

2014 軟體創作達人暑期成長營期末報告

團隊名稱：

身份別	姓名	學校	系別	連絡電話	電子郵件
指導老師	高宏宇	國立成功大學	資訊工程學系	0936562156	hykao@mail.ncku.edu.tw
隊長	陳冠斌	國立成功大學	資訊工程學系	0987953185	g21589@gmail.com
隊員 1	陳昱琦	國立成功大學	資訊工程學系	0953307068	tp6m4fu6250071@gmail.com
隊員 2	陳吉德	國立成功大學	資訊工程學系	0927953522	twooo333@hotmail.com
隊員 3					

原始專案內容

專案題目	HTML5 即時簡報聽眾系統
專案主持人	黃宇新
機構名稱	創新開放軟體專案開發
應用領域	<input checked="" type="checkbox"/> 雲端與社群服務 <input type="checkbox"/> 智慧感知與互動多媒體 <input type="checkbox"/> 行動終端應用 <input type="checkbox"/> 其他應用軟體: _____
技術規格	產品或服務定義： 透過 HTML5 技術，建造一個演講時給予聽眾和講者互動的即時系統。使聽眾可以透過簡報和會眾進行交流，此外，聽眾亦可使用系統取得講者簡報、做筆記或者針對簡報提出問題。
	技術功能描述： 透過 HTML5 以及 CSS3 為主要技術開發。
	輸入輸出資料： 系統使用者分為講者與聽眾，接受講者簡報資料以及相關資訊輸入，聽眾可輸入問題以及筆記。輸出則為簡報內容以及聽眾統整問題、留言內容等。
	作業環境： Android、iOS、Windows、Linux
	程式語言： JavaScript、HTML5、CSS
	介面顯示支援語言(可複選)： <input checked="" type="checkbox"/> 中文 <input checked="" type="checkbox"/> 英文 <input type="checkbox"/> 其他_____

目前完成進度 (詳細說明已完成項目功能，另外可用文字或影音檔(AVI 或 MP4 格式，最長三分鐘為限)補充說明系統完成雛型，請附件一併上傳)：70%

➤ **本專案 Github 網址：**<https://github.com/g21589/Prezan>

➤ **已完成之各項功能說明：**

- **Node.js & Apache 伺服器架設**

以 Apache 來架基本網頁伺服器，演講者與聽眾透過該網頁伺服器來取得投影片內容。另外，對於需要大量請求的及時同步連線則使用 WebSocket 協定，並透過伺服器端的 Node.js 來接收並對所有聽眾進行廣播。

- **演講者與聽眾端的投影片切換同步功能**

本系統中的 HTML5 投影片以 Reveal.js 來完成，在演講者切換投影片時，會送出切換事件，透過 Node.js 伺服器把它廣播給所有聽眾端，聽眾端收到後在做出切換投影片的動作，如此一來便可完成投影片切換功能。

- **滑鼠指標與點擊同步功能**

透過 WebSocket 把在演講者端擷取到的滑鼠相對座標送給所有聽眾端，在聽眾端以一個 div 來貼一個虛擬的滑鼠圖樣，並藉由收到的相對座標來改變放置位置。此外，滑鼠的點擊動作也會送給所有聽眾端，聽眾端利用 CSS3 Animation 製作出點擊特效，並觸發該位置上的 click 事件，如此可以製作出撥放或停止投影片上的影片功能。

- **演講者語音廣播功能**

演講者端傳送原始聲音資料(PCM 16bit signed integer 單聲道)到伺服器端(約 800Kbps = 100KB/s)，伺服器用 LAME 對音訊轉碼成 MP3 (bitRate: 32, SampleRate: 44100, 單聲道)，聽眾端使用 HTML5 Audio Tag 向 Node.js 所架設的 MP3 串流伺服器發請求，伺服器回應 MP3 串流標頭與音樂資料(約 32Kbps = 4KB/s)。

- **私人筆記系統**

使用 HTML5 WebStorage API 將儲存聽眾的筆記內容，為了延長儲存資料的生命週期，採用 localStorage 來儲存在瀏覽器當中。此外，為提供使用者輸入更多樣的筆記內容，採用所見及所得(WYSIWYG)的編輯器套件—jQuery TE，並設計讓筆記系統放在一個側邊欄，由使用者點擊之後彈出，以提升使用者體驗。

