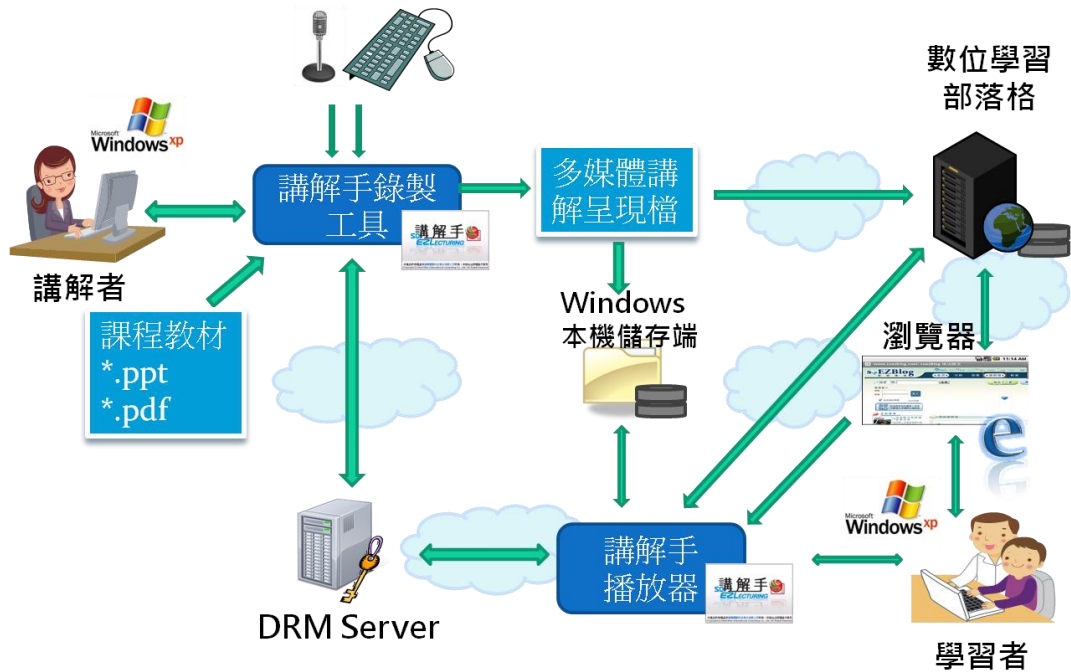


圖 4：錄製與播放架構(以 PC 為例)

■ 錄製端操作流程如下：

1. 講解者使用講解手錄製工具載入 PowerPoint 教材或 PDF 教材，並選擇收音的麥克風之後，開始錄製多媒體講解呈現教材。
2. 在錄製的過程當中，講解者可以使用滑鼠、鍵盤在教材上做講解注釋動作，並利用麥克風錄製講解聲音。
3. 錄製完畢，講解者可以選擇連線到 DRM server 以加入多媒體講解呈現教材保密的資訊。
4. 最後，講解者可以選擇將錄製好之多媒體講解呈現教材發佈到本機儲存端或經由網路上傳分享到數位學習部落格。

■ 播放端的操作流程分為兩個部分：

(一)從本機儲存端播放課程

1. 學習者瀏覽本機端的多媒體講解呈現教材，點選之後會自動啟動講解手播放器。
2. 學習者選擇的多媒體講解呈現教材在發佈時如有加入 DRM 資訊的話，則學習者必須輸入帳號密碼，播放器會自動連線到 DRM server 做認證。
3. 若認證成功，講解手播放器就會開始播放多媒體講解呈現教材。

(二)播放數位學習部落格的課程

1. 學習者藉由 IE 瀏覽器連線到數位學習部落格。
2. 學習者瀏覽數位學習部落格上面的教材資訊並點選播放。
3. 瀏覽器會會呼叫講解手播放器開啟。
4. 講解手播放器便會連線到數位學習開始下載多媒體講解呈現教材。

5. 如果學習者選擇的多媒體講解呈現教材在發佈時有加入 DRM 資訊的話，則學習者必須輸入帳號密碼，播放器會自動連線到 DRM server 做認證。
6. 若認證成功，講解手播放器就會開始播放多媒體講解呈現教材。

3.2.2 錄製與播放架構--以 PC 及 Android 手持裝置為例

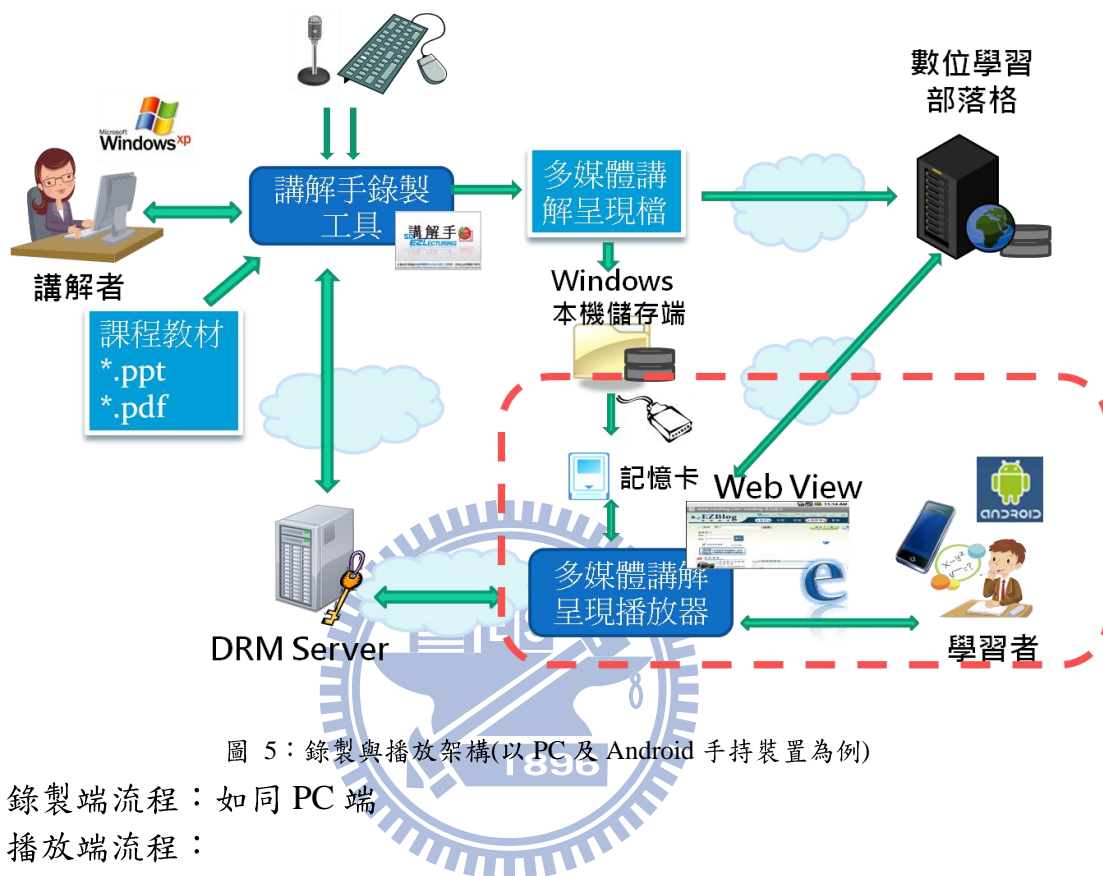


圖 5：錄製與播放架構(以 PC 及 Android 手持裝置為例)

- 錄製端流程：如同 PC 端
- 播放端流程：

(一) 從本機儲存端播放課程

1. 學習者可藉由傳輸工具將多媒體呈現教材傳送到手機記憶卡。
2. 學習者開啟講解手播放器後，選擇瀏覽本機的教材，並點選要播放的教材。
3. 學習者選擇的多媒體講解呈現教材在發佈時有加入 DRM 資訊的話，則學習者必須輸入帳號密碼，播放器會自動連線到 DRM server 做認證。
4. 若認證成功，多媒體講解呈現播放器就會開始播放多媒體講解呈現教材。

(二) 播放數位學習部落格的課程

1. 學習者開啟多媒體講解呈現播放器後，選擇播放網路上的教材。
2. 多媒體講解呈現播放器會由內嵌的網頁連線到數位學習部落格。
3. 學習者瀏覽數位學習部落格上面的教材資訊並點選播放。
4. 多媒體講解呈現播放器便會連線到數位學習開始下載多媒體講解呈現教材。
5. 如果學習者選擇的多媒體講解呈現教材在發佈時有加入 DRM 資訊的話，則學習者必須輸入帳號密碼，播放器會自動連線到 DRM server 做認證。
6. 若認證成功，多媒體講解呈現播放器就會開始播放多媒體講解呈現教材。

3.3 不同平台之播放端比較

本研究選用 Android 1.5 版本做為移植的平台，實作於 HTC hero。而分析比較結果如下：

- 手持裝置之記憶體容量及 CPU 較弱，故須針對播放器作效能的改進。
- 手持裝置之螢幕解析度較小，原始畫面須經內容協調才可呈現。
- Android 平台之 Unicode 編碼與 Windows 平台不同，故須做編碼轉換。
- Android 平台上支援之瀏覽器與 Linux、Windows 平台不同，故與部落格結合方式需要做改變。

	Windows平台 PC播放端	Linux平台 PC播放端	Android平台 手持裝置播放端	WinCE平台 手持裝置播放端
作業系統	MS Windows XP	Fedora Linux 8	Android 1.5	WinCE 6.0
記憶體容量	大 512MB~4GB	大 512MB~4GB (EeePc 701為512MB)	小 512MB (HTC Hero)	小 256MB (WinCE之模擬器)
CPU運算速度	快 1GHz~3.6GHz (一般PC)	快 1GHz~3.6GHz (EeePc 701為2.2GHZ)	慢 528MHz (HTC Hero)	慢 400MHz (WinCE之模擬器)
講解畫面解析度	720* 540	600 * 360 (EeePc 701)	480*320 (HTC Hero)	640*480 (WinCE之模擬器 ARM920T)
Unicode編碼	UTF-16	UTF-8	UTF-8	UTF-8
目前選擇 搭配的瀏覽器	IE	Firefox	WebKit	IE

表格 5：不同平台播放端比較

3.4 播放器功能需求分析

現有Windows平台上多媒體講解呈現播放器主要分為三類主要功能，如表6所示。

功能分類	功能	詳細說明
基本播放器功能	播放功能	播放、暫停、上一頁、下一頁、停止、全螢幕切換、塗鴉、電子白板等
	資料保護	將解出來的檔案放置在記憶體中

	書籤功能	當使用者結束觀看多媒體講解呈現教材後，播放器會自動紀錄觀看的位置紀錄於書籤內。當下一次開啟此份教材時，會讓使用者選擇是否要由加入書籤的位置開始觀看
DRM驗證	DRM驗證	如果要觀看的教材有DRM 保護，會提示使用者輸入帳號、密碼來完成驗證流程
與部落格結合	檔案下載排程	背景預先下載、隨選下載、續傳功能
	網路連線功能	可以透過Http協定的get method來取得網路上的資料、具備檔案續傳功能
	與網頁瀏覽器整合	利用ActiveX與Internet Explorer結合

表格 6：多媒體講解呈現播放器之功能分析

在Linux平台，除了書籤功能沒有實作外，其它功能皆有實作。而Android平台上多媒體講解呈現播放器，包含了所有Linux播放器具備的功能外，增加了以下兩點功能：

- 書籤功能：
方便使用者找到上一次最後播放到的位置。
- 直式/橫式螢幕的切換：
方便學習者可以隨意的轉換手機方向，以符合不同學習者使用習慣。

功能分類	功能	Windows 平台是否支援?	Linux 平台是否支援?	Android 平台是否支援?
基本播放器	播放功能	是	是	是，增加直式/橫式螢幕切換
	資料保護	是	是	是
	書籤功能	是	否	是
DRM 驗證	DRM 驗證	是	是	是
與部落格結合	檔案下載排程	是	是	是
	網路連線功能	是	是	是
	與網頁瀏覽器整合	是 利用 ActiveX 與 Internet Explorer 結合	是 利用 Firefox Plugin 與 Mozilla Firefox 結合	是 利用播放器內嵌網頁與 WebKit 結合

表格 7：多媒體講解呈現播放器功能分析比較

3.5 播放器模組架構分析

由於 Android 平台之播放器是由 Linux 平台的播放器移植而來，故先分析 Linux 平台的播放器模組架構，再設計 Android 平台的播放器模組架構。

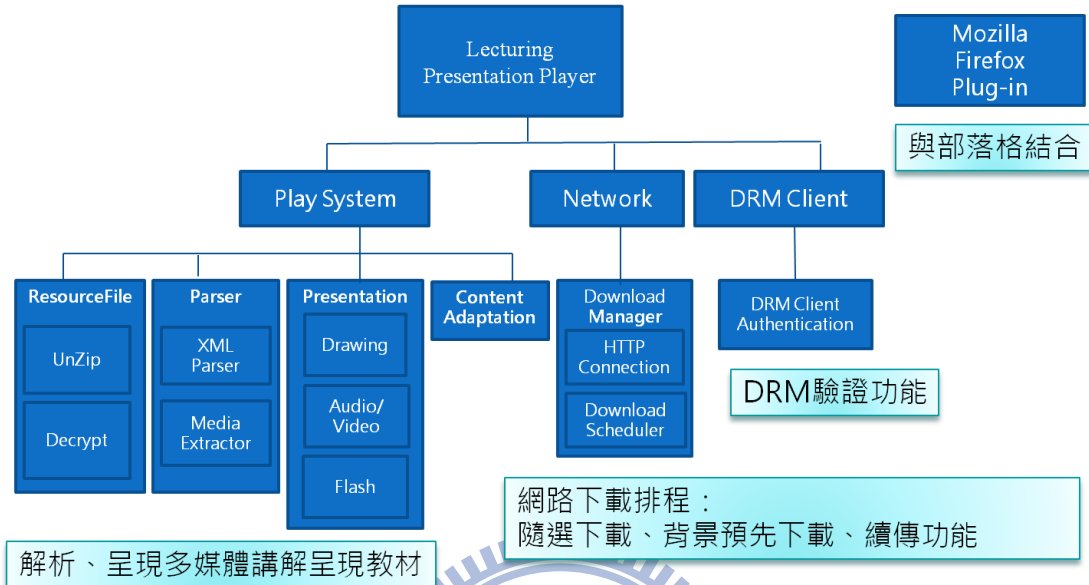


圖 6：Linux 平台之多媒體講解呈現播放器架構

藉由由此播放器架構及上一節播放器功能需求分析，設計 Android 平台上多媒體講解呈現播放器模組架構圖：

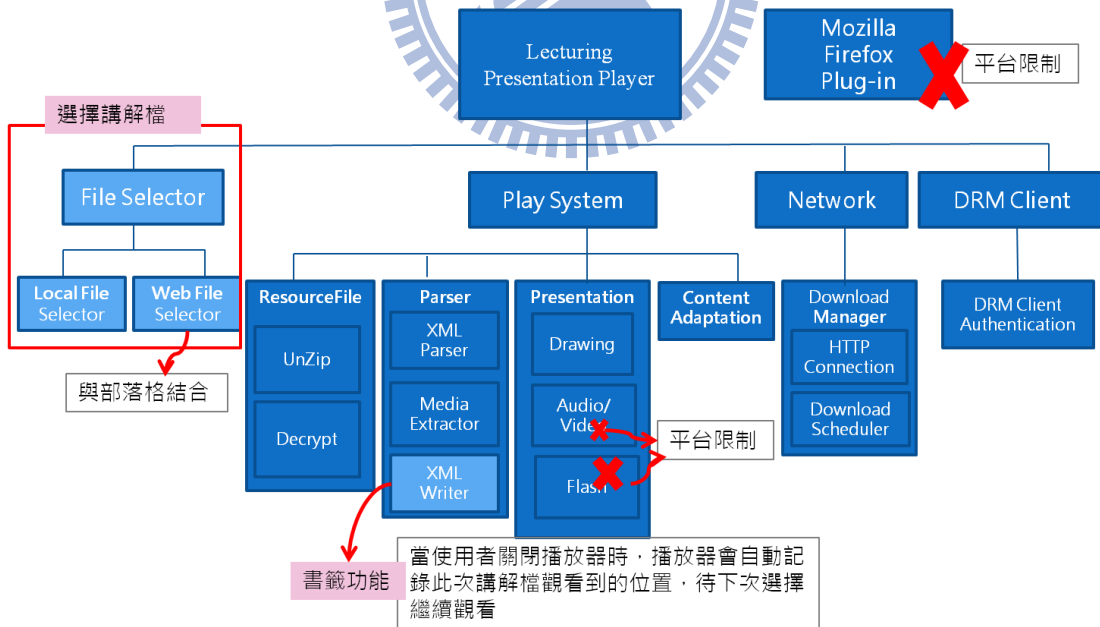


圖 7：Android 平台多媒體講解呈現播放器架構

在 Android 平台上，移掉了 Mozilla Firefox Plugin 模組，並於播放器內加入 Web View Client 與部落格結合。為了實作書籤功能，加入了 XML Writer 模組。而由於 Android 平台的限制，講解檔內的影片與 Flash 無法順利播放，目前尚未實作。各模組的功能介紹如下：

