

遊戲自動化功能性測試工具設計與調查

Automatic Functional Testing Tool-Design and Survey for Games

陳偉凱、劉建宏、陳炳宏、舒一斌

國立臺北科技大學資訊工程系

Woei-Kae Chen, Chien-Hung Liu, Ping-Hung Chen, Yi-Bin Shu

Department of Computer Science and Information Engineering,

National Taipei University of Technology

Email : {wkchen, cliu, t100599001} @ntut.edu.tw, yibinshu@gmail.com

摘要

隨著電腦遊戲的快速成長，其測試亦日趨重要，有別於一般軟體已有相當多自動化功能性測試工具可供使用，目前電腦遊戲似乎尚缺乏適當的自動化功能性測試工具，針對這個問題，徐嘉陞提出一個支援自動化功能性測試的HTML5遊戲開發框架。本論文以此框架為基礎並強化其功能，包含(1)在錄影模式中可重播腳本，並可隨時中斷播放再繼續錄製腳本；(2)提供使用者介面讓測試人員能取得遊戲變數，自動加入驗證項目至腳本中；(3)加上計算測試腳本的Code Coverage功能；(4)支援錄製滑鼠移動事件。本論文徵求開發人員進行個案研究，在進行遊戲開發時，實際使用此框架提供之測試工具，以評估此工具的實用效果。此外，本論文亦針對遊戲業界進行調查，探討業界執行測試的方式、以及對功能性測試的做法，並詢問本論文所提出之工具，是否對開發遊戲時的功能性測試有所助益、以及可能帶來的效益為何。調查結果顯示，功能性測試為所有測試項目中最耗費成本者，而且業界目前普遍未導入自動化功能性測試，受訪者認為本論文所提出之工具可以確實降低測試的成本。

關鍵字：遊戲測試、自動化測試、功能性測試、業界訪談

一、緒論

目前軟體測試越來越蓬勃發展，而為了減少測試成本，已有相當多自動化功能性測試工具被使用，例如 Selenium [1]、Robot Framework [2]、Calabash [3]等工具。但目前遊戲業界似乎尚缺乏相關的自動化功能性測試工具，探討其原因與解決方法成為本論文的主要目標。

為了解遊戲業界缺乏自動化功能性測試工具的原因，本論文對遊戲業界進行訪談，了解業界進行遊戲功能性測試的方式，以及是否有使用相關的測試工具；若沒有，則深入了解未使用工具的原因，並詢問功能性測試與各項測試的成本比例；了解功能性測試所佔的測試比例，以及業界希望有何種自動化測試工具來協助遊戲開發，藉此來釐清業界目前對於遊戲測試的需求為何。

對於遊戲業界的需求，徐嘉陞 [4] 於 2015 年

提出支援遊戲自動化測試之 HTML5[12]遊戲框架，該已提出錄影、播放功能來進行自動化功能性測試，本論文對該遊戲框架進行改善，加強錄影播放功能，使測試人員更方便錄製測試腳本，並提供更多測試輔助功能，例如計算測試腳本的Code Coverage，以及列出遊戲變數供加入驗證等，提供測試者更進一步了解與改善遊戲測試腳本的品質。

此外，本論文進行個案研究，透過徵求開發人員，實際使用自動化功能性測試框架來進行遊戲開發與測試，並對遊戲業界進行訪談，來了解此自動化功能性測試工具是否可在業界中實際運作，並詢問可能帶來的效益。本論文希望能達到下列目標

- 強化錄影、播放工具功能
 - 在錄影模式下可即時重新錄製與重播腳本
 - 提供使用者介面，讓測試人員能取得遊戲物件屬性，作為測試驗證之用
 - 加入計算Code Coverage功能，以確認測試案例的完整性與品質
 - 增加自動上傳測試紀錄，以方便統計測試紀錄
 - 支援滑鼠移動事件的錄製
- 對遊戲業界進行訪談以了解目前的測試方法
 - 了解目前遊戲開發或測試人員是否導入自動化功能性測試
 - 詢問遊戲業界是否願意採用本論文提出之自動化錄影、播放工具
- 實際操作錄影播放工具後的評估結果
 - 探討錄影播放工具是否能夠降低測試時間

調查結果顯示，在遊戲業界中，功能性測試為花費成本最高的測試項目，而且業界認為本論文提出之錄影、播放功能可以減少功能性測試的成本；同時，個案研究的結果也說明本遊戲框架可以減少測試成本，並有效協助發現Regression Error，有助於盡早修正錯誤。