- **問答系統**

1. 利用按鈕在 audience 端設置一個按鈕，按鈕點開後可讓 audience 輸入想問的問題。
2. 按下 ask 按鍵後，audience 的問題會透過 socket.io，在 server 端觸發 QA 事件。
3. 當 QA 事件被觸發，server 會發送 audience 的訊息到 speaker 及 broadcast 到其他 audience，speaker 端接收到 QA 事件會有一個 alert 顯示問題給 speaker，speaker 必須按下確認才會使 alert 消失；而其他 audience 接收到 broadcast 出來的事件則會在右下角出現問題敘述，並在幾秒後自動消失。

在本專案中，為了實現講者與聽眾的即時簡報系統，我們採用以上架構圖，其中包含三大模組，分別詳述如下：

1. Server 端：我們的系統會架構在一個基於 Node.js 與 Apache 的伺服器，講者端與聽眾端都是透過這個伺服器溝通，其中 Apache 負責傳統的網頁伺服器，而大量連線請求的同步機制則透過 Node.js 以 WebSocket 技術來實現。
2. 演講者端：在講者端的 HTML5 網頁中，我們分成**主動功能**與**被動功能**，**主動功能**為講者能主動去更新投影片或產生資訊給聽眾的功能，**被動功能**是由聽眾所傳出的訊息，而講者可以看得見。
 - A. 主動功能：
 - i. 基本投影片控制：換頁
 - ii. 編輯投影片：講者可在需要時，隨時更改、編輯、新增、刪減投影片
 - B. 被動功能：
 - i. 顯示投票結果：在聽眾投票結束後，會直接統計票數，並輸出簡易投票結果表格
 - ii. 顯示提問：當聽眾端提出問題時，會即時在講者端顯示問題
 - iii. 顯示警示燈：若有許多觀眾都按下警示燈，表示講者報告速度過快，警示燈可以提醒講者放慢速度
3. 聽眾端：在聽眾端，我們將畫面分為兩個子畫面，一個子畫面為隨時與講者同步的投影片，另一個子畫面則可供聽眾自行捲動畫面，並在上面做筆記。
 - A. 同步功能：
 - i. 播放投影片：投影片會隨演講者換頁而同步更動
 - ii. 提問：隨時可向演講者提問，提出的問題因為會顯示在講者的畫面上，因此其他聽眾也能看到問題
 - iii. 警示燈：如果演講者報告速度過快，聽眾可以按下警示燈提醒講者放慢腳步
 - B. 非同步功能
 - i. 筆記：提供聽眾隨時做筆記，最後會輸出為一個檔案

為了增進效能，溝通的部份採用 HTML5 的 WebSocket 技術，在伺服器端與用戶端雙方做主動式溝通，而不用傳統的 Long Polling 等方式。

表 1：系統規格

類型	規格
CPU	Intel(R) Xeon(R) CPU 3050 2Cores 2.13GHz
RAM	2 GB
HDD	320 GB
OS	Ubuntu server 14.04 LTS

➤ 對於串流壓縮語音的部分有測試兩種做法：

1. WebSocket 版

- 演講者端傳送原始聲音資料(PCM 16bit signed integer 單聲道)到伺服器端(約 800Kbps = 100KB/s)
- 伺服器用 LAME 對音訊轉碼成 MP3 (bitRate: 32, SampleRate: 44100, 單聲道)
- 用 Socket.io 廣播給所有聽眾 (約 32Kbps = 4KB/s)
- 聽眾端用 HTML5 Web Audio API 的 decodeAudioData()解碼並撥放

2. HTTP 版

- 演講者端傳送原始聲音資料(PCM 16bit signed integer 單聲道)到伺服器端(約 800Kbps = 100KB/s)
- 伺服器用 LAME 對音訊轉碼成 MP3 (bitRate: 32, SampleRate: 44100, 單聲道)
- 聽眾端使用 HTML5 Audio Tag 向 Node.js 所架設的 MP3 串流伺服器發請求，伺服器回應 MP3 串流標頭與音樂資料 (約 32Kbps = 4KB/s)