- File Selector 模組
 - ◆ Local File Selector
讓使用者選擇本機端的講解檔。
 - ◆ Web File Selector
播放器與數位學習部落格的結合。讓學習者可以使用內嵌網頁瀏覽點選數位學習部落格內的教材。當使用者點選要播放的教材後，此模組將會將檔案下載的位置傳送給 Network 模組進行檔案的下載。
- PlaySystem 模組
負責處理多媒體講解呈現教材、解析 XML 檔案、播放呈現和 UI 互動的功能。此模組亦整合了控制播放的 UI 介面，藉由使用者的命令來控制 Presentation 模組的播放與呈現。
 - ◆ ResourceFile 模組
處理多媒體講解呈現教材。將封裝的檔案進行解壓縮、解密，最後將資料寫入記憶卡或記憶體裡面。
 - ◆ Parser 模組
解譯 XML 檔案。由於多媒體講解呈現教材裡的資料紀錄是採用 XML 檔案格式，故需使用 XML parser 進行 XML 檔案的解析，並將解析完的資料儲存在類似資料庫管理的 Media Extractor 當中，以供 Presentation 模組做內容呈現的部分。
 - ◆ Presentation 模組
呈現多媒體講解呈現教材的內容。多媒體講解呈現教材包含多種檔案格式，故 Presentation 模組包含了處理、呈現不同多媒體檔案類型的子模組。Presentation 模組會啟動一個 Thread 用來計算播放的時間，並依照播放的時間控制內容的呈現。
- Network 模組
下載多媒體講解呈現教材。當接收到檔案下載的位置後，會先使用 Download Scheduler 模組負責設計下載排程，再呼叫 HTTP Connection 模組進行檔案的下載。
- DRM Client 模組
DRM 的驗證。包括提供 DRM Client 的 UI 介面供使用者輸入帳號與密碼，並使用 DRM Client authentication 模組連線到 DRM Server 進行 DRM 的認證。

3.6 程式移植策略

3.6.1 開發工具與語言

在 Android 平台支援 Java 及 C/C++ 語言開發，並可使用 JNI(Java native interface)做彼此溝通的介面。可將 C/C++ 程式編譯成 native code library 供 Java 程

式載入使用。〔9〕〔15〕

使用 Java 語言撰寫程式，可以使用便利的開發工具 Eclipse 再搭配模擬器與除錯支援，並可以使用 Application Framework 提供的 API 以增加開發程式的速度。

反之，如使用 C/C++ 撰寫程式，由於不需要透過 Dalvik Virtual Machine 執行，可加速軟體執行效率，但程式會變得較為複雜與不安全，程式的除錯也變得較為困難，故須謹慎選擇開發的語言。

開發工具與語言比較：

	Windows 播放器	Linux 播放器	Android 播放器	
開發語言	C++	C/C++	C/C++	Java
開發工具	BCB6	GTK+	NDK	Eclipse + Android SDK

表格 8：開發工具與語言比較

Linux 上的播放器是使用 C/C++ 開發的，希望能盡量沿用既有的程式。但 Android 平台不支援 Linux 上管理圖型介面的工具程式庫 GTK (GIMP ToolKit) 〔10〕〔11〕，且不支援 Firefox Plugin，故使用 Android 提供之 API，處理 UI、多媒體處理與呈現及與部落格之整合。

因此，本論文的移植策略為『盡量 reuse、修改原本 C/C++ 程式碼，C/C++ 難以實作部分使用 Java 改寫並配合 Android 提供之 API』。

3.5.3 程式移植方式

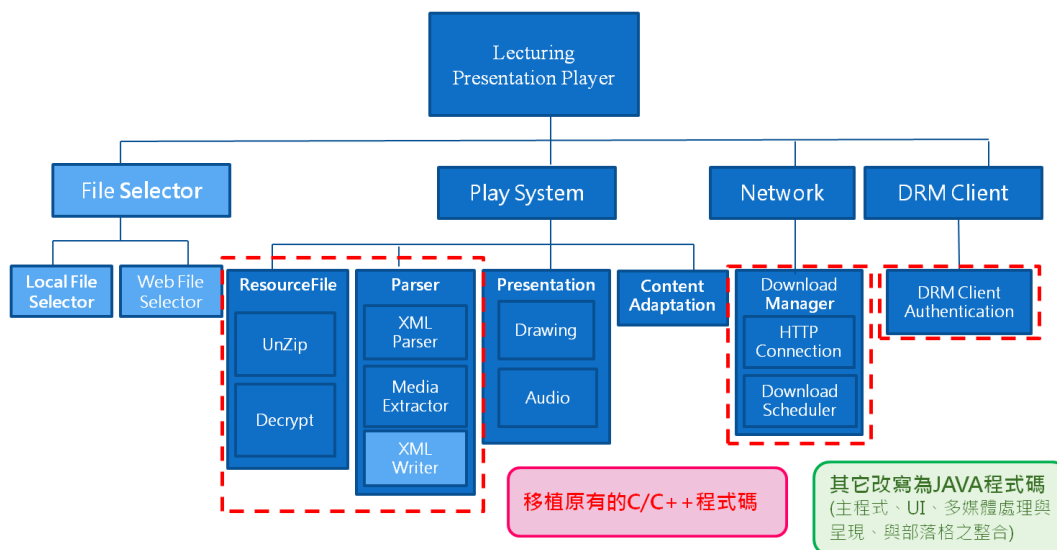


圖 8：程式移植方式

(一) C/C++程式移植部分

針對 Android 平台上不支援的資料型態及函式庫做修改。Windows 平台上講解手播放器使用了 BCB 及 VC 定義的資料型態，但在 Linux 平台上並不支援。故採用標準 C++的資料型態替換 BCB 和 VC 內部定義的資料型態 [22]。然而，Android 的 Bionic C 只提供少數的標準 C++函式庫，故需額外尋找替代的函式庫及資料型態，替代方式如下：

Component	Platform	Windows	Linux	Android
Extra include Standard C++ Library		No need	No need	<u>STLport</u>
Unicode Processing		TtntString	Glib	<u>utils</u>
XML DOM Model		XMLDOC	LIBXML2	LIBXML2
ZIP decompress Processing		zlib	zlib	zlib
Communication between C++ and Java programs		No need	No need	<u>JNI</u>

表格 9：C/C++程式之元件替換方式

為了要使用標準 C++函式庫，故使用了 STLport 這套函式庫，但在 Android 平台的 STLport 並不支援 wstring、wchar 及其相關函式，故額外使用 Android 底層的函式庫 libutils，並使用 String16、char16 替代。最後，加入 JNI 介面與 Java 程式溝通。而處理解析 XML 檔案與解壓縮的部分，還是使用 LIBXML2、zlib 這兩套函式庫。

(二)Java 程式移植部分：

由於 Java 程式負責處理呈現多媒體檔案，故先探討 Android API 對多媒體檔案的支援，最後再分析元件替換的方式。

■ Android API 對多媒體檔案的支援：

講解檔使用到的檔案類型	檔案格式	API是否支援?	備註
圖片	PNG、JPG、BMP、GIF	支援	
聲音	WMA	支援	
影片	WMV	部分支援	支援WMV的Versions 7, 8 and 9. Simple profile only (解析度 160x120)
Flash	SWF	尚未支援	手機可安裝Flash插件觀看Flash檔但Android尚未提供播放Flash的API

表格 10：Android API 對多媒體檔案的支援

講解檔裡面使用到的影片檔為桌面錄影檔案，其解析度皆大於 160x120，故

無法在 Android 平台上播放。Flash 檔案部分，雖然 HTC Hero 可以觀看，但目前無法得知如何指定播放本機端的 SWF。故目前只處理圖片檔及聲音檔。

■ 元件替換方式：全部使用 Android 提供之 API 替換，如表格 11 所示。

Platform Component	Windows	Linux	Android
General	C++	C++	Java
Drawing Related	GDI+	GDK	android.graphics
Image Related	GDI+	GDK-pixbuf	android.graphics
Audio Related	Customized Media Player	Mplayer	android.media
UI Related	GDI+	GDK	android.view android.widget
Web Related	IE plugin	FireFox plugin	android.webkit

表格 11：Java 程式之元件替換方式

總結：

本研究彙整了多平台之多媒體講解呈現播放器架構之元件替換方式，整理如圖 9。其中描述了移植 Windows 平台之多媒體講解呈現播放器於 Android 平台、Windows CE 平台、Linux 平台所需的元件替換，詳細之移植方式請參考 [20][21][23][27]。

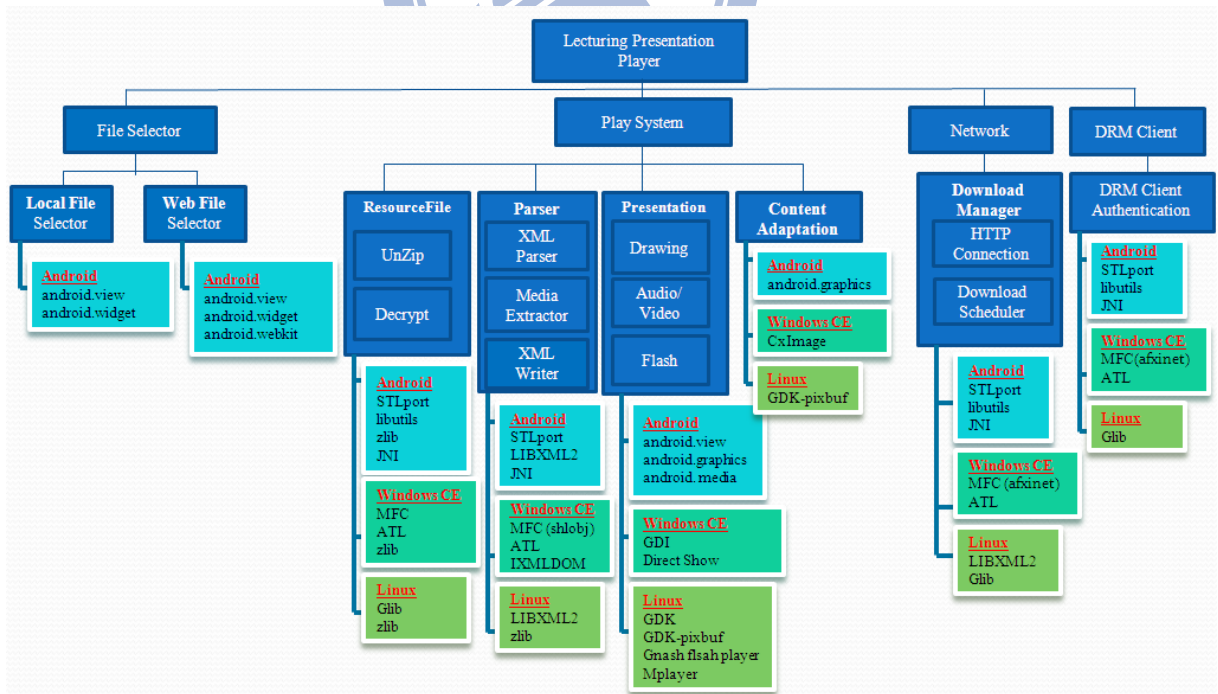


圖 9：多媒體講解呈現播放器架構之元件替換

3.6 多媒體講解呈現檔

在 Windows 平台上，講解手錄製器發佈出來的檔案為多媒體講解呈現教材，以下簡稱為講解檔。由於本研究需載入播放講解檔，故先探討講解檔的架構，再了解內部的儲存資訊及檔案封裝的方式。

■ 講解檔案架構：

一份講解檔的架構類似於 Powerpoint 檔案分頁的方式，一頁投影片教材發佈成一個多媒體講解呈現檔(*.bst)。如果一份講解檔包含了多頁的投影片教材，則會發佈成多個 bst 檔，並使用一份 publish.xml 檔案紀錄整份講解檔的資訊。另外，還有一些其它的檔案例如: teacher.png、index.htm、lecturing.js 等。teacher.png 是講師圖片檔，index.htm、lecturing.js 則是講解檔的網頁畫面資訊。以下介紹較為重要的 publish.xml 與 bst 檔之內容。



圖 10：多媒體講解呈現檔架構

◆ publish.xml：

紀錄整份講解檔的相關資訊，一份講解檔只有一份 publish.xml。包含了以下五種資訊：

1. 發佈資訊：整份講解檔的主題跟講師資訊。
2. 面板資訊：播放器外觀套用的面板資訊。
3. 播放資訊：播放此份講解檔的方式。
4. 課程資訊：講解檔內每個 bst 檔的個別資訊。
5. DRM 資訊：DRM 認證的相關資訊，DRM Server IP、Course ID。

