內容

[壹、 研究動機與目的 2](#_Toc375212612)

[貳、 專題架構 3](#_Toc375212613)

[參、 相關背景與技術 4](#_Toc375212614)

[肆、 EzWebGame Library 7](#_Toc375212615)

[伍、 Case Study : EzWebChecker 9](#_Toc375212616)

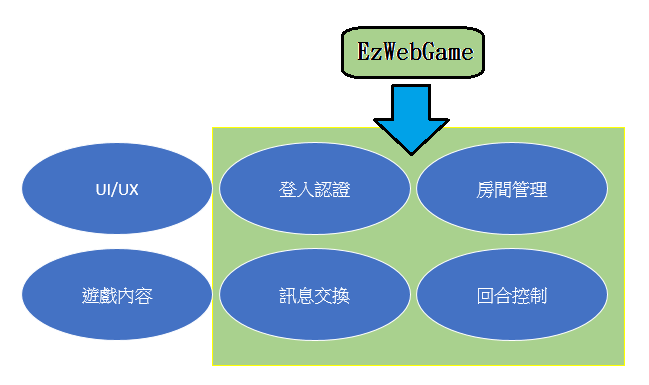
[陸、 結論 10](#_Toc375212617)

[柒、 參考文獻 11](#_Toc375212618)

1. 研究動機與目的

隨著Web Game的流行以及HTML5 2D Canvas技術的崛起，開發者對於快速開發網頁遊戲的需求趨於明顯。我們想要建構一個平台，讓開發者能夠很輕易並快速的開發網頁遊戲。於是EzWebGame因此誕生了。

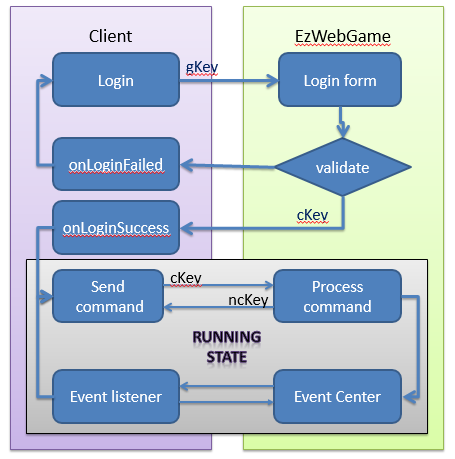
我們想建構一個對開發者友善的平台。將繁複的Message溝通、回合控制、遊戲房間控制以及認證授權全部包裝起來。並釋出前端EzWebGame Library讓開發者能以用Event Trigger的模式來開發遊戲。



1. 專題架構

EzWebGame架構大致上可以分成登入認證階段以及執行階段。登入認證階段開頭由Client呼叫login library傳遞gKey並另開新視窗到EzWebGame登入頁面。EzWebGame會根據gKey辨識要登入哪一個遊戲，而gKey由開發者向EzWebGame註冊取得。表單驗證失敗之後會觸發onLoginFailed事件，驗證成功則觸發onLoginSucess事件並回傳communication Key。取得communication Key後進入實作遊戲內容的執行階段。

EzWebGame把遊戲切成遊戲房間管理、回合控制、信息交換等三個部分。開發者可以用EzWebGame Library的method以及event來建構、串接這三部分。傳遞指令會用communication Key來做驗證，communication Key效用只有一次性並且在Server處理完要求後會發送新的communication Key。以此方式來確認指令的來源合法性。



(圖一) EzWebGame主要架構

1. 相關背景與技術

* Comet programming

Comet是一個Web應用程序模型，其中一個長期持有的HTTP請求允許Web服務器將數據推送到瀏覽器，沒有明確要求它的瀏覽器。Comet是一個總稱，涵蓋多種技術可以實現這互動。所有這些方法依賴於瀏覽器中，如JavaScript，而不是在非默認的插件默認包含的功能。彗星方法不同於幅的原始模型，在該瀏覽器請求一次一個完整的網頁。

Comet is a web application model in which a long-held HTTP request allows a web server to push data to a browser, without the browser explicitly requesting it.[1][2] Comet is an umbrella term, encompassing multiple techniques for achieving this interaction. All these methods rely on features included by default in browsers, such as JavaScript, rather than on non-default plugins. The Comet approach differs from the original model of the web, in which a browser requests a complete web page at a time