表 2：語音同步之兩方案比較

	WebSocket	HTTP
優點	<ul style="list-style-type: none">• 聲音延遲較短，約 2~3 秒	<ul style="list-style-type: none">• 因為緩衝的關係，聲音非常平順。• 瀏覽器支持 HTML5 Audio tag 的 MP3 格式就可撥放 (Chrome, Firefox, IE, ...)
缺點	<ul style="list-style-type: none">• 傳送的兩個聲音片段間會有中斷發生。• 只支援 Chrome(在目前 Firefox 中的 decodeAudioData()函數似乎認不出 MP3 格式)。	<ul style="list-style-type: none">• 聲音延遲較長，約 5~10 秒。

經過討論，發現 WebSocket 也是包在 HTTP 中做傳輸，如此，兩者僅差在資料緩衝的方式，因此選擇 HTTP 的方式來廣播音樂，並記錄緩衝時間來把聽眾端的動作時間往後延遲該緩衝時間，來達成影音同步的效果。

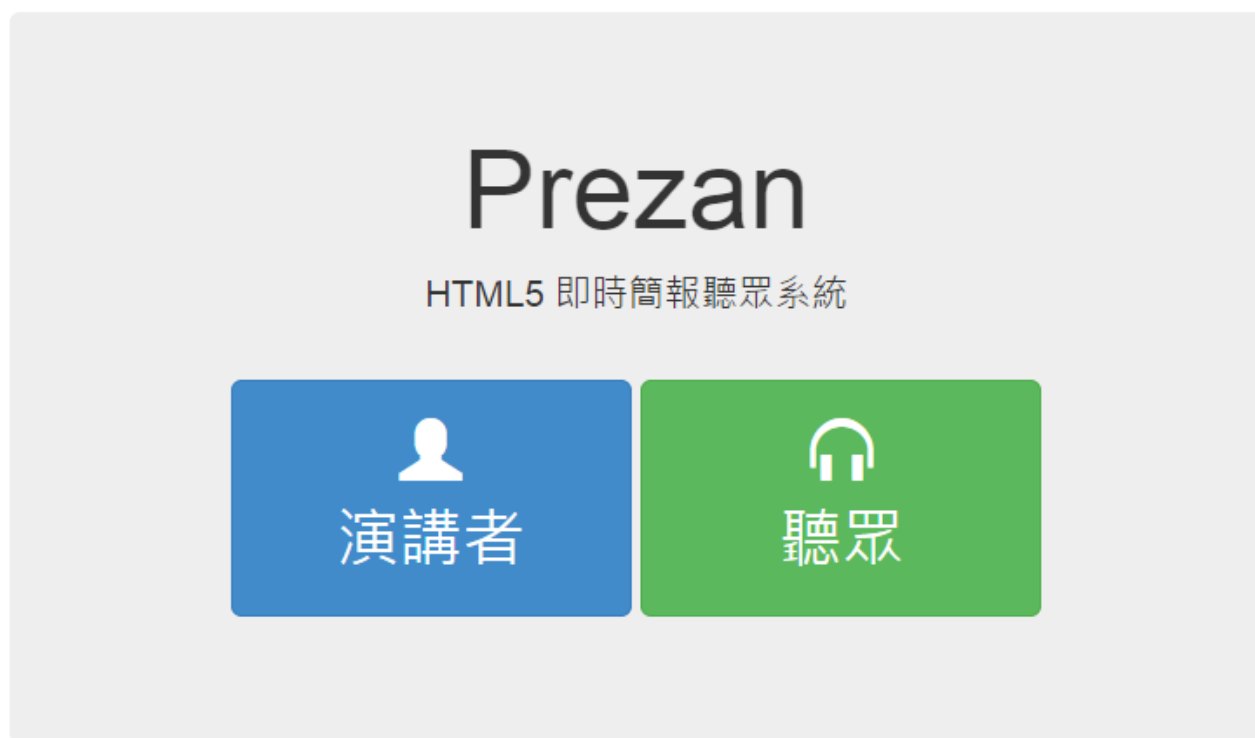


圖 1：入口頁面(可選擇是您的腳色是演講者還是聽眾)