藉由此檔案才可得知此份講解檔之 bst 檔播放清單，並得知要下載的 bst 檔及取得 DRM 驗證所需資訊。故在播放前需檢查此份檔案是否存在。

◆ bst 檔：



圖 11：bst 檔編碼流程

教材錄製工具會將場景圖資訊及腳本資訊分別錄製，並經過壓縮及加密之後，寫入標頭檔並封裝成 bst 檔。故解密前，必須要先取得解密金鑰，才可對 bst 檔進行解密。

本小節會先藉由分析加密金鑰產生方式，進而了解解密金鑰之取得方式。最後，分別介紹場景圖的資訊及腳本資訊的內容。

加密金鑰的產生存放方式

分為以下兩種：

1. 由 Triple-DES 演算法隨機產生，儲存於 DRM Server

在錄製端發佈檔案時，如果講解者有選擇加入 DRM 驗證的話，則會使用 Triple-DES 演算法隨機產生一個 Key，並利用此 Key 將 bst 檔做加密，再透過 DRM 驗證的方式傳送到 DRM Server 存放，並將 DRM 驗證的相關資訊存入 publish.xml 當中。

2. 使用 Default Key，不須額外存放。

在錄製端發佈檔案時，如果講解者沒有選擇加入 DRM 驗證的話，則使用 Default Key 將 bst 檔做加密。

解密金鑰的取得方式

由於 Triple-DES 演算法是採取對稱式加密演算法 (Symmetric Encryption Algorithm)，也就是加解密是使用同一把金鑰。故取得解密金鑰方式亦分為兩種：

1. 連線到 DRM Server 取得

如果 publish.xml 有 DRM 驗證資訊，則讓使用者輸入帳號密碼後，連線到 DRM Server 取得。

2. 使用 Default Key

如果 publish.xml 沒有 DRM 驗證資訊，代表是使用 Default Key 加密，所以直接使用 Default Key 解密即可。

場景圖(SceneGraph)資訊

紀錄了此頁投影片的資訊。講解手教材錄製工具會將 PPT 檔案的內容解析成一個個多媒體檔案，並將多媒體檔案的資訊(例如:大小、位置、名稱)存入 PPTXML 檔案中。

投影片的呈現就像一場戲一樣，投影片上的多媒體檔案視為一個個演員，如圖片演員、特效演員、影音演員等。而 PPTXML 則可視為演員演出的劇本，演員會依照劇本紀錄的資訊做演出。在同一個時間點呈現的畫面稱做一個場景；而場景隨著時間的更迭可能會有所改變(例如:特效演員的出現、聲音影片的播放)，故一份 bst 檔可能會有多個場景，而多個場景聚集成場景圖。

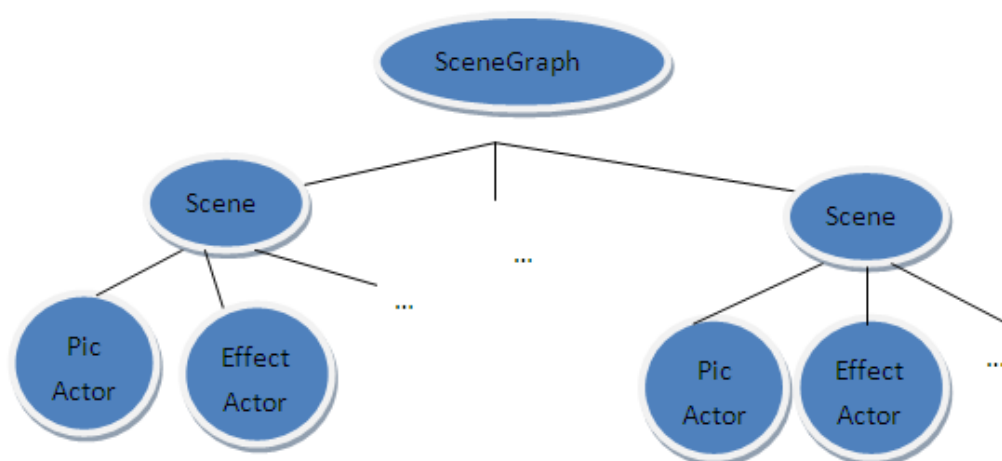


圖 12：場景圖資訊

腳本資訊

包含了講解者的塗鴉註釋動作、聲音及桌面錄影的多媒體檔案。講解手教材錄製工具會每 0.01 秒擷取一次，並存入 Script 檔案中。而聲音及影片多媒體檔案會每十秒鐘切割成一個檔案做儲存。

Script 檔案紀錄的內容：以時間為單位，條列式的紀錄某個時間點所發生的動作及動作參數。時間單位為 0.01 秒。

時間	動作	參數
----	----	----

```

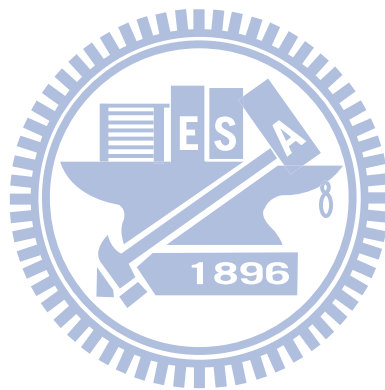
001853:Scene.MouseMove(95,176);
001853:Scene.MouseMove(95,176);
001861:Scene.Draw(95,176,BeginPen,ColorPen,255,3);
001861:Scene.MouseMove(95,175);
001865:Scene.Draw(95,175,OnPen,ColorPen,255,3);
001865:Scene.MouseMove(95,175);
001867:Scene.Draw(95,175,OnPen,ColorPen,255,3);
001867:Scene.MouseMove(95,172);
....
010989:Scene.Draw(755,523,BeginLine,Straight,16711680,3);
011195:System.End();
  
```

圖 13：腳本系統之 script 檔內容

紀錄的動作分為下列四種：

1. 塗鴉相關：畫筆、螢光筆、橡皮擦、打字、畫線、畫幾何圖形
2. 滑鼠相關：滑鼠移動、游標隱藏/顯示
3. 特殊控制：開啟/關閉電子白板、開始/結束桌面錄影
4. 系統控制：開始/結束錄影

在播放時，依照播放的時間點，讀取此份檔案解析完的資訊，將講解者的塗鴉註釋動作呈現出來。



四、系統實作

本章先介紹整個系統的播放流程圖，了解整個播放的流程後，再以功能做分類，分別做詳細的介紹。主要分成以下三類功能：

1. 基本播放器功能：處理、播放 bst 檔及書籤功能。
2. DRM 驗證：輸入帳號密碼，連線至 DRM Server 做驗證。
3. 與部落格結合：播放器與部落格進行溝通、從部落格下載檔案。

4.1 多媒體講解呈現播放器於手機端的播放流程

播放端的播放方式分為兩種，播放本機儲存端的講解檔、播放部落格上的講解檔。本小節將先針對兩種播放流程做介紹。由於一份講解檔可能包含多份的 bst 檔，本小節會再更深入的介紹「bst 檔播放排程」以了解 bst 檔的播放順序，並介紹「bst 檔下載排程」以了解 bst 檔下載排程設計的概念。

4.1.1 播放本機儲存端講解檔的流程圖

播放本機儲存端講解檔的流程圖如下：

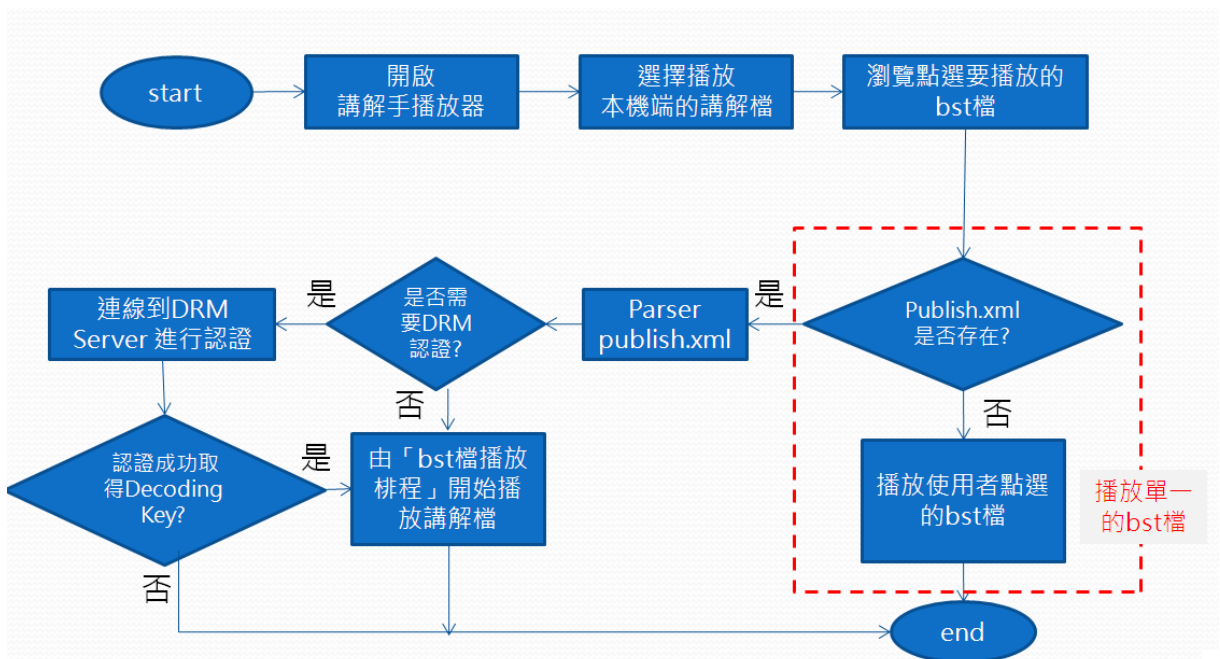


圖 14：本機儲存端講解檔之播放流程

步驟如下：

1. 使用者先開啟講解手播放器並選擇播放本機端的講解檔
2. 使用者藉由檔案選擇清單選擇要播放的 bst 檔。
3. 檢查此份 bst 檔的 publish.xml 檔案是否存在，如果 publish.xml 檔案不存在的話，因為無法得知此份講解檔所包含的 bst 檔，故直接播放使用者選擇的 bst 檔，播放完此 bst 檔便結束播放流程。

4. 如果 publish.xml 存在的話，就會解析 publish.xml。由於 publish.xml 檔會紀錄 DRM 的驗證資訊及 bst 檔案的播放清單等等，故須先解析此份檔案才可做後續的檢查、播放動作。
5. 由解析出的資訊檢查此份講解檔是否需要 DRM 的驗證，如果不需要 DRM 驗證的話，則直接進入「bst 檔播放排程」開始播放講解檔。
6. 如果此份講解檔需要 DRM 驗證的話，則跳出視窗讓使用者輸入帳號密碼，再連線到 DRM server 做驗證。如果驗證通過且成功取得 Decoding Key，便可進入「bst 檔播放排程」開始播放講解檔。如使用者驗證失敗的話，則無法進行播放。

4.1.2 播放部落格上的講解檔之流程圖

播放部落格上的講解檔之流程圖如下：

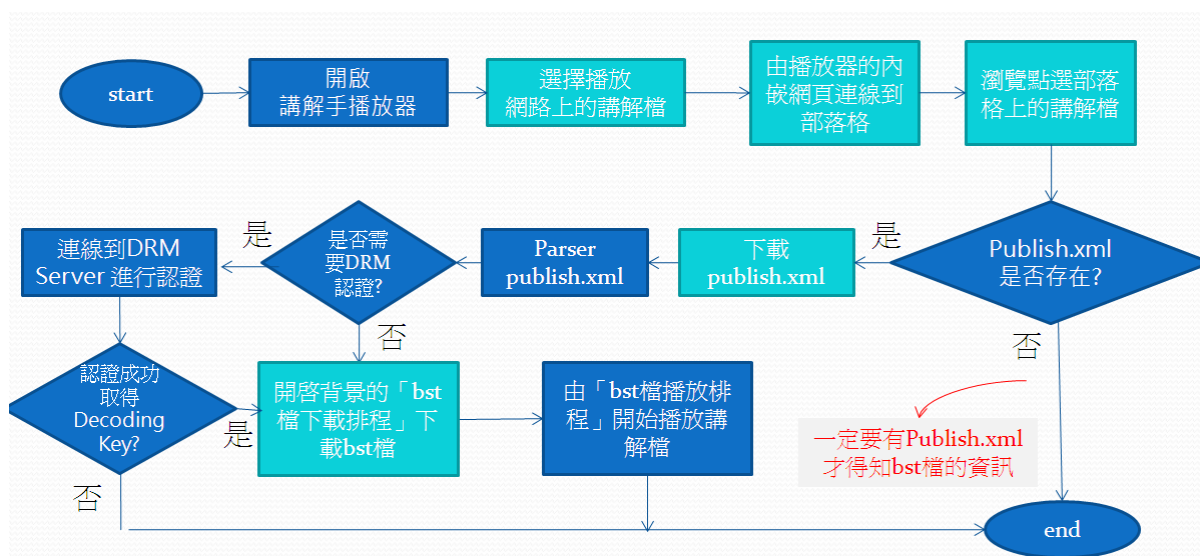


圖 15：部落格上講解檔之播放流程

播放網路端的流程與播放本機端的流程類似，只有少部分做了改變。淺綠色的部分為相異的部分。步驟如下：

1. 使用者開啟講解手播放器，選擇播放網路上的檔案。
2. 播放器開啟內嵌網頁供使用者選擇瀏覽數位學習部落格上面的講解檔。
3. 當使用者點選網頁上的播放按鈕時，網頁會將下載 publish.xml 的 URL 網址傳送給播放器，讓播放器下載 publish.xml。
4. 如果欲下載的 publish.xml 不存在，則因為不知道此份講解檔包含了哪些 bst 檔而無法下載檔案，故不會進行播放的動作，便直接結束程式。
5. 下載完 publish.xml，同樣會去做解析 publish.xml，並去檢查是否需要 DRM 驗證。如不需要 DRM 驗證的話，則會開啟「bst 檔下載排程」開始下載此份講解檔所包含的 bst 檔，才進入「bst 檔播放排程」開始播放講解檔。

6. 如果此份講解檔需要 DRM 驗證的話，亦會提供使用者輸入帳號密碼的視窗。如果驗證通過且成功取得 Decoding Key，才會會開啟「bst 檔下載排程」，之後進入「bst 檔播放排程」開始播放講解檔。