二、相關研究

Ramadan 與 Hendradjaya [5] 提出遊戲是一種提供娛樂性質的軟體，有別於一般的軟體，遊戲品質是難以被進行測試的，因此需要一個對於遊戲質量進行驗證的測試項目。該論文主要以使用者經驗、功能性、可維護性以及可移植性等四個構面，提出八個對於遊戲質量的測試項目，使遊戲開發可

以針對所提出的八個測試項目進行測試，對於遊戲的質量進行評估。

Cho 等人 [6] 提出了一個針對線上遊戲的 scenario-based 黑箱測試，在線上遊戲進行模擬大量虛擬玩家的黑箱 scenario-based 測試。透過分析封包資料，整理出遊戲中的遊戲地圖和玩家進行人物操作的相關封包資料。讓測試人員透過傳遞分析出的遊戲地圖資訊並針對遊戲中的人物操作來建立測試的 scenario。

Zhao 等人 [7] 提出一改善 WAP Mobile Game 的 Message Sequence Chart，於 Message Sequence Chart 架構中加上 Servlet，透過 Servlet，將遊戲資料傳給 Servlet 進行運算，使 Mobile device 本身不需要負責運算，以減少手機的硬體負擔，再將 Message Sequence Chart 轉換成 TTCN-3 的 graphic presentation format(GFT)，並透過 GFT 產生遊戲的測試案例。

徐嘉陞 [4] 提出一支援遊戲自動化測試之 HTML5 遊戲框架，該遊戲框架已具備基本的遊戲錄影播放功能，可錄製測試人員實際進行遊戲的操作過程，並轉為測試腳本，再以人工方式在測試腳本中加入驗證項目，並播放測試腳本，查看驗證項目是否通過。該論文並藉由進行實驗，來對於人工手動測試、直接撰寫腳本與錄製腳本等三種方式，比較進行測試所需的成本。

三、框架設計與實作

3.1. 系統需求

徐嘉陞提出之遊戲框架，已具備基本的錄影播放功能，然而該論文尚有下列問題未予克服，例如：(1) 無法錄製滑鼠移動事件；(2) 在錄影模式中，若錄製發生失誤時，需要重新錄製腳本；(3) 錄製腳本結束後，需要人工閱讀腳本加入驗證項目等。為改善上述問題，本論文主要針對錄影模式和播放模式，提出下列改善之目標並增加新功能。

● 錄影模式

在錄影模式中，為了讓測試人員能夠更加方便的錄製測試腳本，提出下列改善目標。

■ 可即時重播與接續錄製腳本

原先之遊戲錄影框架中，當測試人員進行遊戲錄製時，是採人工進行遊戲，可能會因為遊戲操作失誤，導致未能完成原先預計錄製的測試案例腳本，此時只能重新錄製該測試案例腳本，如此一來將會浪費許多時間在錄製重覆的腳本上。因此希望在錄製發生失誤時，可重播當前已錄製之腳本，到錯誤發生前予以中斷播放，並繼續錄製後續的腳本，以節省重複錄製相同腳本內容的時間。

■ 提供使用者介面，讓測試人員能取得遊戲物件屬性，作為測試驗證之用

原先之遊戲錄影播放框架中，當測試人員需要在腳本中加入測試驗證時，只能在測試腳本錄製完成後，查看腳本內容並手動加入要驗證的變數和預期數值；亦可在錄製時加入註釋，以便於找出原先

欲加入驗證的地方，再以人工加上驗證。本論文希望在錄影模式中，測試人員可以隨時查看當前遊戲中的所有變數，並提供使用者介面，讓測試人員能夠在錄製腳本同時加上所要驗證的內容，減少錄製腳本的時間。

■ 支援滑鼠移動事件的錄製

原先之遊戲錄影播放框架中，只支援鍵盤事件和滑鼠點擊事件，為了支援更多類型的遊戲，加入了錄製滑鼠移動事件的功能，讓錄製的腳本內容更完善。

● 播放模式

在播放模式中，為了讓測試人員可以更加瞭解當前測試情況，加上計算 Code Coverage 和自動上傳測試執行紀錄等功能，讓測試執行更完善。

■ 加入計算 Code Coverage 功能，以確認測試案例的完整性與品質

除了知道測試的通過與否，為了讓測試人員能夠更了解目前測試腳本內容的完整性，希望測試腳本播放結束後，能夠統計出目前所有測試腳本的 Code Coverage 和涵蓋範圍，讓測試人員可以即時知道目前還有哪些功能尚未經過測試。

■ 增加自動上傳測試紀錄，以方便統計測試紀錄

在播放模式中，當所有測試內容播放完畢後，將所有測試結果上傳至伺服器，包含測試腳本名稱、測試執行時間、測試通過與否等資訊，讓測試人員可以查看所有測試腳本的執行紀錄，作為日後設計新的測試案例參考。

