2014 軟體創作達人暑期成長營期中報告

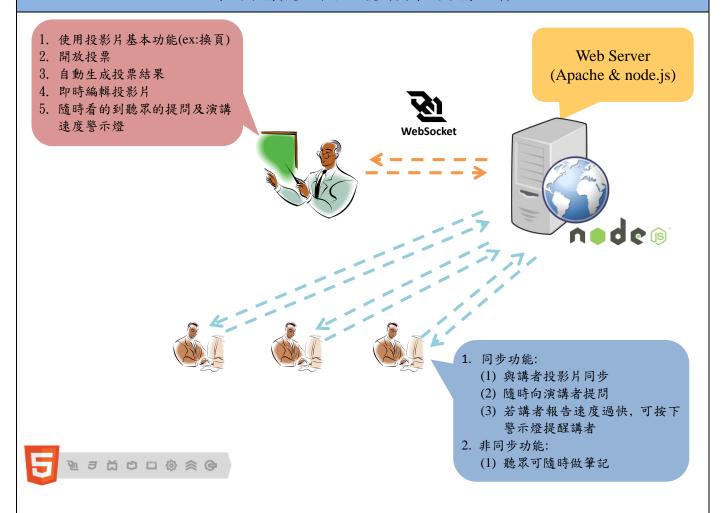
團隊資訊 姓名 系別 連絡電話 身份別 學校 電子郵件 指導教授 高宏宇 國立成功大學 資訊工程學系 0936562156 hykao@mail.ncku.edu.tw 隊長 資訊工程學系 陳冠斌 國立成功大學 0987953185 g21589@gmail.com 隊員1 陳昱琦 國立成功大學 資訊工程學系 0953307068 tp6m4fu6250071@gmail.com 隊員2 資訊工程學系 陳吉德 國立成功大學 0927953522 twooo333@hotmail.com 隊員3 專案內容 專案題目 HTML5 即時簡報聽眾系統 專案主持人 黄宇新 機構名稱 創新開放軟體專案開發 雲端與社群服務 □ 智慧感知與互動多媒體 □ 行動終端應用 應用領域 □ 其他應用軟體:_ 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: 技術規格 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):■ 中文 ■ 英文 □ 其他_

專案描述

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。

系統架構(使用平台、技術與系統架構設計)



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. 開放投票:講者可以在任何時間點舉行投票,開放時會即時顯示於聽眾端
- iii. 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。

第一階段完成進度(必須要有技術報告、程式碼及執行檔)

▶ 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

演講者端採用 HTML5 的 getUserMedia 取得麥克風的聲音輸入,接著把取到的聲音資料由 32bit 浮點數陣列降成 16bit 整數陣列以減少資料量,然後經由 WebSocket 直接傳送二進位的音頻資料到伺服器端,伺服器端收到資料後再廣播出去給各位聽眾端。經過實際測試,這樣的傳送方式在每個客戶端上約需 700~800kbps 的頻寬。

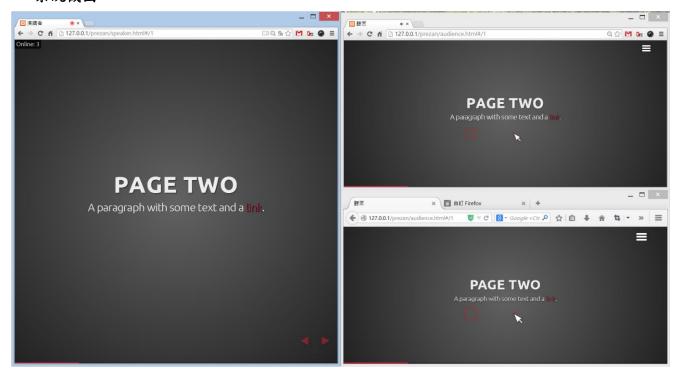
● 私人筆記系統

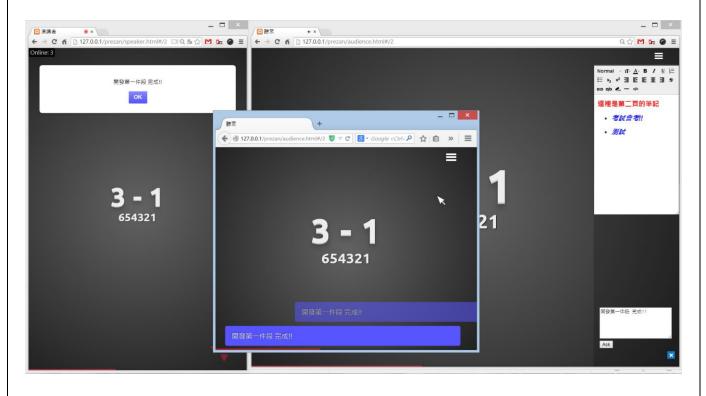
使用 HTML5 WebStorage API 將儲存聽眾的筆記內容,為了延長儲存資料的生命週期,採用 locaoStorage 來儲存在瀏覽器當中。此外,為提供使用者輸入更多樣的筆記內容,採用所見及所得(WYSIWYG)的編輯器套件—jQuery TE,並設計讓筆記系統放在一個側邊欄,由使用者點擊之後彈出,以提升使用者體驗。

● 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。

> 系統截圖





第二階段預備完成項目及內容描述

- 1. 完成投票系統。
- 2. 嘗試改善目前語音同步的品質與降低網路流量(例如壓縮原始 PCM 音頻資料為 MP3 再進行傳送)。
- 3. 改善介面並加入更多視覺特效。
- 4. 在不同的裝置平台上做實際測試。

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. 瀏覽器(Firefox、Chrome、Safari、Opera、IE)
- 9. 文字編輯器(Vim、Sublime Text、Notepad++)

專案成果預定授權條款

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

▶ 本專案預計採用 MIT license 授權條款。