4.1.3 bst 檔播放排程

bst 檔播放排程如下：

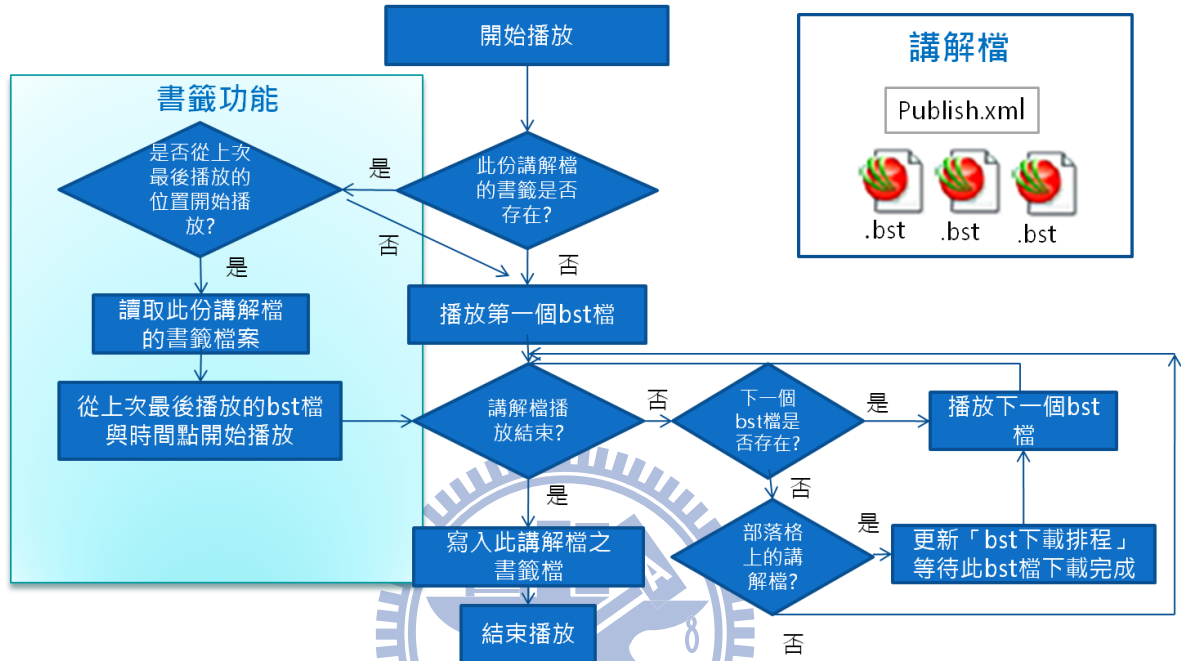


圖 16：bst 檔播放排程

檢查書籤檔是否存在。如果存在，則詢問使用者是否要從最後播放的位置開始播放。如果要的話則從書籤內紀錄的位置開始播放；反之，則會從播放清單內的第一個檔案開始進行播放。

而在播放每個 bst 檔案之前，會先檢查此份 bst 檔是否存在，如果存在的話就會開始播放；如果不存在且檔案在部落格上尚未下載下來的話，則會更新「bst 下載排程」等待此份 bst 檔下載完成才開始播放。如果是本機端的檔案且不存在的話，則會跳過此份 bst 檔，檢查此份講解檔是否播放結束，如果還沒結束，就直接播放下一個 bst 檔。

4.1.4 bst 檔下載排程

此下載排程的設計是為了減少學習者等待講解檔下載的時間而設計的。此排程包含了以下四種特色：

- 使用者正要觀看的bst檔下載完成便可開始觀看

由於一份講解檔可能包含了多份的bst檔，如果等到所有的bst檔都下載完成才開始觀看，會耗費使用者等待的時間。而每一份bst檔都是可以獨立播放的，故設計只要下載完正要觀看的bst檔，便可開始播放此bst檔。預設為下載完第一個bst檔便可開始播放。

■ 背景預先下載其它的bst檔

bst檔正在播放的過程當中，如果還有其它bst檔尚未下載完成的話，則可在背景開啟一個thread下載其它的檔案，以減少觀看前的等待下載時間。

■ 隨選下載使用者選擇觀看的bst檔

在檔案下載的過程當中，使用者使用bst檔選擇清單選擇觀看其它bst檔時，如果選擇的bst檔尚未下載完成的話，則會中斷目前正在下載工作，立刻去下載使用者選擇觀看的檔案，並改變之後檔案下載的順序。

■ 支援bst檔案的續傳功能

由於下載工作會因為隨選下載而被終止，故增加了續傳的功能，以避免bst檔下載到一半被迫中斷而需全部重新下載的狀況。實作方式將於之後4.4.1節做詳細的介紹。

4.2 基本的播放功能實作

本章節將先介紹 bst 檔的處理流程，再分別介紹各個模組的實作方式。下圖為講解手播放器針對單一的 bst 檔的播放流程：

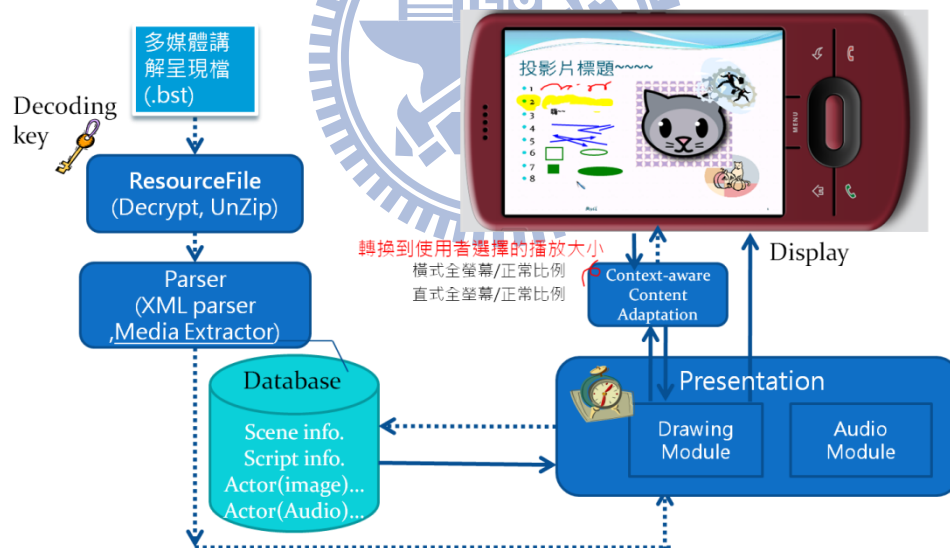


圖 17：bst 檔播放流程

1. 使用 ResourceFile 模組將 bst 檔進行解密、解壓縮，並將解碼出來的檔案儲存於記憶卡或記憶體中。
2. 將解碼出來的 XML 檔案作解析，並將解析出來的場景圖資訊及腳本資訊的資料存入 Media Extractor 模組裡類似 Database 的結構中。
3. 啟動 Presentation 模組的運作，開始計算播放的時間。
4. Presentation 模組內的 Audio 模組會開始播放聲音檔。

5. Presentation 模組內的 Drawing 模組會依照播放的時間從 Database 中取得更新講解畫面的所需資訊，將資訊經過內容協調後，繪製於畫布上。
6. 將講解畫面經過內容再經過一次內容協調後，轉換到使用者選擇呈現的比例及方向，最後呈現於螢幕上。

4.2.1 檔案解析

由檔案解析模組負責實作。將 bst 檔解碼，並將解碼後的檔案寫入記憶卡或記憶體裡。

bst 檔的解碼方式

可參考 3.6 多媒體講解呈現檔之 bst 檔介紹，以了解 bst 檔編碼方式及解密金鑰取得方式。

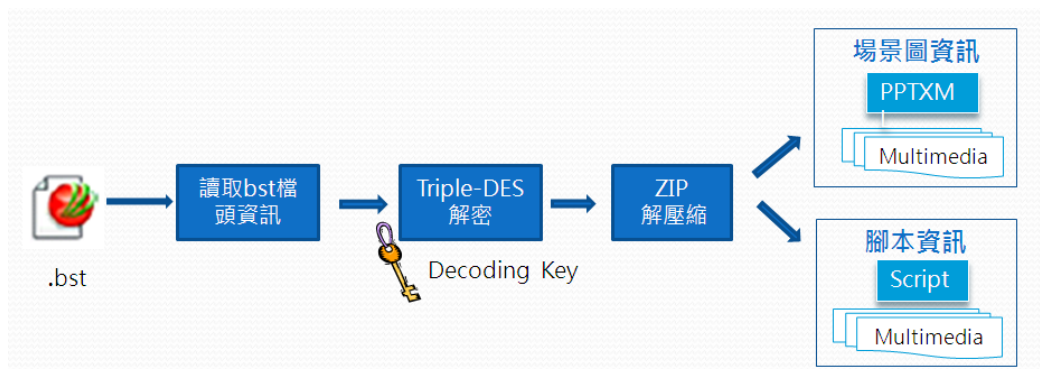


圖 18：bst 檔解碼流程

解碼時，採用與編碼反向的方式操作，步驟如下：

1. 讀取標頭檔：用來得知檔案的位置及大小。
2. 解密：找到到檔案並使用解密金鑰對檔案做 Triple-DES 解密動作。
3. 解壓縮：使用 zlib 將解完密的檔案做 ZIP 解壓縮的動作。

將檔案寫入記憶卡或記憶體

講解檔在解碼後，會將內部儲存的檔案全部擷取出來。為了保護講解檔案，將部分的檔案存入記憶體當中，以防止被串改重製講解檔。由於手持裝置之記憶體較小，不是每種檔案都適合存放於記憶體，故將聲音、影片檔較耗費空間的檔案儲存於記憶卡。其它 XML 檔案及圖片檔則存於記憶體。

此部分程式移植的方式：

由於 bst 檔案的編碼及 XML 檔案寫入都是在 Windows 平台上執行的，而 Windows 平台 unicode 的編碼方式為 UTF-16，在 Android 平台上的 unicode 的編碼方式為 UTF-8，故需先做 unicode 轉換。本研究利用 Android 底層函式庫 libutils 定義的資料型態 String16 處理 UTF-16 之編碼格式，並使用其函式做 unicode 轉換。

4.2.2 資料解析

由檔案解析模組負責實作，包含三個子模組：

1. XML Parser 模組：解析 XML 檔案
2. XML Writer 模組：寫入 XML 檔案，用於寫入書籤檔。
3. Media Extractor 模組：將解析完的資訊存入類似資料庫的結構中。

運作的流程如下：

- 一、 解析 XML 檔案並將資訊存入 Media Extractor 管理。

處理的檔案:紀錄課程發佈資訊 publish.xml、紀錄場景圖資訊的 PTTXML、紀錄腳本系統資訊的 script 檔。

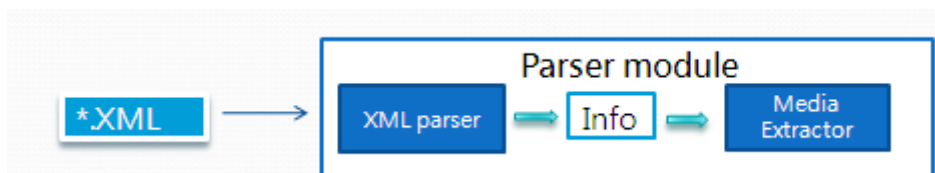


圖 19：解析 XML 運作流程--寫入資料結構

- 二、 解析 XML 檔案，並將 Info 寫入 XML 檔案。

處理的檔案:書籤檔。

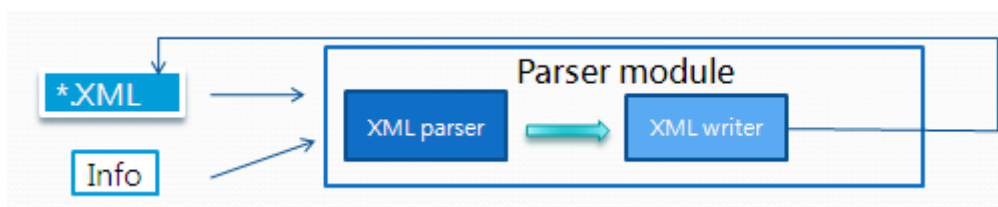


圖 20：解析 XML 運作流程--寫入 XML 檔

XML Parser、XML Writer 模組：

在 Linux 平台上使用 Libxml2 取代 Windows 平台上的 XMLDOC，並模擬 XMLDOC 的函式接口，本研究的作法如同 Linux 平台。詳細的取代方法可參考 [22]。

Media Extractor 模組：

將解析完的資料存入類似資料庫的結構內。包含:課程發佈資訊 publish.xml、場景圖資訊 PPTXML、腳本系統 script 檔。

1. 課程發佈資訊 publish.xml

此份檔案最重要的資訊為課程資訊，包含了每個 bst 檔的主題、時間、檔案名稱。而其它 publish.xml 內的資訊則是使用 map 的方式紀錄起來。可使用 TrackFileNames 取得包含的 bst 檔名稱。



(a)

(b)

圖 21：(a) 課程發佈資訊資料結構 (b) 對應的 bst 頁面列表

2. 場景圖內容

包含場景資訊及演員的個別資訊，使用 CSceneGraph 的物件儲存。一個場景圖裡面可能會有許多個演員(CActor)，故使用一個 vector 做存放，而每個演員都有其特定的屬性，並由 m_ImageFileName 做為 Key 取出存在記憶體裡面的演員圖檔，如圖 21 所示。



圖 22：場景圖資訊關連圖

腳本資訊以時間為單位條列式的紀錄講解者的動作及其參數，並紀錄於 script 檔案。Script 檔案經過解析後存入 CScriptSystem 的物件中，再以一個 vector 存放講解者的動作。由於 script 檔案每個時間點紀錄了時間、動作、動作參數三種資訊，故 CScriptAction 包含了 Time、Action、Parameters 三個變數。其關聯如下圖 22 所示：

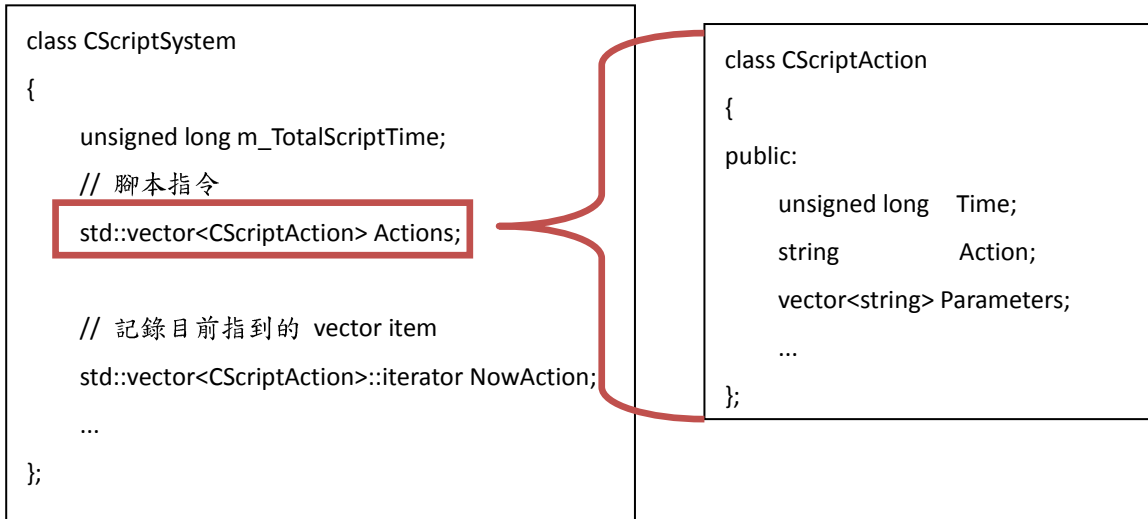


圖 23：腳本系統資訊關連圖

4.2.3 情境感知內容協調

由內容協調模組負責實作。在 Windows 平台上錄製的講解檔的畫面解析度是 720*540，但 HTC Hero 的螢幕解析度僅為 480*320，故需作內容協調。而為了讓使用者可便利地切換畫面呈現的方向(垂直/水平)及比例(全螢幕/正常比例)，故採用情境感知內容協調方式。也就是說，畫面呈現的大小會隨著使用者的切換而做改變，假設為 W*H。