3.2. 系統架構與設計

依據前一個小節的分析，將徐嘉陞提出的遊戲錄影播放框架進行改善，圖 1 為目前的系統類別圖。

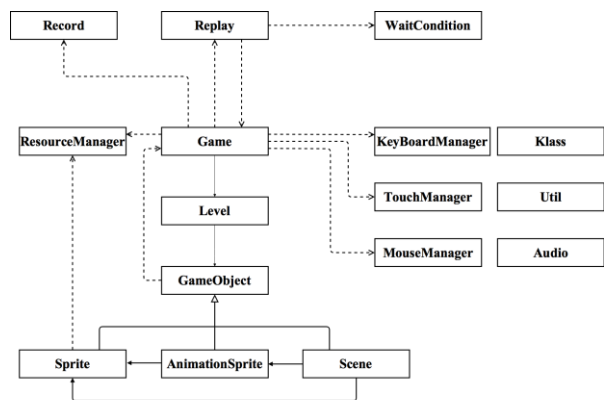


圖 1 支援自動化測試之遊戲框架類別圖

以下說明目前錄影播放的系統執行方式：首先，Game 為遊戲框架中的主要 Class，而 Replay 為播放模式時負責模擬遊戲中的指令，如鍵盤指令、滑鼠指令以及等待指令，將指令傳給 Game 來達到由腳本播放遊戲內容的功能。而 Record 則為錄影模式，當遊戲開啟錄影模式時，會產生 Record

這個 Class, Record 則會將測試人員輸入的指令自動記錄下來, 並轉換為測試腳本。

錄影模式中, 主要是讓 Record 接收遊戲中所有指令 (遊戲本身的生命流程並沒有改變), 將其轉換為腳本內容, 並顯示在畫面上, 由於遊戲的 Game 可由 Record 操作, 因此在錄影模式中可以隨時暫停遊戲。

播放模式中, 會先載入遊戲的測試腳本, 將腳本上的遊戲指令傳給 Replay, 讓 Replay 逐步執行指令, 達到播放錄製內容的功能, 並在每個腳本執行完時, 呼叫重新執行遊戲, 來讓播放模式可以播放多個測試腳本。

3.3. 框架使用之技術與方法

依據 3.1 節所提出的改善目標分為以下幾點。

- 改善錄影功能

在原先的播放模式中, 若腳本在錄製時發生人為失誤, 必須重新錄製整個測試案例, 導致測試人員需要耗費許多時間在重複錄製相同的腳本上。本論文增加在錄影模式中可以直接重播目前已錄製之腳本內容的功能 (如圖 2 所示), 使得測試人員可以直接重播腳本中錄製完成的部分, 完整的錄影模式狀態圖如圖 3 所示。



圖 2 腳本重播按鈕

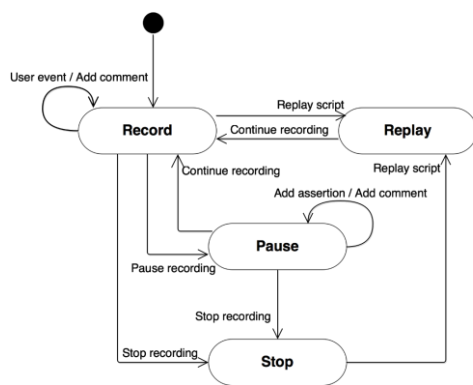


圖 3 錄影模式狀態圖

當錄製腳本時, 點擊錄影模式的重播按鈕, 可以將目前已錄製的腳本指令, 傳給 Replay, 透過 Replay 將已錄製的指令轉換成鍵盤或滑鼠指令傳

給 Game, 來播放目前已錄製的腳本內容, 而播放內容亦逐步顯示在左方的腳本視窗。當點擊繼續錄製腳本按鈕時, 即中斷當前播放的腳本, 回到錄影模式, 讓測試人員可以繼續錄製後續的腳本。

- 遊戲物件屬性 Assertion 支援

為了讓測試人員能夠在腳本中快速加入 Assertion, 在錄影模式中, 透過 javascript 從 Game 中取得當前遊戲 Level 中的所有變數, 並建立成樹狀選單顯示在畫面上 (如圖 4 所示), 讓測試人員可以點選清單中的變數; 若有可以進行驗證的變數, 則可以點選 Assert 按鈕, 自動將選擇之變數轉換成測試腳本, 以加快測試人員錄製腳本的速度。