* KineticJS

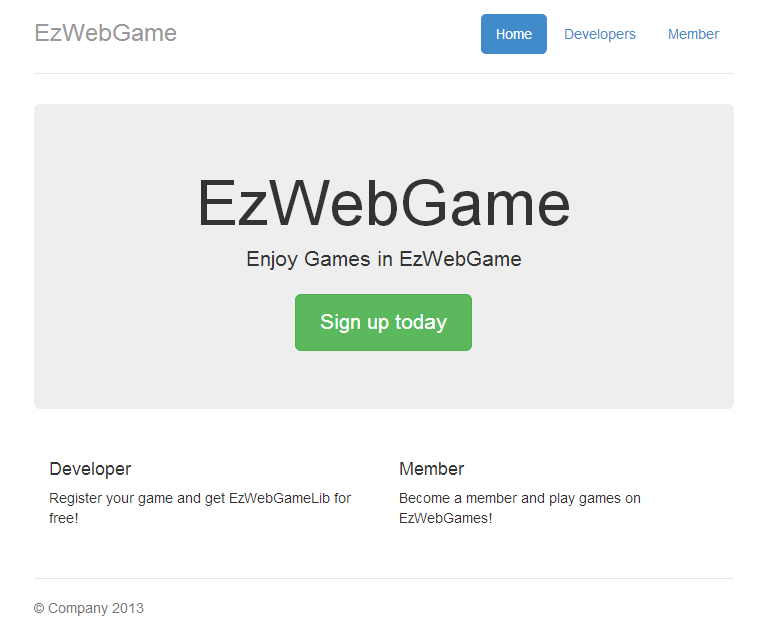
KineticJS是一個HTML5 Canvas的JavaScript框架，它針對桌面以及行動等應用程式提供高效能的動畫、變形、節點網路、圖層、過濾器、快取、事件處理等功能。

* CodeIgniter + Bootstrap

CodeIgniter 是一套給 PHP 網站開發者使用的超輕量級MVC應用程序開發框架和工具包。她提供一套豐富的標準庫以及簡單的介面和邏輯結構，其目的是使開發人員更快速地進行項目開發。

Bootstrap是一個免費的工具集，用於開發網站。其中包含HTML、以CSS為基礎的設計樣板、表格、按鈕、導覽列等介面組件、以及選用性的JavaScript擴充功能。

EzWebGame使用CodeIgniter以及Bootstrap來實作server side以及Web概念網站，概念網站用來提供開發者註冊遊戲並取得EzWebGame Lib以及提供玩家註冊帳號。



1. **EzWebGame Library**

* API
* **登入登出介面**

|  |  |  |
| --- | --- | --- |
| **Function name** | **Parameter** | **Description** |
| login | none | 登入 |
| logout | none | 登出 |

* **遊戲大廳介面**

|  |  |  |
| --- | --- | --- |
| **Function name** | **Parameter** | **Description** |
| listRoomInfos | none | 得到房間列表 |
| createGameRoom | 1. Room Title 2. Players Min Number 3. Players Max Number | 創立並進入房間 |
| joinGameRoom | 1. Room ID | 加入房間 |

* **房間介面**

|  |  |  |
| --- | --- | --- |
| **Function name** | **Parameter** | **Description** |
| startGameRoom | none | 開始遊戲 |
| leaveGameRoom | none | 離開房間 |

* **遊戲執行中介面**

|  |  |  |
| --- | --- | --- |
| **Function name** | **Parameter** | **Description** |
| doStep | 1. Instruction | 傳送指令 |
| finishStep | none | 結束此回合 |
| finishGame | none | 告知達成獲勝條件 |
| replyCheck | 1. bool | 確認其他玩家獲勝與否 |

* **其他介面**

|  |  |  |
| --- | --- | --- |
| **Function name** | **Parameter** | **Description** |
| isTurnSelf | none | 查詢是否輪到自己 |
| getNowTurnUserOrder | none | 查詢現在玩家的順序 |
| getUserTurnOrder | 1. (Player ID) | 查詢自己的順序，或指定玩家的順序 |
| getUserId | none | 查詢自己的ID |

* Event
* **認證事件**

|  |  |  |
| --- | --- | --- |
| **Event name** | **Parameter** | **Description** |
| onLoginSuccess | None | 登入成功 |
| onLoginFail | 1. Error Message | 登入失敗 |
| onLogout | none | 登出成功 |

* **大廳中事件**

|  |  |  |
| --- | --- | --- |
| **Event name** | **Parameter** | **Description** |
| onListRoomDone | 1. Room Infos | 收到大廳房間訊息 |
| onRoomCreated | 1. Room Info | 收到房間已創立訊息 |

* **房間中事件**

|  |  |  |
| --- | --- | --- |
| **Event name** | **Parameter** | **Description** |
| onRoomLeaved | none | 離開房間成功 |
| onRoomChanged | 1. Room Info | 房間狀態改變 |
| onRoomJoined | 1. Room Info | 房間加入成功 |
| onRoomStarted | 1. Room Info | 遊戲開始 |

* **遊戲中事件**

|  |  |  |
| --- | --- | --- |
| **Event name** | **Parameter** | **Description** |
| onReceiveStep | 1. Instruction | 收到指令訊息 |
| onChangeTrun | 1. playerInfo | 輪到下一位玩家 |
| onCheckWin | 1. Player ID | 某玩家說他獲勝，要確認他是否真的獲勝，回覆給伺服器 |
| onAccomplishGame | 1. playerInfo | 玩家達成遊戲條件 |
| onGameFinish | 1. rank | 遊戲結束 |