播放資訊協調

場景圖資訊和腳本資訊紀錄了許多物件資訊，其中紀錄物件呈現的位置是使用二維座標方式表示。

Windows 平台原始講解畫布大小為 720*540，其座標系統為

$$\{ X, Y \mid 0 < X < 720, 0 < Y < 540 \}$$

Android 平台將畫面解析度為 W*H，其座標系統為

$$\{ X', Y' \mid 0 < X' < W, 0 < Y' < H \}$$

轉換公式： $X' = X * W / 720$ ， $Y' = Y * H / 540$

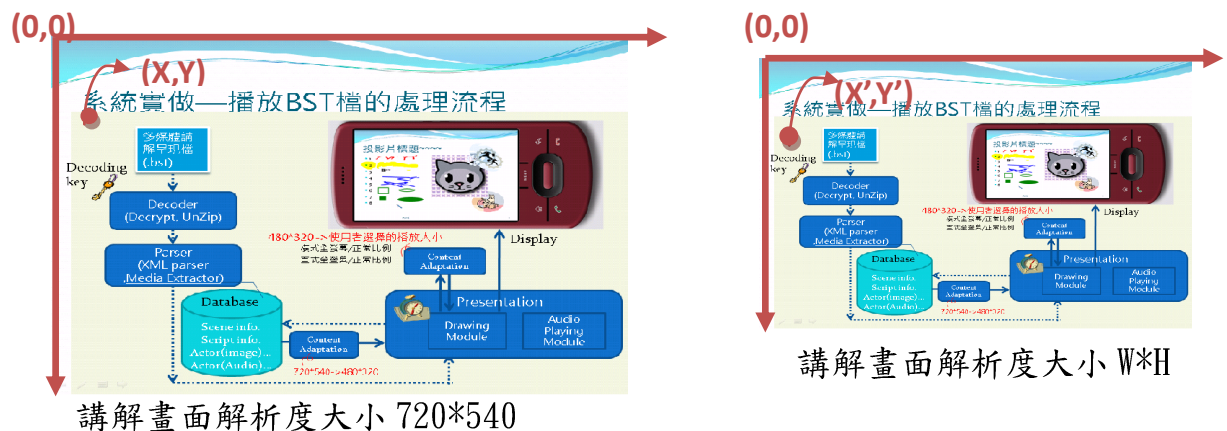


圖 24：播放資訊協調示意圖

多媒體圖片內容協調

採取解析度轉換的方式，並使用 android.graphics 內提供的圖片處理相關函式庫將圖片的解析度做轉換。假設原始圖片的寬度是 OriWidth，高度是 OriHeight，轉換出來的新圖的寬度是 NewWidth，高度是 NewHeight，轉換公式如下：

$$\text{NewWidth} = \text{OriWidth} * W / 720$$

$$\text{NewHeight} = \text{OriHeight} * H / 540$$



講解畫面解析度大小 720*540

圖 25：多媒體圖片內容協調示意圖

4.2.4 內容呈現

分為畫面呈現與聲音呈現二部分。Drawing Module 負責畫面呈現，Audio Module 負責聲音播放。

Drawing Module

採用 Java 語言重新開發，使用 android.graphics 取代 Linux 中使用的 GDK 這套繪圖函式庫。繪圖模組負責的是畫布上的繪圖呈現，包含了場景演員的呈現和腳本動作的呈現。

本研究將繪圖系統分為三個圖層討論。場景演員的呈現自成一層，因為講解者加註不會影響投影片的內容，可避免清除講師加註的內容時更改到原始投影片內容。講解者加註的內容分成兩層，分為講解者塗鴉與講解滑鼠呈現，因為滑鼠位置的更新不會影響塗鴉內容，分為兩層可方便更新滑鼠位置。如圖 26 所示。

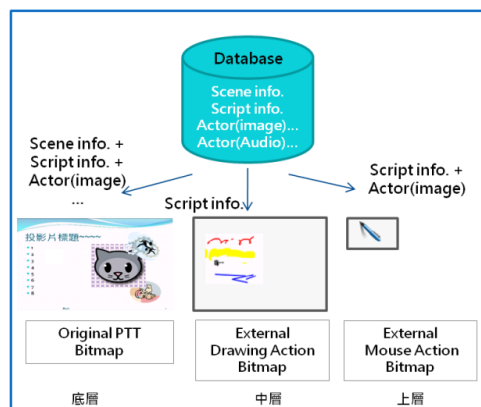


圖 26：畫布分層更新示意圖

中層和上層會依照播放的時間進行更新，而底層目前只實作靜態演員的部分，故不會隨著時間做更新。三個圖層都會去 Media Extractor 這個類似資料庫的模組中取得更新所需的資訊，三個圖層獨立更新再結合起來呈現，以下介紹每一層的做法：

1.底層：

投影片演員的呈現。演員有分為動態演員與靜態演員兩部種類型，由於動態演員牽扯到許多特效的呈現，故本研究目前只呈現靜態演員的部分。而場景圖的演員資訊儲存於 Media Extractor 模組中，可參考 4.2.2 小節的圖 22 了解場景圖資訊的關連，將演員繪製於 Original PTT Bitmap 上。

2.中層：

講師加註的內容，只有塗鴉動作的呈現，不包含滑鼠的呈現。可參考圖 23，了解取得腳本資訊的方式，並依照播放時間將塗鴉繪製於畫布上。使用到的函式如下表 12 所示：

塗鴉	Function name
畫筆	Canvas.drawLine
螢光筆	Canvas.drawLine
打字	Canvas.drawText
畫線 - 直線、箭頭	Canvas.drawLine Canvas.drawPath
畫幾何圖形- 空心/實心橢圓、 空心/實心四邊形	Canvas.drawOval Canvas.drawRect

表格 12：塗鴉動作實作方式列表

3.上層：

滑鼠的貼圖呈現。類似於中層的做法，在播放時需依照播放的時間點讀取 script info.取得滑鼠所在的座標位置、屬性等資料，將滑鼠繪製於畫布上。

圖層結合的方式是將三個圖層繪製於一張 Bitmap 上後，再經過一次內容協調，呈現到螢幕上。

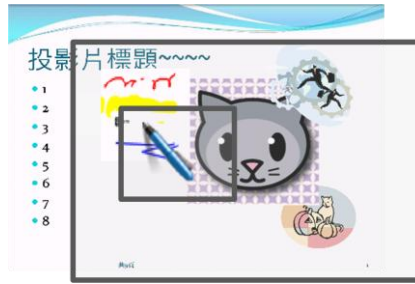


圖 27：圖層結合方式

4.3 DRM 驗證

由於 DRM 資訊會紀錄於 publish.xml 中，所以在播放端播放 bst 檔前，會先檢查 publish.xml 檔是否有紀錄 DRM 資訊。如果有的話，就會透過 DRM Client 模組做驗證。

DRM Client 模組負責提供視窗讓使用者輸入帳號密碼，再連線到 DRM Server 進行驗證。流程如下：

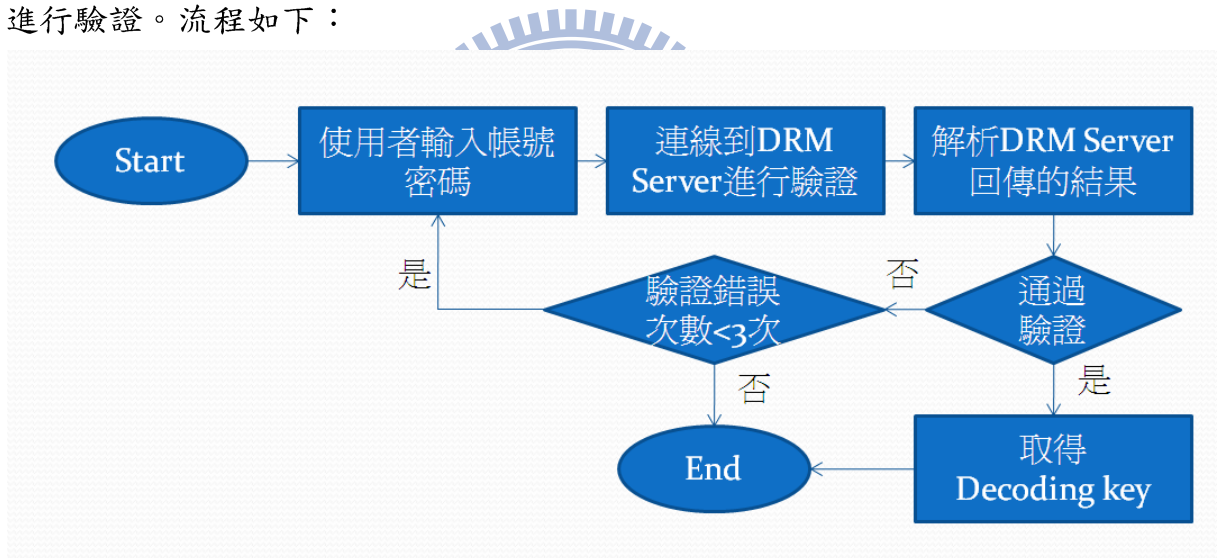


圖 28：DRM Client 模組運作流程

1. 提供視窗讓使用者輸入帳號密碼。
2. 利用使用者輸入的帳號密碼、publish.xml 裡的 DRM Server IP 與 Course ID 連線到 DRM Server 做驗證。
3. DRM Server 會回傳驗證結果於 XML 檔案裡，故解析此檔案檢查驗證是否通過。
4. 如果驗證通過，則可在 DRM server 回傳的 XML 檔案裡取得 Decoding Key。
5. 如果沒通過，且錯誤的次數小於三次的話，則重新輸入帳號密碼，再次驗證。反之，結束程式。

4.4 網路結合相關

4.4.1 播放器與部落格結合

由 Web File Selector 模組負責提播放器與部落格的結合。此模組會擷取使用者點選的講解檔資訊，並下載講解檔中的 publish.xml 檔案。

在 Android 平台上，可在應用程式中嵌入 WebView 以呈現網頁內容，並且透過 Webview 與網路上的 java script 檔案作 function call 的溝通，使應用程式與網頁之間的整合應用變得更為便利。本研究藉由此特點，設計播放器與部落格結合的方式，如下圖所示：

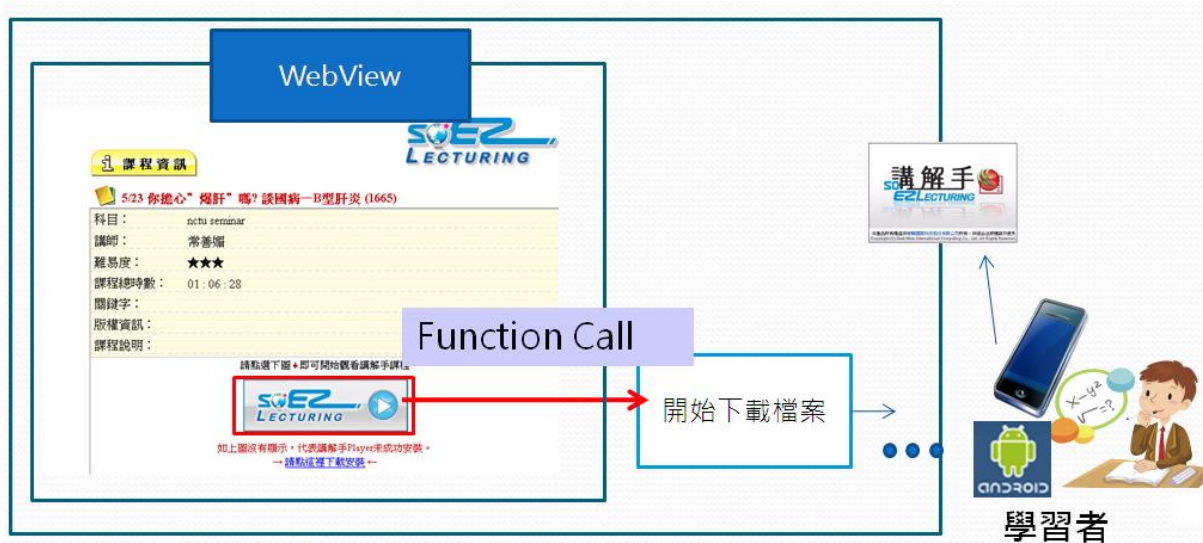


圖 29：播放器與部落格結合示意圖

當使用者開啟講解手播放器並選擇播放網路上的檔案時，會跳出內嵌的 WebView 供使用者選擇瀏覽部落格上面的資訊。使用者可以瀏覽網路上的講解檔資訊並點選播放按鈕開始播放。當按下播放按鈕時，便會以 function call 的方式傳送此放講解檔的 publish.xml 的 URL 給播放器，讓播放器可以開始下載檔案。分為兩個部分實作，一為修改講解檔內的網頁，一為播放器內嵌 WebView 的實作。

修改網頁

講解檔的網路頁面是由講解手錄製器在發布講解檔時所產生的，其中網頁中的播放按鈕及其功能是由 Lecturing.js 所實作的。修改方式：繪製播放按鈕的圖片，並設定 click 此圖片時，呼叫播放器的 function，並傳送 publish.xml 的 URL 參數。

播放器內嵌 WebView

將 WebView 內嵌於程式之中，並加入下載檔案的介面。

4.4.2 下載部落格上的講解檔

檔案下載模組會實作「bst 檔下載排程」，負責連線到 Content Server 下載講解檔。實作的方式使用 [23] 設計之檔案下載排程器，而下載檔案的方式則是使用 HTTP 協定中的 GET 方式。