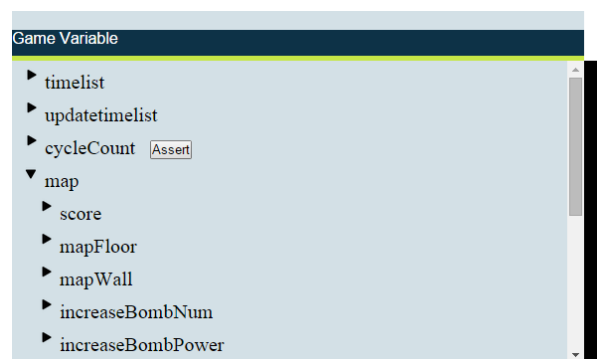


圖 4 遊戲功能變數清單

- 計算 Code Coverage 功能

在播放模式中, 為了讓測試人員更加了解目前完成的測試情形, 加入計算 Code Coverage 的功能, 原本播放模式的驗證是基於 Qunit [8] (一 javascript unit test framework) 來進行測試, 現在則加上 Blanket.js [9]。Blanket 是一支援 Qunit、Mocha [10] 跟 Jasmine [11] 等 javascript 測試框架的計算 Code Coverage 的 library。在遊戲執行時, 動態載入遊戲中所有的 javascript 檔案, 並依 Qunit 的測試執行內容, 來計算出目前所有錄製腳本的 Code Coverage, 當腳本播放結束後, 將 Code Coverage 結果顯示在畫面上, 讓測試人員可以查看目前測試結果, 並可以找出腳本中尚未測試的功能, 方便測試人員設計新的測試腳本。

由於遊戲框架中的遊戲物件 js 檔為動態載入, 並且在計算 Code Coverage 時需要計算的是遊戲的 production code, 需要和遊戲框架中的程式碼加以區隔。因此, 在遊戲框架中, 只有載入遊戲 production code 時, 才會呼叫 Blanket 去動態加入遊戲的 production code 的來源, 讓 Blanket 在執行測試時, 可以計算出各個遊戲物件的 Code Coverage, 圖 5 所示為播放模式的活動圖。

- 自動記錄測試執行資訊

在播放模式中, 為了記錄所有的測試執行資訊, 因此在 Qunit 框架中, 加上了上傳測試結果的功能, 並且先建立一個 websocket 的 server。當測試人員進入播放模式執行測試時, 會發送請求向 websocket server 建立連線, 並傳送測試結果到伺

伺服器上，測試結果包含：執行測試端 IP、測試端專案名稱、測試日期、測試腳本名稱、執行時間以及測試通過與否，將所有資訊轉為報表，讓日後測試人員方便隨時查看測試紀錄。

測試記錄報表格式為 csv 檔，檔案詳細內容如圖 6 所示，各欄位由左至右依序為執行測試端 IP、專案名稱、測試日期、測試腳本名稱、測試執行時間(ms)、測試通過與否等資訊。

● 支援滑鼠移動事件錄製

有些遊戲在遊戲進行時需要詳細的滑鼠移動軌跡(例如電流急急棒)，為了支援這些類型的遊戲，在錄影模式中增加了錄製滑鼠移動事件的功能，讓測試人員在進行遊戲時的滑鼠移動事件，可以被錄製下來並轉換為腳本指令並且在播放時可以重現完整的滑鼠移動軌跡。測試人員可以根據遊戲類型以及測試需求，在設定檔中決定(修改)是否開啟錄製滑鼠移動功能，以避免測試腳本中記錄過多不必要的資訊，如圖 7 所示。

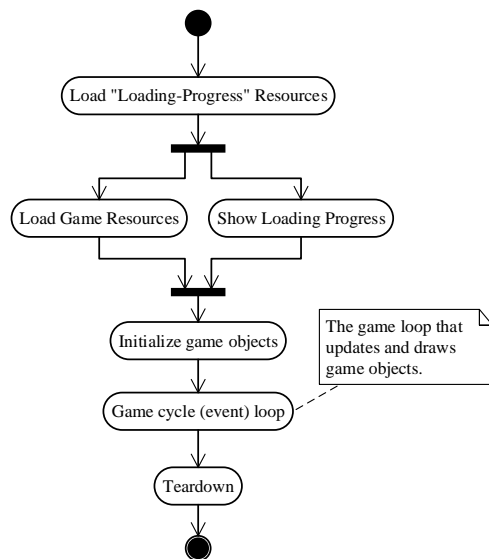


圖 5 播放模式活動圖

	A	B	C	D
1	140.124.147.72	g34	2015/5/5 12:03	space attack
2	140.124.147.72	g34	2015/5/5 12:04	space attack
3	140.124.147.72	g34	2015/5/5 12:05	space attack
4	140.124.147.72	g34	2015/5/5 12:06	space attack
5	140.124.184.106	g34	2015/6/5 17:02	Test call army need to wait
6	140.124.184.106	g34	2015/6/5 17:02	Test call next army need to wait