***Player Info****【玩家資訊】: 包含 userId, userName。*

***Players Info****【玩家資訊】: Player Info Array*

***Room****【房間資訊】: 包含 max, min, title*

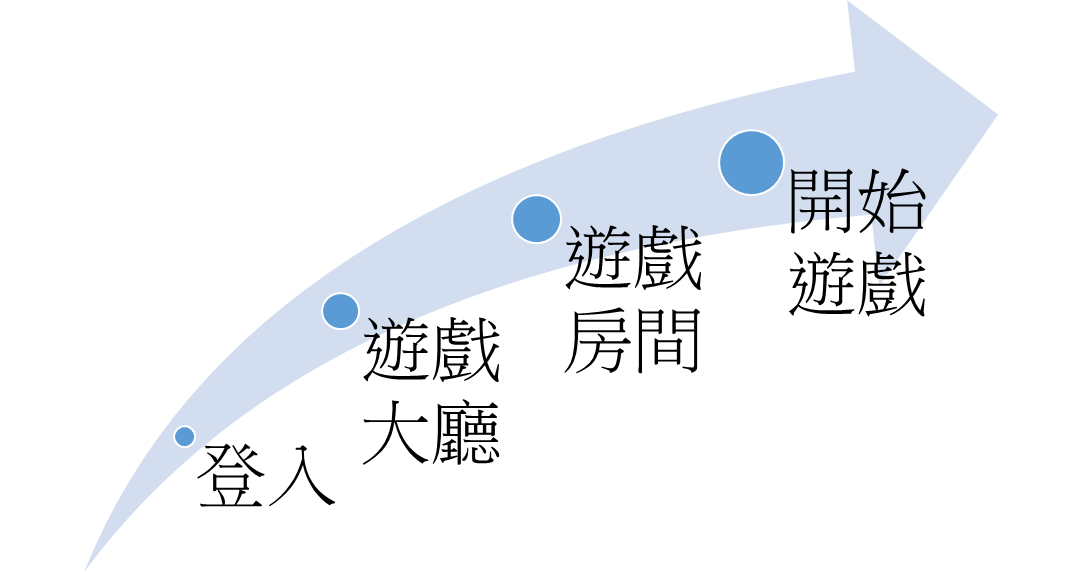
***Room Info****【房間資訊】: 包含 Room, Players Info*

***Room Infos****【房間資訊】: Room**Info Array*

1. Case Study : EzWebChecker

* 摘要

EzWebChecker是一款簡易多人線上跳棋遊戲，並搭配EzWebGame Framework 來加速遊戲的開發效率，使得開發遊戲中，可省去最麻煩的認證部分、回合控制器、房間系統與線上玩家互傳訊息功能。



* 設計

遊戲初始階段，透過呼叫EzWebGame提供的API來實現登入認證、進入遊戲大廳、進入遊戲房間，並實做認證、大廳、房間事件來使遊戲能做相對的動作，且搭配KineticJS Layer的切換來完成整個遊戲。

遊戲進行階段，呼叫doStep來對所有房間內的玩家傳送移動指令字串，而本遊戲的指令設計以Json為格式，方便OnRecieveStep事件解析此字串能代表的動作與參數，再畫出該指令動畫或更改遊戲參數。finishStep及onChangeTurn事件，使得遊戲可以得知目前輪到的玩家已經結束該回合並得知下一位輪到誰。最後若玩家抵達終點，呼叫finishGame來通知EzWebGame自己已獲勝，而由系統指派一位玩家，來決定是否該玩家已達獲勝條件，若獲勝則由系統的回合控制器中排除，而onAccomplishGame事件來處理有玩家獲勝該執行之動作，onGameFinish事件告知所有玩家遊戲排名。



1. 結論

往往遊戲開發者設計遊戲時，能得知目前發生甚麼事情，而只專注於事情發生後，程式需要做的動作與送出的指令、畫面如何安排與配置，許多煩瑣的登入流程、大廳系統、房間系統都不用操心且還能輕易得知這些資訊

EzWebGame成功的提供一個方便開發前端網頁遊戲的架框。不過讓開發者能減少在後端心力的同時，也意味著大部分程式碼會暴露在前端上。該如何確保安全性是一個值得後續研究的議題。

1. 參考文獻

[1] Comet Programing  
www.en.wikipedia.org/wiki/Comet\_(programming).

[2] CodeIgniter  
http://www.codeigniter.org.tw

[3] Bootstrap  
http://getbootstrap.com/

[4] HTML5 Canvas  
http://www.w3schools.com/tags/ref\_can vas.asp

[5] HTML5 Server Sent Events  
http://www.w3schools.com/html/html5\_serversentevents.asp