下載排程器的運作流程如圖 29 所示：

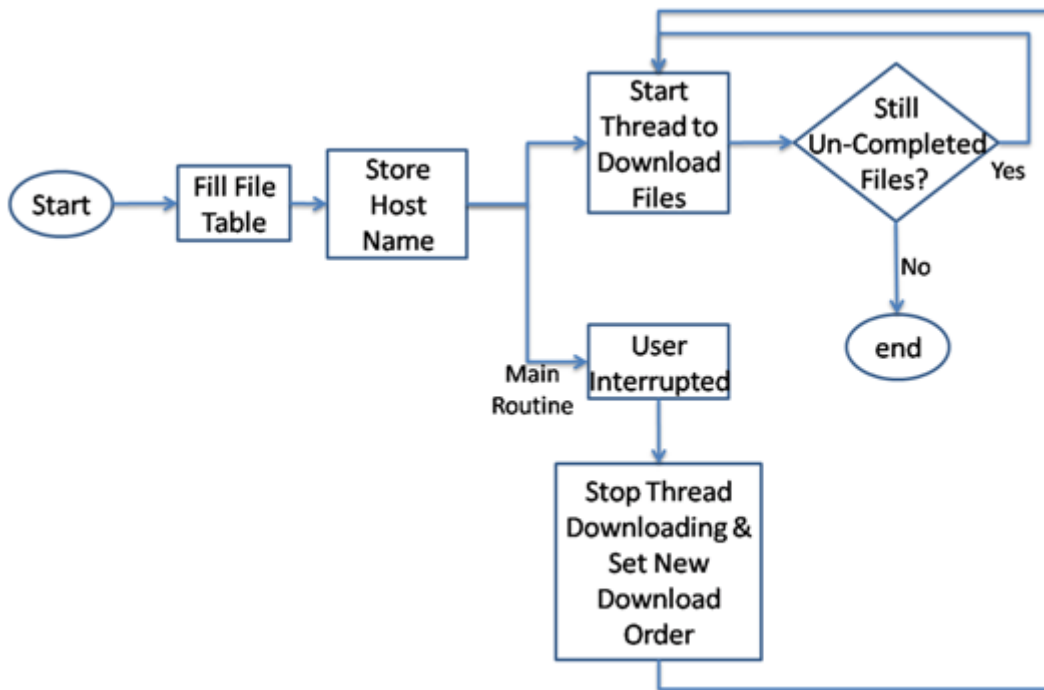


圖 30：下載排程器運作流程

將 publish.xml 解析出的資料建立三個 table：

(1) 檔案下載列表：

儲存 bst 檔案的下載相關資訊。包含了每個 bst 檔的下載位置、檔案大小、已下載的檔案大小、下載狀況。

(2) Domain Name 與 IP 對照表：

下載檔案前，必須先取得真實的 IP 才能進行下載。事先儲存，可避免一直重複去 Domain Name server 取得真實的 IP 位置。

(3) 檔案下載順序表：

儲存檔案的下載順序，預設由第一個檔案開始下載。

建立好 table 後，便會開啟一個 thread，依照檔案下載順序表開始下載檔案，並更新檔案下載列表裡的已下載的檔案大小、下載狀況。

當使用者透過 bst 檔案選擇清單選擇播放其它 bst 檔時，Main routine 就會中斷下載檔案的 thread，並更改檔案下載順序表。舉例來說，目前檔案下載的順序為 001-bst.bst -> 002-bst.bst -> 003-bst.bst -> 004-bst.bst，而 thread 正在下載 001-bst.bst。當使用者選擇播放 003-bst.bst 檔案時，目前的 thread 會被中止掉，並預測下一個播放的檔案為 004-bst.bst，故更改檔案下載順序表，如圖 31 所示。

而下載到一半的檔案要再繼續下載時，便可由檔案下載列表資訊得知有多少尚未下載完成，進而做續傳的工作。



圖 31：檔案下載範例示意圖

4.5 播放器效能改進

由於手持裝置之 CPU 較弱、記憶體不足，容易造成畫面、聲音呈現之延遲，而無法正確呈現出原本講解的內容。故本研究針對 Presentation module 做效能改進。改進的模組包含了 Drawing Module 與 Audio Module：

- (1) Drawing Module：畫布更新時間計算方式、螢光筆混色方法。
- (2) Audio Module：聲音檔案預先載入。

4.5.1 畫布更新時間計算方式

如沿用 Linux 平台上講解手播放器的更新畫布方式於在 Android 平台上，會產生畫面嚴重延遲的狀態，故本研究設計了改進的方法。本節先說明 Linux 平台上講解手播放器的更新畫布的方式，並推測延誤的原因，最後提出改進的方法。
原始的畫布更新方式

設定計時器每 0.1 秒啟動一次，將播放的時間(PlayingTime)增加 0.1 秒並更新畫布。如下圖所示：

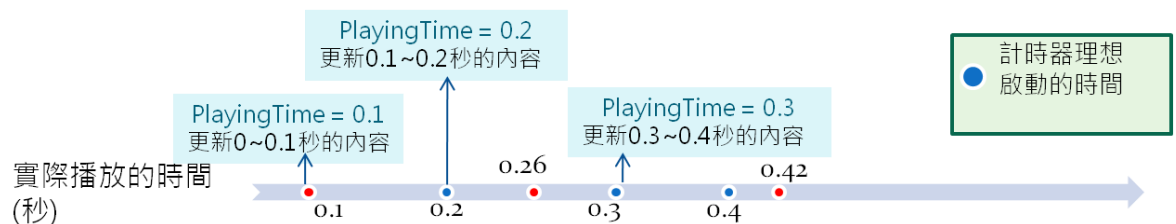
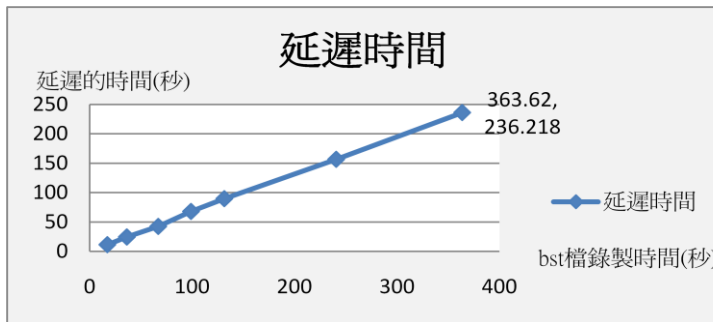


圖 32：原始的畫布更新方式--預期結果

然而，在 Android 手持裝置上延遲的時間如表 13 所示。延遲的狀況會隨的時間增長而變得更嚴重。



表格 13：原始畫布更新方式之延遲時間

推測延遲原因

手持裝置上 CPU 效能不足，計時器的啟動本身就會有延遲。而畫布繪製的時間較長，平均約為 0.14 秒，而造成計時器無法每 0.1 秒啟動一次。另外，可能有其它因素消耗 CPU 的資源，拖延畫布更新的時間。實際測試，平均每 0.166 秒才將 PlayingTime 增加 0.1 秒。而造成 PlayingTime 比實際播放的時間延遲很多，如圖 33 所示。

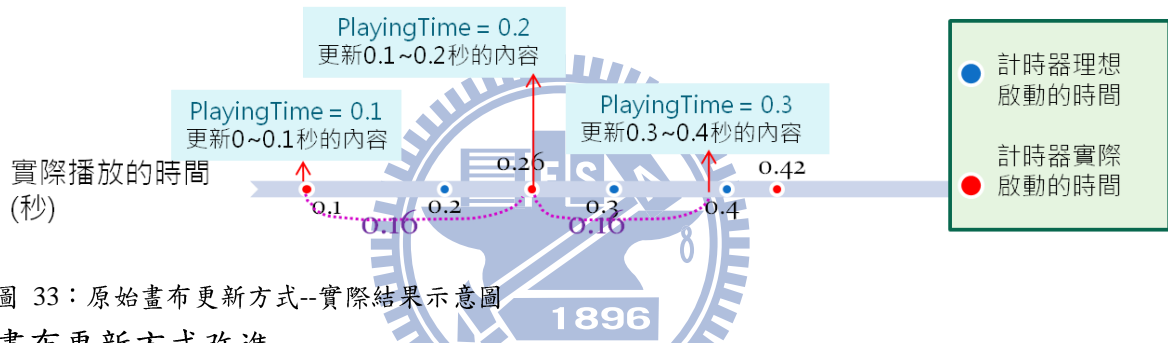


圖 33：原始畫布更新方式--實際結果示意圖

畫布更新方式改進

每一次更新畫布，不是固定更新 0.1 秒的內容。而是以系統時間計算實際播放的時間，並更新尚未繪製的內容。

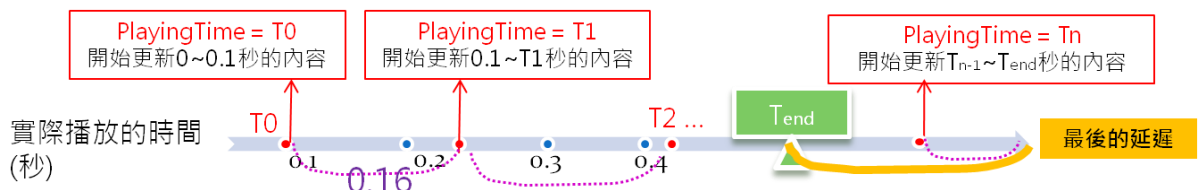


圖 34：畫布更新方式改進方法

PlayingTime 計算方式如下：

$$\text{PlayingTime} = \text{System.currentTimeMillis() - firstTimePoint} - \text{pauseTimePeriod}$$

開始播放到現在經過的時間
暫停的時間長度

目前的系統時間
開始播放的系統時間

- System.currentTimeMillis()：目前系統的時間。
- firstTimePoint：第一次按下播放按鈕的系統時間。
- pauseTimePeriod：暫停所經歷的時間。

間。如圖 36 所示，假設上一個時間點的螢光筆繪製座標為，目前時間點的繪製座標為。

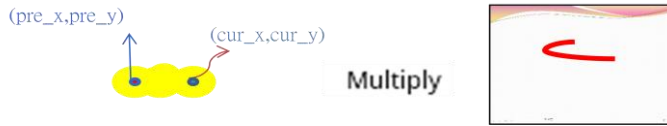


圖 36：螢光筆混色--原始方法示意圖

改進的做法

只處理螢光筆繪製到的區域，如圖 37 所示。每次混色前，利用螢光筆繪製的座標位置、與畫筆寬度(`paintSize`)，計算出螢光筆繪製到的區域(`left, top`)、(`right, bottom`)，並將之繪製於另一張與此區域同大小的畫布上。當混色時，只要處理螢光筆繪製到的區域即可。

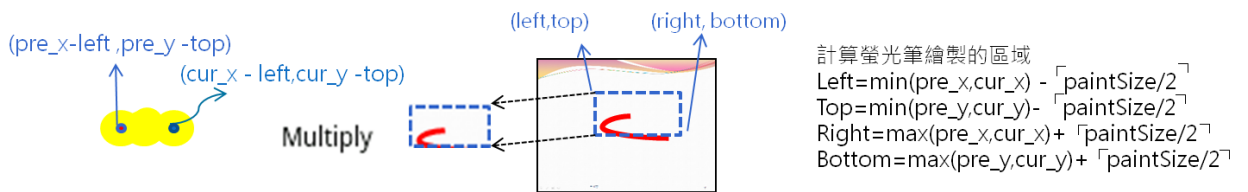


圖 37：螢光筆混色--改進方法示意圖

4.5.3 聲音檔案播放

講解者的講解過程的聲音會每 10 秒切成一個檔案做儲存。也就是說，一個 `bst` 檔可能會有好幾個聲音檔。因此需連續播放多個聲音檔。

原本的做法如圖 38 所示。當目前的聲音檔播放完畢後，檢查是否有下一個聲音檔要播放，如果有的話，則載入、播放。實作後發現，在聲音檔切換的過程中，會有聲音明顯中斷的現象。

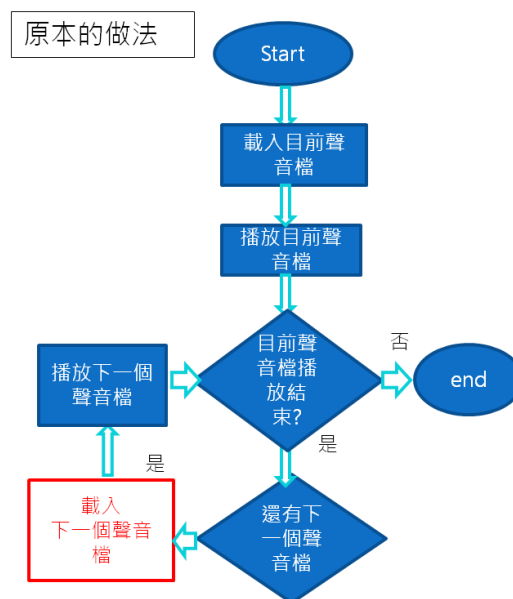


圖 38：聲音播放--原始做法

因此，本研究採用預先載入的機制。當播放目前的聲音檔時，便開始檢查是否有下一個聲音檔需播放，如果有話，則預先載入。待目前的聲音檔播放完畢，便可直接播放下一個檔案，而不需要做載入的動作，藉此減緩延遲的狀況。