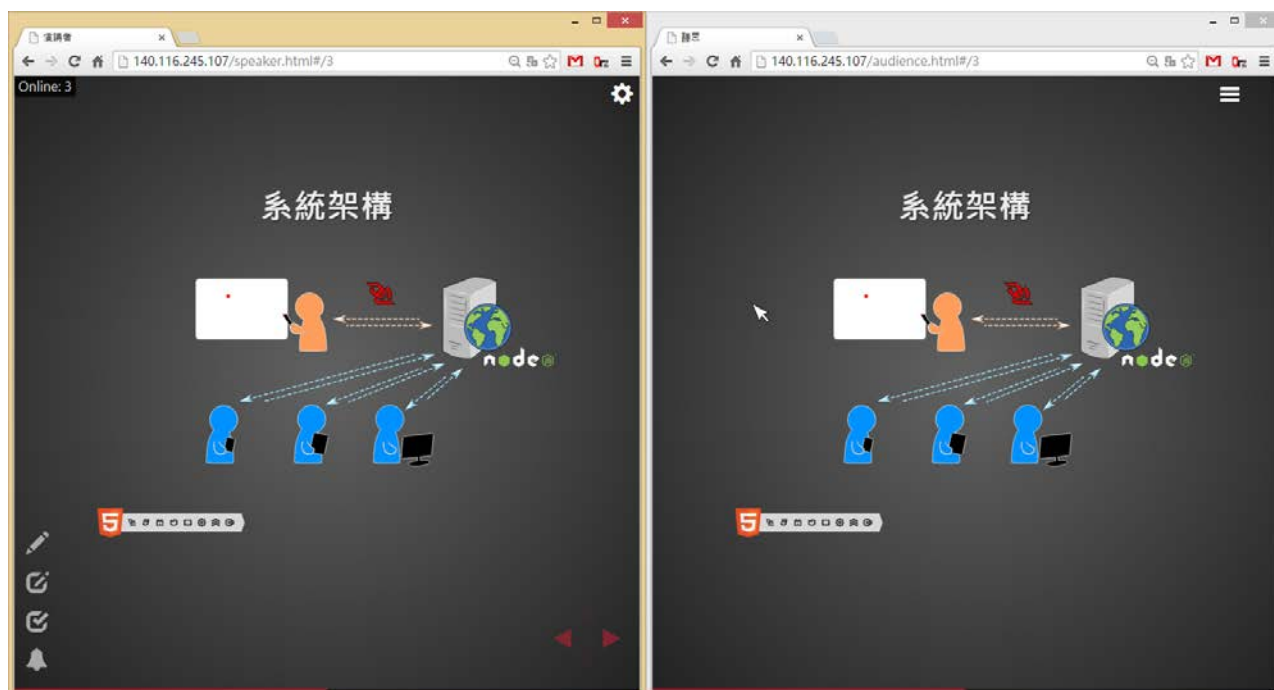


圖 2：演講者與聽眾同步



圖 3：編輯投影片

軟體清單（專案開發所需相關軟體元件清單）

1. Reveal.js (<https://github.com/hakimel/reveal.js>)
2. Node.js (<http://nodejs.org/>)
3. Socket.IO (<http://socket.io/>)
4. jQuery (<http://jquery.com/>)
5. jQuery TE (<http://jqueryte.com>)
6. Slidebars (<http://plugins.adchsm.me/slidebars>)
7. alertify.js (<http://fabien-d.github.io/alertify.js>)
8. Node-LAME (<https://github.com/TooTallNate/node-lame>)
9. Bootstrap (<http://getbootstrap.com/>)
10. TinyMCE (<http://www.tinymce.com/>)
11. Raphaël (<http://dmitrybaranovskiy.github.io/raphael/>)
12. 瀏覽器(Firefox、Chrome、Safari、Opera、IE)
13. 文字編輯器(Vim、Sublime Text、Notepad++)

專案成果預定授權條款（請務必依循與尊重現有專案的授權模式，避免不相容的議題）

項目	授權方式	程式碼來源
Reveal.js	MIT license	https://github.com/hakimel/reveal.js
Node.js	MIT license	https://github.com/joyent/node
Socket.IO	MIT license	https://github.com/LearnBoost/socket.io
jQuery	MIT license	https://github.com/jquery/jquery
jQuery TE	MIT license	http://jqueryte.com
Slidebars	MIT license	https://github.com/adchsm/Slidebars
alertify.js	MIT license	https://github.com/fabien-d/alertify.js

Node-LAME	MIT license	https://github.com/TooTallNate/node-lame
Bootstrap	MIT license	https://github.com/twbs/bootstrap
TinyMCE	LGPL 2.1	https://github.com/tinymce/tinymce
Raphaël	MIT license	https://github.com/DmitryBaranovskiy/raphael