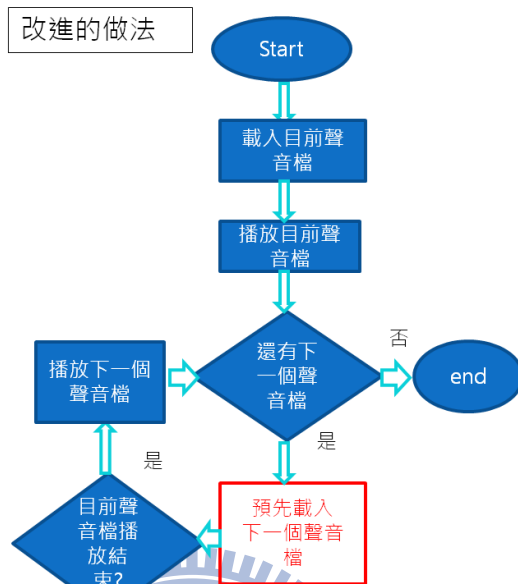


圖 39：聲音播放--改進做法

五、系統展示

5.1 多媒體講解呈現教材的錄製與發佈

5.1.1 多媒體講解呈現教材的錄製

1. 開啟講解手 3.3 版本，選擇[檔案]->[新增講解檔]，選擇要匯入的投影片。

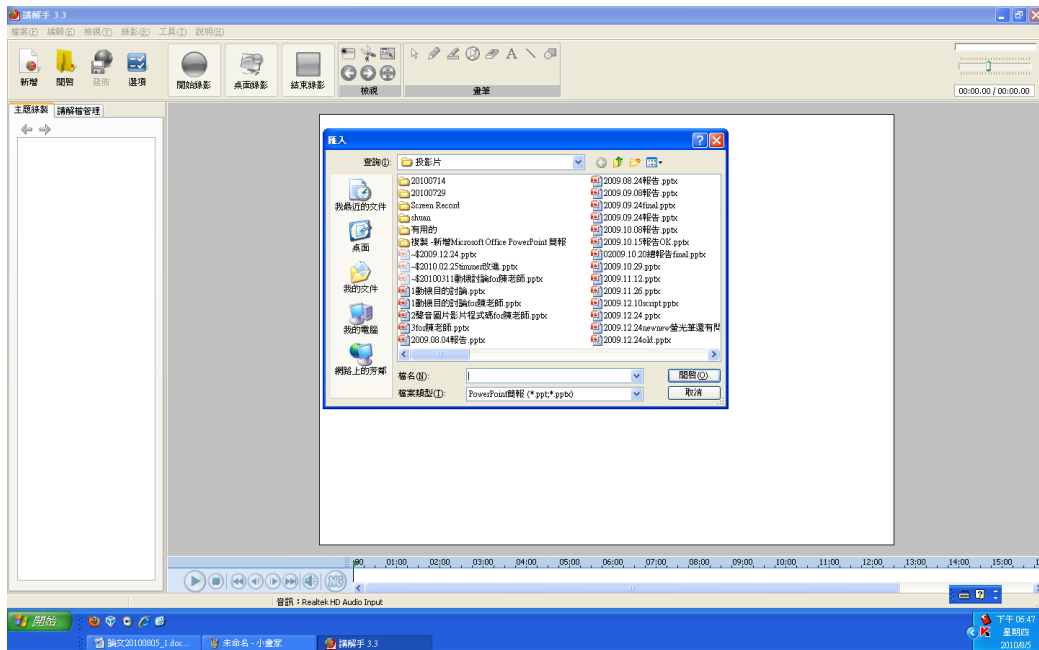


圖 40：教材錄製--匯入投影片

2. 投影片匯入成功，按下[開始錄影]，便可開始錄影。



圖 41：教材錄製--投影片載入成功

3. 在錄影的過程中可以使用塗鴉等功能，並使用麥克風錄製聲音。

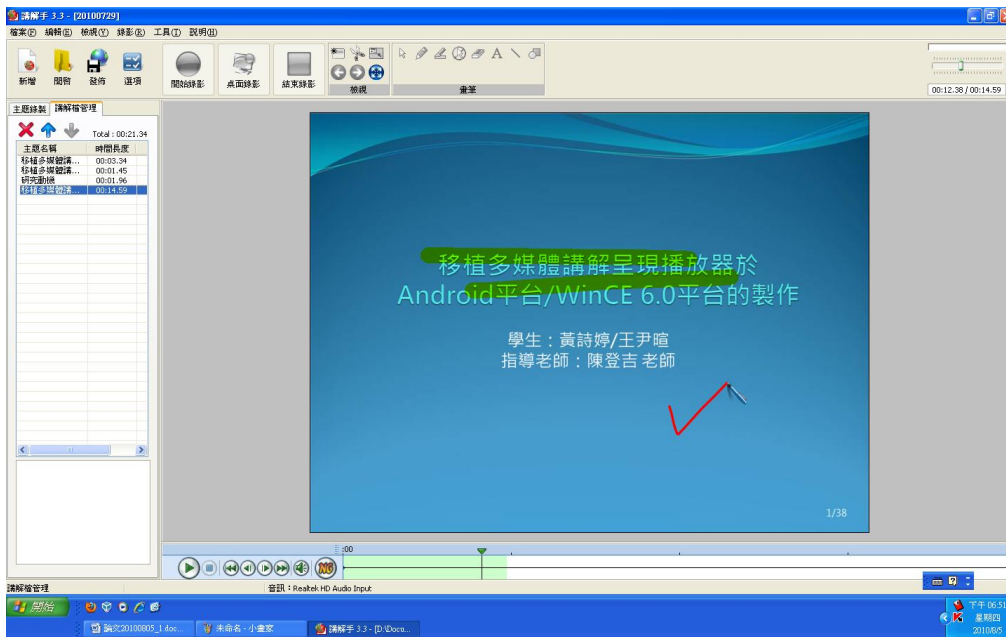


圖 42：教材錄製--錄製畫面

5.1.2 多媒體講解呈現教材的發佈

1. 錄製完畢可點選左上方之[發佈]，將講解檔發佈於本機端或上傳於部落格。



圖 43：教材發佈--發佈畫面

發佈於部落格

1. 選擇發佈的平台，本研究選擇發佈於樣板國度。

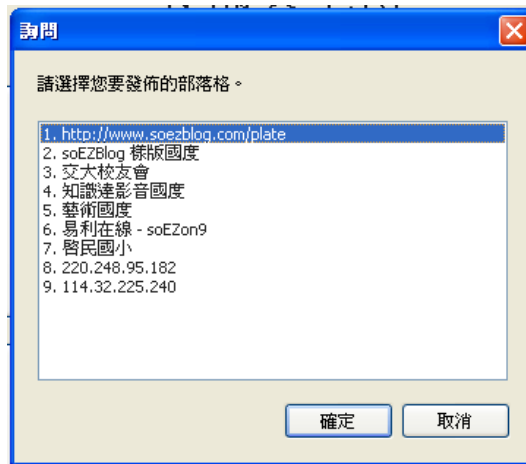


圖 44：發佈教材於部落格--選擇發佈的部落格

2. 填寫使用者的帳號密碼，以確定使用者有發佈權限。

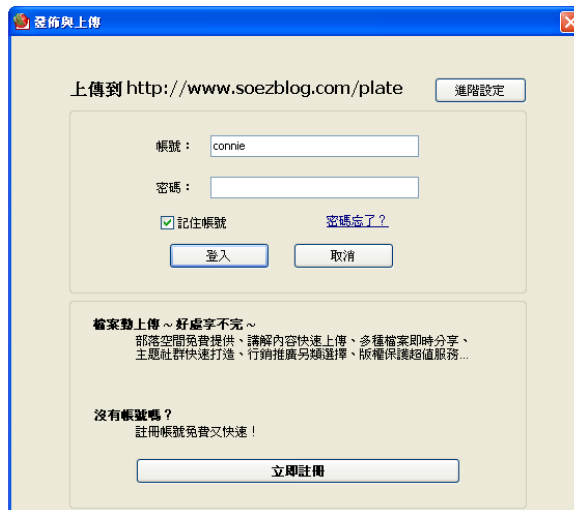


圖 45：發佈教材於部落格--輸入使用者帳號密碼

4. 填寫發佈資訊，如講解檔資訊、播放設定等欄位。



圖 46：發佈教材於部落格--填寫發佈資訊

4. 選擇使用的面板

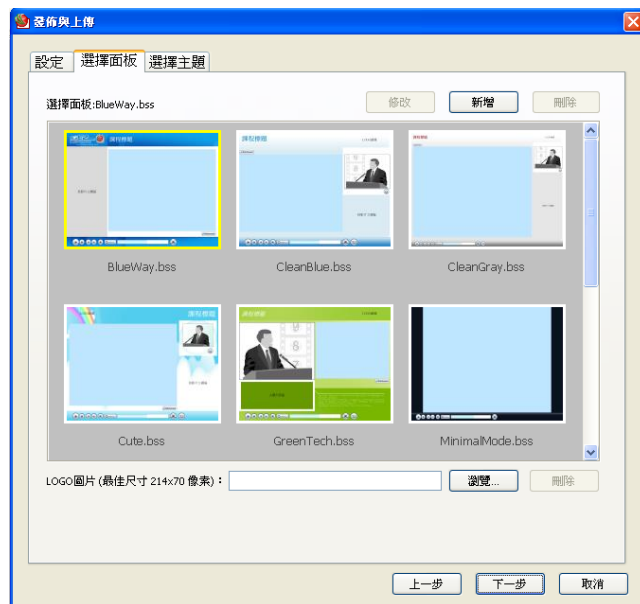


圖 47：發佈教材於部落格--選擇面板資訊

5. 確認要發佈的主題。按下確定後，講解手會自動將解講檔上傳於使用者的部落格。

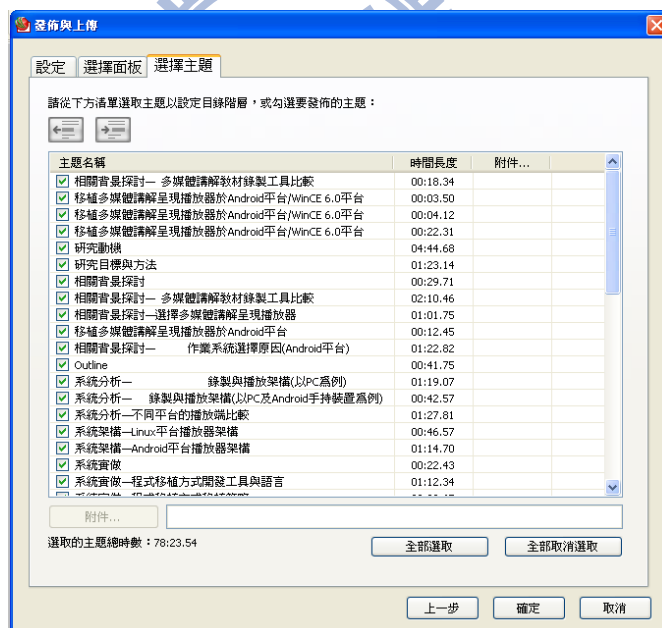


圖 48：發佈教材於部落格--選擇主題

發佈到本機儲存端

1. 同樣地，需填寫發佈資訊、選擇面板、確認要發佈的主題。

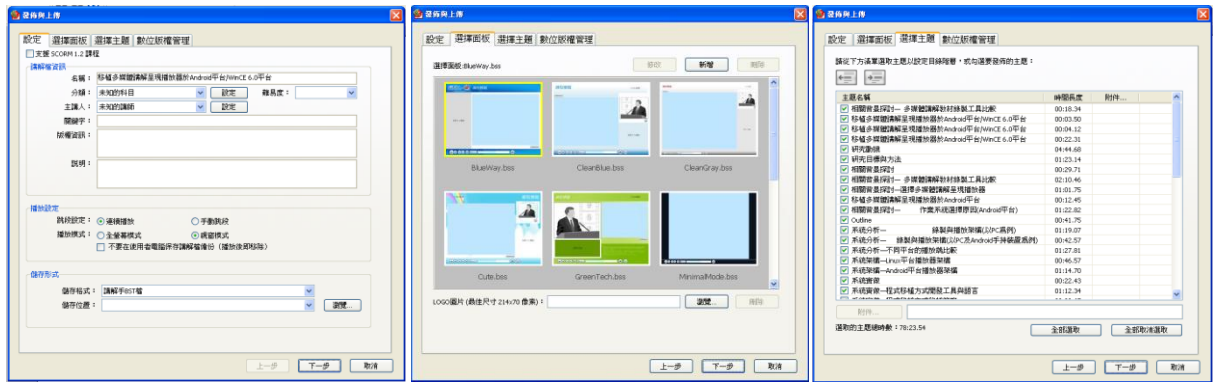


圖 49：發佈教材於本機端--填寫資訊

2. 發佈於本機端可以選擇加入 DRM 保護機制，使用者填寫 DRM 帳號密碼後，就可享有 DRM 保護機制。按下[確定]，便開始發佈檔案。

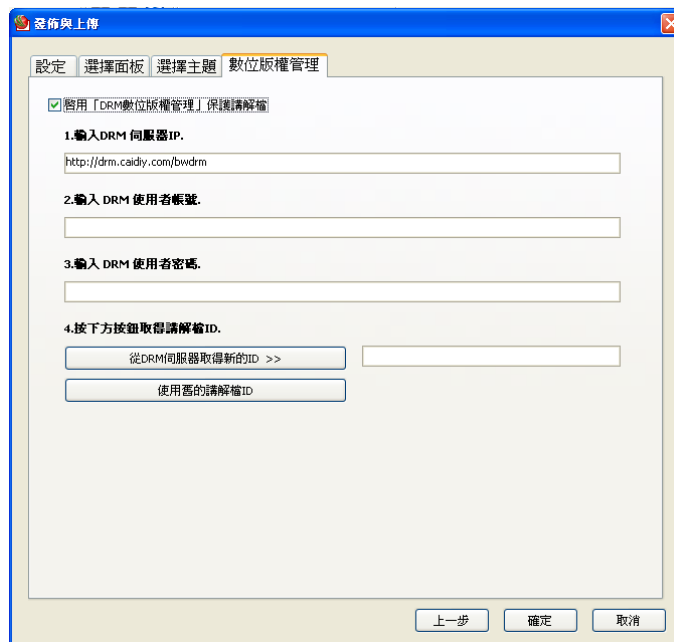


圖 50：發佈教材於本機端--填寫 DRM 資訊

5.2 Windows 平台上播放

5.2.1 播放本機端的講解檔

1. 使用者瀏覽本機端的檔案，點選任何一個 bst 檔。

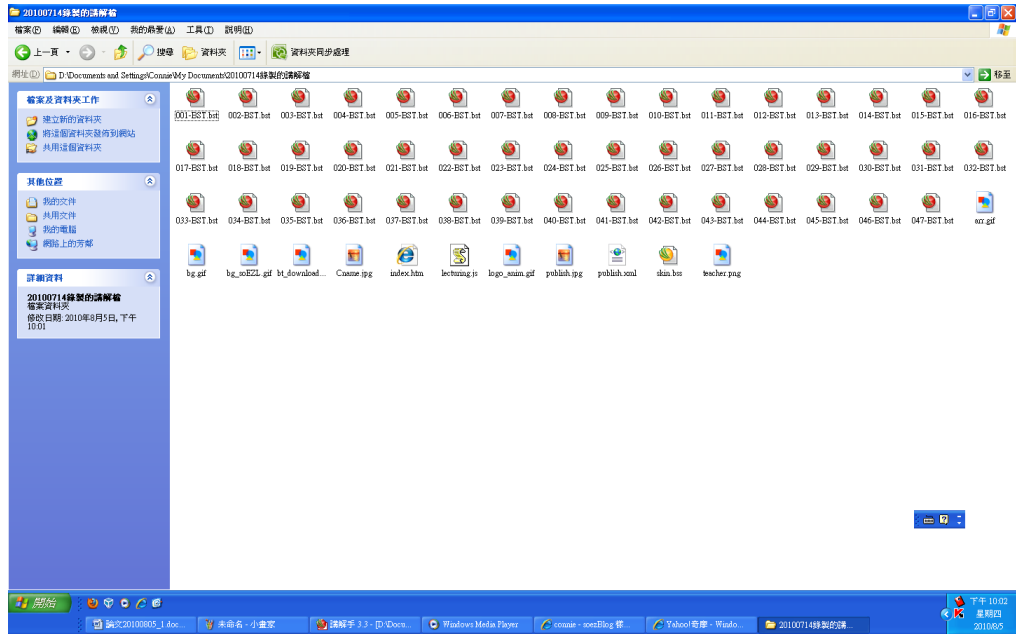


圖 51：播放本機端講解檔(Windows 平台)--本機端資料夾

2. 如果點選的檔案有受到 DRM 保護的話，則必須要輸入帳號密碼，通過驗證後，才可以開始播放。如果不需要直接跳至步驟 3。

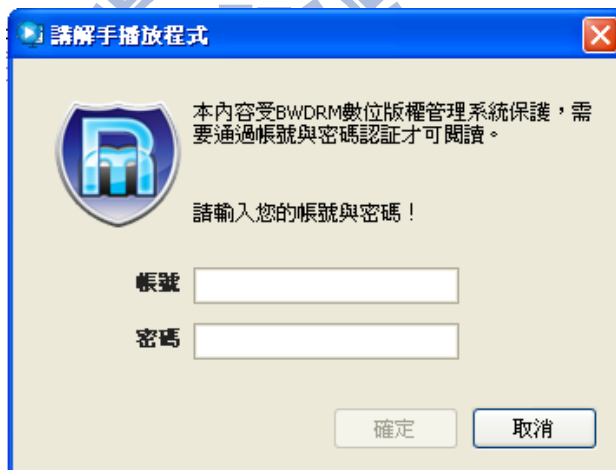


圖 52：播放本機端講解檔(Windows 平台)--DRM 驗證

3. 檢查是否有書籤檔，如果有，則詢問使用者是否從上次最後讀取的主題開始播放？選擇[是]，由書籤紀錄的位置開始播放；選擇[否]，則從第一個主題開始播放。

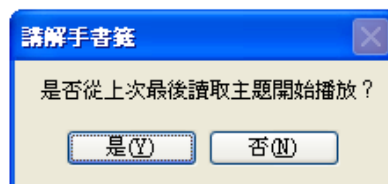


圖 53：播放本機端講解檔(Windows 平台)--書籤



圖 54：播放本機端講解檔(Windows 平台)--播放畫面

5.2.2 播放部落格上的講解檔

1. 使用者使用 IE 瀏覽器連線到樣版國度部落格，便可瀏覽部落格內的講解檔。



圖 55：播放部落格上的講解檔(Windows 平台)--部落格畫面

2. 瀏覽部落格上的講解檔資訊。按下播放鍵，會自動開啟講解手播放器。



圖 56：播放部落格上的講解檔(Windows 平台)--講解檔網路上的資訊

3.同樣地，會檢查 DRM 資訊及書籤檔，才會開始播放。

5.3 Android 平台上播放

不管播放本機端或部落格上的檔案，都要先開啟播放器。

1. 開啟播放器。

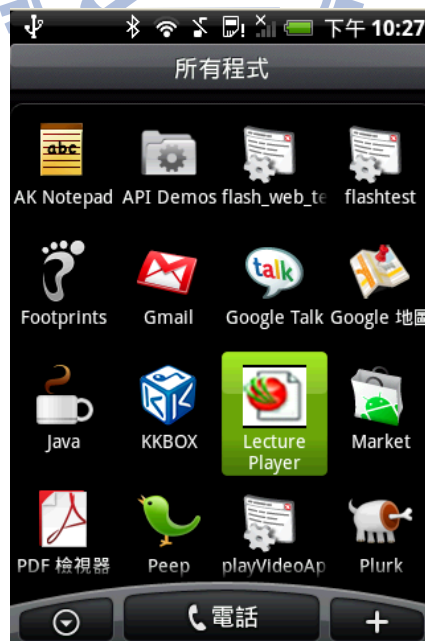


圖 57：播放講解檔(Android 平台)—開啟播放器

2.選擇要播放本機端檔案或網路端的檔案。



圖 58：播放講解檔(Android 平台)—選擇播放本機端/網路端的檔案

5.3.1 播放本機端的講解檔

1. 瀏覽本機端的檔案，並點選欲播放的 bst 檔。

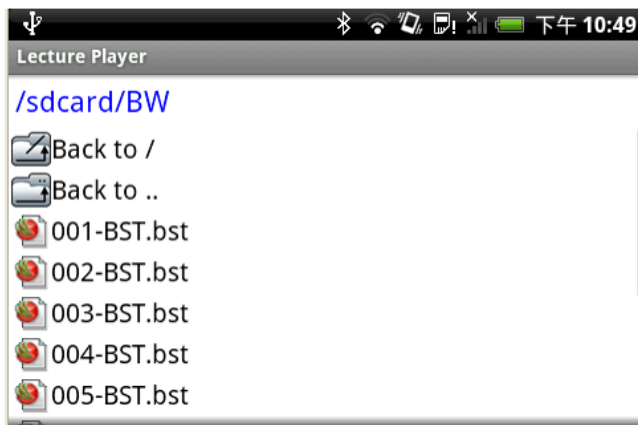


圖 59：播放本機端的講解檔(Android 平台)--檔案選擇清單

2. 如果點選的檔案有受到 DRM 保護的話，則必須要輸入帳號密碼，通過驗證後，才可以開始播放。如果不需要直接跳至步驟 5。

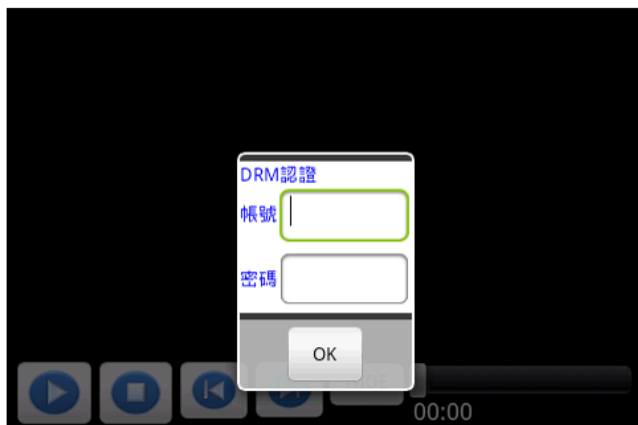


圖 60：播放本機端的講解檔(Android 平台)--DRM 驗證

5. 檢查是否有書籤檔，如果有，則詢問使用者是否從上次最後讀取的主題開始播放?選擇[是]，則由書籤紀錄的位置開始播放；選擇[否]，則從第一個主題開始播放。



圖 61: 播放本機端的講解檔(Android 平台)--書籤功能

6. 開始播放。



圖 62: 播放本機端的講解檔(Android 平台)--播放畫面

7. 播放時，呈現塗鴉、滑鼠游標的畫面。

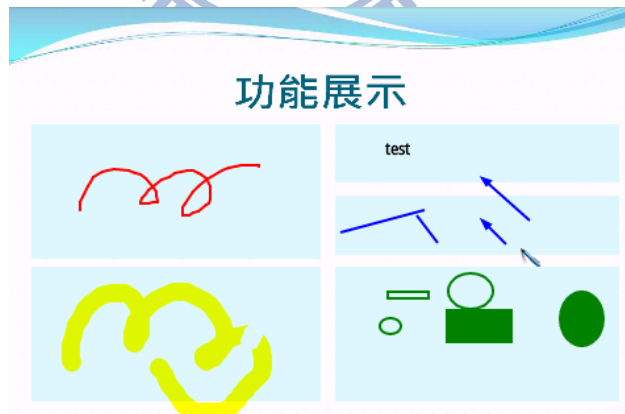


圖 63: 播放本機端的講解檔(Android 平台)--塗鴉、滑鼠呈現畫面

8. 呈現電子白板的畫面。

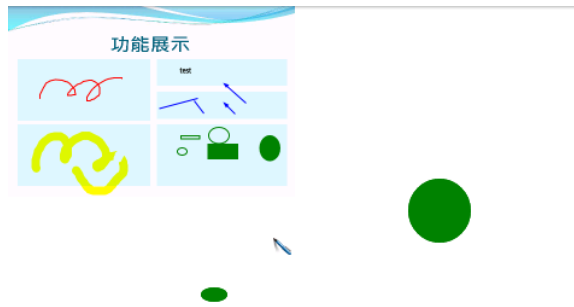


圖 64：播放本機端的講解檔(Android 平台)--電子白板呈現畫面

5.3.2 播放部落格上的講解檔

1. 開啟播放器，並選擇播放網路端的檔案。
2. 播放器會自動連線到樣板國度部落格。使用者可瀏覽部落格上的檔案。



圖 65：播放部落格上的講解檔(Android 平台)--部落格畫面

3. 點選開始播放按鈕。



圖 66：播放部落格上的講解檔(Android 平台)--講解檔網路上的資訊

4. 播放器則會開始檢查 DRM 資訊及書籤資訊，再開始播放檔案。

5.3.3 功能展示

功能選單

按手機上的[menu]鍵，便會秀出功能選單。再按一次便可隱藏。



圖 67：播放器功能(Android 平台)--功能選單

播放控制列

控制列上提供播放的基本功能：播放/暫停、停止、上一頁、下一頁、隱藏控制列、播放時間軸。當按下控制列上的[HIDE]按鈕，便可隱藏控制列。如需開啟控制列，則可藉由功能清單上的第一個選項[開啟控制列]開啟。



圖 68：播放器功能(Android 平台)--播放控制列

課程清單

按下功能清單上的[顯示課程清單]，便會秀出講師資訊及包含的主題資訊，如需要跳頁，可直接點選主題。

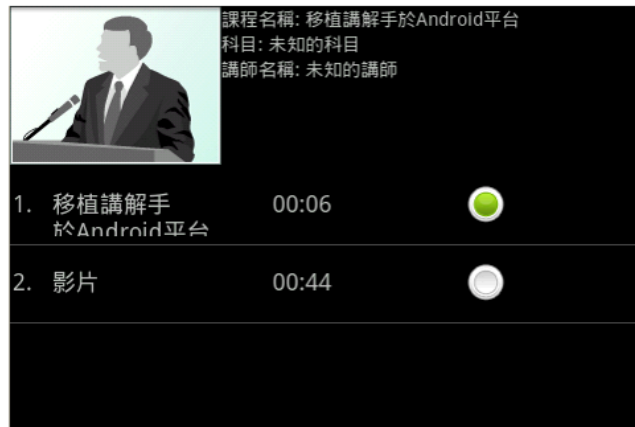


圖 69：播放器功能(Android 平台)--課程清單

切換螢幕長寬比例、方向

按下功能清單上的[切換長寬比例]便可切換為正常比例播放或全螢幕播放。再搭配變換手機的方向，便可改變直式播放或橫式播放。



圖 70：播放器功能(Android 平台)--畫面比例、方向切換

六、總結

6.1 結論

本研究分析目前市面上相當熱門且成功的數位內容服務架構，設計多平台之行動數位學習架構。搭配既有的教材錄製工具、數位學習部落格，並移植多媒體講解呈現播放器於 Android 平台，以期讓使用者可使用新興的 Android 平台之裝置進行數位學習。而選擇搭配既有的教材錄製工具、數位學習部落格有以下兩點好處：

1. 教材不需經過修改便可觀看，可以減少講解者製作教材的負擔。
2. 講解者可以使用原本的工具，不需要額外學習新的工具及平台。

本研究除了分析、解決移植播放器於 Android 平台需要注意到的系統、函式庫支援差異等問題外，亦探討在 CPU 能力較差的手持裝置上須注意到的效能改進等議題。本研究提出了三點改進方式，以達到良好的呈現效果：

1. 畫布更新時間計算方式改進：
利用系統時間計算實際播放的時間，避免因為畫布畫製時間過長與 Timer 啟動延遲等因素，而造成播放畫面延遲的現象。
2. 螢光筆混色方法改進：
在螢光筆筆跡混色前，先將需要混色的區塊切割出來再做混色的動作，以避免不必要的運算，而導致畫面延遲。
3. 聲音檔案播放方式改進：
在播放目前聲音檔的過程中，預先載入下一個將要播放的聲音檔，以減緩聲音檔切換的過程中，聲音明顯中斷的現象。

近年來，越來越多的裝置皆搭配了 Android 平台，除了智慧型手機之外，尚有數位家庭、嵌入式系統等裝置，相信會有越來越多使用者可以受惠於此服務。

6.2 未來展望

Android 平台目前對部分多媒體檔案支援仍不足，例如：無法播放大解析度的 WMV 檔、無法指定播放本機端的 Flash 檔。因此，本研究尚未針對此二類型檔案做處理。由於此二種類型檔案的內容協調轉換需要消耗較大的資源，並不適合於資源較為薄弱的手持裝置，所以將其設計在教材錄製端或 Content Server 上轉換。



圖 71：未來展望示意圖

轉換的方式是採用靜態內容協調，事先針對不同的平台轉換出適合的檔案，並儲存下來，以減少學習者等待的時間(例如：轉換檔案所花費的時間)。並加入平台偵測之機制，當使用者使用不同的裝置觀看時，將網頁內容協調，以呈現適合使用者瀏覽之格式，並自動選擇適合的講解檔讓使用者下載。

另外，目前播放器尚未實作動態演員呈現的功能(例如：特效演員、內嵌影片演員)，希望未來可以加入，以呈現更完善的播放畫面。最後，期望利用論文提出之多平台行動數位學習之架構，將多媒體講解呈現播放器移植於其它廣泛被使用的平台。

參考文獻或資料

- [1] 資策會講師群，「數位學習最佳指引」，資策會教育訓練處編印，第1-2頁，2003。
- [2] C. Quinn, "mLearning: Mobile, Wireless, In-Your-Pocket Learning," 2000.
- [3] 江明涓、劉晃溢，「Mobile e-learning實行技術研究與產業應用案例探討」，2004台灣商管與資訊研討會，台北大學三峽校區，2004。
- [4] J.D. Mooney, "Strategies for Supporting Application Portability," IEEE Computer, vol.23, issue 11, p. 59-70, Nov. 1990.
- [5] B. Vitor, T. A. Maria, "Context-Aware Content Adaptation: Standards and Research Trends," 2009,
http://www.visnet-noe.org/pdf/V_Barbosa-and-M_T_Andrade.pdf
- [6] S. R. Subramanya, B. K. Yi, "Digital rights management," IEEE Potential, 25(2): 31-34, March/April 2006.
- [7] Android|Official Website, <http://www.android.com/>
- [8] Open handset alliance, <http://www.openhandsetalliance.com/>
- [9] Android developers, <http://developer.android.com/index.html>
- [10] Walkingice, "How Android Differs from GNU/Linux?" COSCUP, 2009.
http://walkingice.twbbs.org/~walkingice/files/document/coscup_2009_walki ngice.pdf
- [11] GTK+, <http://www.gtk.org/>
- [12] STLport, <http://www.stlport.org/>
- [13] zlib, <http://www.zlib.net/>
- [14] The XML C parser and toolkit of Gnome, <http://www.xmlsoft.org/>
- [15] Java Native Interface: Programmer's Guide and Specification,
<http://java.sun.com/docs/books/jni/download/jni.pdf>
- [16] 洪啟彰，「樣板式多媒體內容在手持行動裝置之情境及內容協調與呈現-以樣板式的多媒體英文試題為例子」，交大碩士論文，2006。
- [17] A. K. Dey, G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," Technical Report GIT-GVU-99-22,1999.
- [18] A. Celentano, O. Gaggi, "Context-aware design of adaptable multimodal documents,"2004.
- [19] A. K. Dey, "Understanding and Using Context," In Personal and Ubiquitous Computing Journal 5(1), p. 4 - 7, 2001.
- [20] 謝佳成，「多媒體講解呈現播放器於PDA行動裝置之設計及製作」，交大碩士論文，2007。
- [21] 張筱靈，「移植多媒體講解呈現播放器於Linux平台上的製作 - 內容協調與呈現技術」，交大碩士論文，2008。

- [22] Z. Lei, N. D. Georganas, "Context-based Media Adaptation in Pervasive Computing," Proc. of Canadian Conference on Electrical and Computer Engineering, 2001.
- [23] 林鴻哲，「移植多媒體講解呈現播放器於Linux平台的製作 - 資料保護與網路功能整合」，交大碩士論文，2008。
- [24] 台灣數位學習科技，PowerCam 5，<http://www.powercam.com.tw/>
- [25] 訊連科技，串流大師4.0，<http://tw.cyberlink.com/>
- [26] 智勝國際，講解手3.3，<http://www.caidiy.com/>
- [27] 王尹暄，「多媒體講解呈現播放器於WinCE 6.0裝置之設計及製作」，交大碩士論文，2010。