本專案預計採用 MIT license 授權條款。

專案開發討論紀錄（參與團隊須全歷程專案專案管理開發討論紀錄平台網址）：10%

2014 軟體創作達人暑期成長營第一階段專案開發

第 1 次輔導諮詢討論紀錄表

日期：2014 年 6 月 21 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新

他們目前有的技術

- webRTC web 通訊 目前有視訊會議
- 簡報同步 4K 畫面 用多台螢幕同時 show
- 透過 socket 請每個機器播不同片段的影片(socket 他們有做過)
- 視訊會議: node.js, streaming 點對點

我們需要的技術

- 目前不需要處理影像，只處理音訊
- 用 reveal.js 簡報同步
- 一對多(主講者對聽眾)

他們的技術與我們需要的技術之差異

- 他們做的是 p2p, 我們要做的是一對多
- 類似網路廣播電台模式也 OK：
- 臨時 create 一個房間,把大家加進來

將 webRTC 移轉到我們系統上的可能狀況

- 可能不太合用，因為 webRTC 是 p2p, 未必能用一對多(這部分可以再 survey 看看)
- webRTC 可能也可以針對音訊處理
- 黃經理要問一下其他同事關於 webRTC
 - 也許可以約時間去他們公司討論 介紹他們目前的技術
- 在簡報中播影片：可能講師的影片已經播完了 結果其他人同步還沒播完
 - *reference for webRTC:*
 - <http://www.webrtc.org/reference/architecture>

建議

- 只同步簡報很可惜，同步投影片中的影片吧
- 目前可以只考慮在區域網路中的同步機制，不考慮 internet 同步，所以應該不難

未來開發模式討論

- GitHub=> 創專案，讓經理有存取權限，markdown
 - 依循 TSOC 規範之 checkpoint 與黃經理討論
- markdown reference:
 - <https://github.com/acl-translation/acl-chinese/tree/master/zhTW>
 - <https://github.com/othree/markdown-syntax-zhtw/blob/master/syntax.md>
 - <http://www.yolinux.com/TUTORIALS/LinuxTutorialAudioStreaming.html>
- IceCast <-- Streaming MP3 (MPEG 1 Layer 3 Audio): Using an IceCast 1 streaming server (detailed in this tutorial) with xmms or WinAmp client

工作

1. 架 node.js 的 server (架在 245.107)
2. 完成基本簡報同步換頁功能(reveal.js & WebSocket)
3. Survey WebRTC

2014 軟體創作達人暑期成長營第一階段專案開發

第 2 次輔導諮詢討論紀錄表

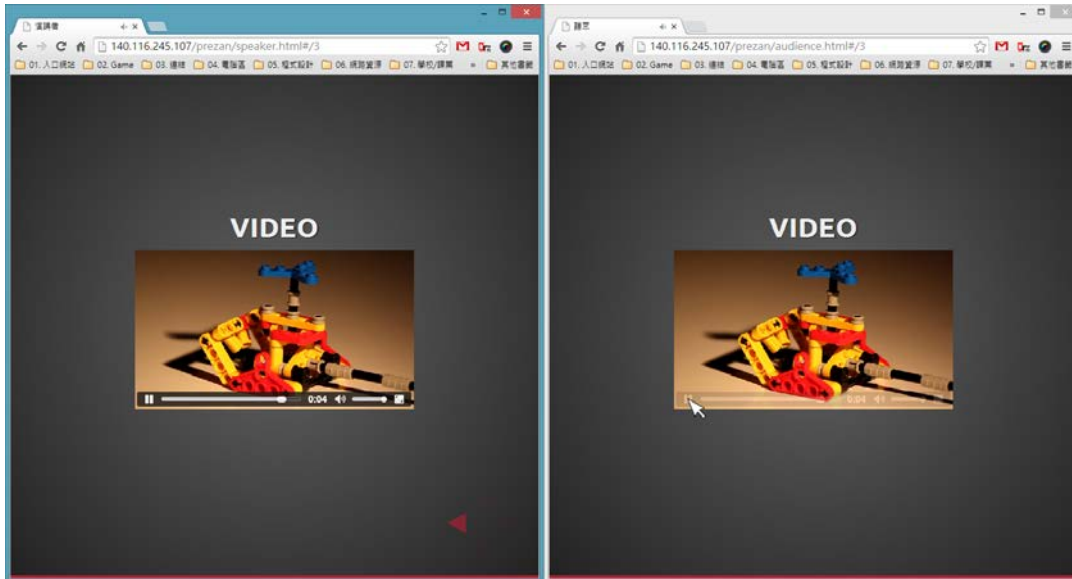
日期：2014 年 7 月 1 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新

Demo 上禮拜成品與問題討論

- demo 方式：演講者端由冠斌操控並講解，黃經理在聽眾端會看到投影片的同步效果
 - 演講者端：<http://140.116.245.107/prezan/speaker.html>
 - 聽眾端：<http://140.116.245.107/prezan/audience.html>
- Github 網址：<https://github.com/g21589/Prezan>
- 滑鼠動作同步時會有不準確的問題
 - 經理建議：可用 keyboard event 的方式傳送試試看
 - 這裡是什麼意思阿 就是放影片的時候用鍵盤放嗎?

- Yes 就類似鍵盤快捷鍵 按某個鍵就播影片
- 但這樣會不會不直覺 還要用鍵盤按才能播放
- 恩 而且這樣滑鼠不準的問題好像還是沒解決到



To be continue...

- 與聽眾的互動（message）的畫面呈現
 - 畫面
 - 用分割的方式 => 解析度的問題（四分割與九分割）
 - 滑出來，使用者打開才會跑出來（平常隱藏，有空才會出來，用疊上去的）=> 因為投影片比例還是比較重要，聊天室的部分不要同步
 - 模擬筆（可畫重點、塗鴉）
- streaming 的同步較為重要
 - 聲音
 - 影片
 - 影片的同步我們算已經有做了嗎？
 - 沒有 僅能播放
 - 不就是同步撥放的意思嗎？不然影片同步什麼意思
 - 時間軸吧 比如說要考慮到網路延遲時的問題
 - 喔喔了解
 - 聽眾端的影片播放，需要讓他不能被控制嗎
 - 應該演講者可以滑時間軸並同步到聽眾端，但聽眾端也可以滑時間軸且不會同步到演講者端，然後就爆炸了XD 打完發現衝突

下次實際 Meeting

- 見面地點
 - 我們到導師公司：導師可以直接請相關技術的同事給予指導
 - 請導師來台南：僅能由導師給予一些建議，而無法請同事來分享
- 交通方式
 - 搭火車到火車站 再轉捷運到三多商圈站 再走路 5 分鐘

2014 軟體創作達人暑期成長營第一階段專案開發

第 2 次輔導諮詢討論紀錄表

日期：2014 年 7 月 11 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新

07/11 Fri. 面談會議

1. 已有的現成套件 togetherjs: <https://togetherjs.com/>
 1. 多人滑鼠同步
 2. 多人音訊串流同步—採用 WebRTC
2. 聲音串流同步技術
 1. WebRTC
 - i. 演講者端與 Server 端結合，演講者端與每一個聽眾端建立連線
 1. 演講者端需要 encode 多條連線，效能有限，約在 3~9 人的 PC 上可行
 - ii. Server 端介接：需要搞 WebRTC 底層，有難度，依目前時程規劃不太可行
 2. IceCast 網路電台伺服器
 - i. 由演講者端送音頻到 Server 端，經由 IceCast 處理成串流後，送到聽眾端用 audio tag 直接撥放
 - ii. 可行、架的起來就 OK
 3. Flash 一般常用
3. EtherPad—即時共同編輯的編輯器（不過本專案沒有共同編輯之打算）
4. 介面要更潮、缺美工

2014 軟體創作達人暑期成長營第二階段專案開發

第 1 次輔導諮詢討論紀錄表

日期：2014 年 7 月 30 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新

進度報告

目前串流壓縮音訊的部分有兩種做法

2. WebSocket 版

- 演講者端傳送原始聲音資料(PCM 16bit signed integer 單聲道)到伺服器端(約 800Kbps = 100KB/s)
- 伺服器用 LAME 對音訊轉碼成 MP3 (bitRate: 32, SampleRate: 44100, 單聲道)
- 用 Socket.io 廣播給所有聽眾 (約 32Kbps = 4KB/s)
- 聽眾端用 HTML5 Web Audio API 的 decodeAudioData()解碼並撥放
 - 優：聲音延遲較短 約 2~3 秒
 - 缺
 - 傳送的兩個聲音片段間會有中斷發生
 - 只支援 Chrome (在 Firefox 中的 decodeAudioData()似乎認不出 MP3 格式)
- 可能的改進方式:
 1. 嘗試 Ogg Vorbis 格式
 2. 使用 JS 版的 MP3 decoder (JSMad -> 難用!?)

2. HTTP 版

- 演講者端傳送原始聲音資料(PCM 16bit signed integer 單聲道)到伺服器端(約 800Kbps = 100KB/s)
- 伺服器用 LAME 對音訊轉碼成 MP3 (bitRate: 32, SampleRate: 44100, 單聲道)
- 聽眾端使用 HTML5 Audio Tag 向 Node.js 所架設的 MP3 串流伺服器發請求，伺服器回應 MP3 串流標頭與音樂資料 (約 32Kbps = 4KB/s)
 - 優
 - 聲音非常平順(應該是緩衝的關係)
 - 瀏覽器支持 HTML5 Audio MP3 格式就可撥放 (Chrome, Firefox, IE, ...)
 - 缺：聲音延遲較長 約 5~10 秒

建議

- 讓 video delay 配合聲音的延遲
 - detect 聲音延遲的時間多久，讓投影片轉換也跟著延遲
- 如果是方法 1，延遲 2~3 秒，可以 tune 一個時間長度，讓影音同步，而聲音品質方面可以考慮自己寫 buffer
- 如果是方法 2，用更智慧的方法 detect 聲音與畫面延遲差多少
- 總之一定要有 buffer 才能避免斷音
- 黃經理建議：傾向於使用方法 2
 - 瀏覽器支援各方面較好
 - 但 detect 延遲時間必須動態的處理，動態的計算延遲時間，再將畫面延遲

分工

- 冠斌
 - 音訊同步
 - 介面美化
- 吉德
 - 塗鴉
- 昱琦
 - 投票

2014 軟體創作達人暑期成長營第二階段專案開發

第 2 次輔導諮詢討論紀錄表

日期：2014 年 8 月 14 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新

• 語音同步

關於語音同步的部分，目前採用上次討論時說的第二種作法，也就是 audio tag 的做法，在資料開始 buffer 時紀錄目前時間，聲音開始播放的時候再取一次時間，然後這兩個時間差，當作延遲時間。

• 介面

1. 新增入口頁面 (可選擇是演講者或是聽眾)
2. 在演講者端右上新增設定按鈕、左下角新增圖示按鈕 (使用 CSS3 特效)

• 更新投影片內容

打算直接呈現本專案作品的內容，DEMO 時順便使用它來報告。

• 筆跡註記

- 採用 SVG 作為顯示筆跡的方式
- 目前是每一次切換頁面會在該頁建立 canvas 供筆跡註記
- 之後會改成由 Speaker 點擊按鈕啟用筆跡註記功能，才建立 canvas
- 目前尚有 canvas 在視窗比例調整後，滑鼠點擊位置會有偏移的問題
- 尚未撰寫同步的功能

2014 軟體創作達人暑期成長營第二階段專案開發

第 3 次輔導諮詢討論紀錄表

日期：2014 年 8 月 20 日

專案題目	HTML5 即時簡報聽眾系統				
專案題目代號	CS05	團隊名稱	1KM	輔導老師	黃宇新
<ul style="list-style-type: none">● 新增功能<ol style="list-style-type: none">1. 線上編輯投影片 在演講者端可線上編輯投影片，並於編輯完成後同步給所有聽眾。2. 叫醒聽眾 (手機上) 演講者端加入"叫醒聽眾"的功能，按下去後會在所有聽眾端重複隨機時間的震動。 使用 Vibration API3. 聲控切換投影片上下頁 (Chrome) 透過 Web Speech API，辨識使用者的說話內容，在辨識後會得到辨識的文字結果與信心值，判斷文字為"上一頁"或是"下一頁"即可觸發切換投影片事件。● 黃經理的疑問<p>Q：語音同步的延遲時間與頻率</p><p>A：中斷其實不會很明顯</p>● 黃經理的建議<p>因為我們是做簡報系統，因此有簡報優勢，可用機器人簡報我們的簡報系統(利用事先錄音、錄影筆跡滑鼠，寫成 script 自動當場測試)</p><p>用影片會比較沒有說服力，但可以當備案。</p>					