圖 6 測試記錄檔格式

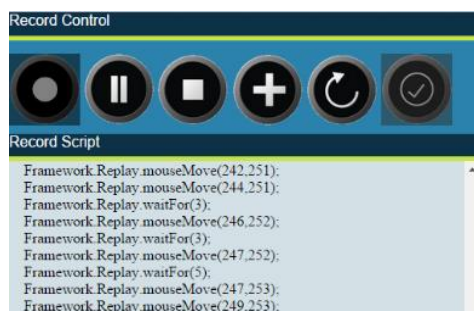


圖 7 錄製滑鼠移動

四、業界訪談

本節將介紹本論文訪談台灣遊戲業界，對於遊戲測試的進行方式，了解目前遊戲功能性測試是否有使用自動化工具、功能性測試所佔的測試成本比例，以及對於本論文中提出的自動化功能性測試工具有什麼看法與意見，並詢問是否有使用意願。

4.1. 訪談目的與問題設計

為了解目前遊戲業界的測試進行方式，對於功能性測試是否有採用自動化測試，以及對於本論文提出的自動化功能性測試工具是否會想要使用，本論文設計了表 1 的題目，來對業界進行訪談。

表 1 訪談題目總表

編號	訪談問題
Q1	受訪人資料
Q2	受訪人公司資料
Q3	請問貴公司在進行遊戲開發的過程中有做過下列那些測試？
Q4	描述貴公司目前進行測試的方式
Q5	針對貴公司開發之遊戲的功能性測試是否有採用自動化或半自動化？
Q6	依上題，若貴公司沒有進行自動化或半自動化功能性測試，為何沒有採用相關之自動化功能性測試工具，以及是否有雇用人力透過人工方式進行測試？
Q7	貴公司在進行功能性測試時，對於功能性測試通過與否的準則如何定義？
Q8	貴公司在遊戲專案開發的生命週期（Life Cycle）中，對於開發之不同階段時分別採用何種測試方式？
Q9	貴公司在遊戲專案開發時之各項測試與開發的成本比例為何？
Q10	現場展示本論文之自動化功能性測試工具，詢問貴公司之看法與意見，以及是否會想要使用？
Q11	對於貴公司目前之測試方式是否滿意？對於測試的方式是否還有可改善的空間或目標？
Q12	貴公司希望有何種自動化測試工具，對於遊戲開發可以帶來更大的幫助？

4.2. 訪談結果

本業界訪談共有七位受訪者接受訪談，表 2 至表 5 分別為各受訪結果的統計資料。

表 2 受訪者資料總表

受訪者	職稱	總年資	身份	測試經驗
P1	程式長	8 年	開發人員	有
P2	程式課長	8 年	開發人員	有
P3	技術總監	5 年	開發人員	有
P4	軟體品質工程師	1.5 年	測試人員	有
P5	遊戲製作人	3 年	製作人	有
P6	開發主任	7 年	開發人員	有
P7	程式工程師	4 年	開發人員	有

表 3 受訪公司資料總表

公司名稱	公司規模	成立時間	主要開發之遊戲類型	開發遊戲數量(年)
C1	7 人	2014 年	休閒類遊戲	1
C2	400 人	1995 年	單機 RPG 類型	4-6
C3	100 人	2007 年	手機遊戲	3
C4	1000 人	1995 年	PC 線上遊戲	2-3
C5	800 人	1989 年	博奕類遊戲	4-6
C6	800 人	1989 年	單機 RPG、MMORPG	4-6
C7	400 人	1995 年	博奕類遊戲	4-6

表 4 各項測試成本比例總表(百分比)

測試項目	P1	P2	P3	P4	P5	P6	P7	Avg
Usability	0	0	5	7	14	6	0	4.6
Functional	20	25	15	22	14	30	20	20.9
Regression	0	0	5	16	0	6	0	3.9
Performance	10	10	5	7	8	9	10	8.4
Security	0	0	5	0	0	9	0	2.0
Unit	10	10	15	14	0	0	0	7.0
Fun	30	25	15	7	31	16	0	17.7
Internally Complete	0	0	2.5	1.5	21	6	0	4.4
Maintainability	10	10	5	3	1.8	4	0	4.8
Balance	10	20	20	5	9	6	70	20.0
Service Compatibility	0	0	5	8.5	1.2	4	0	2.7
Portability	10	0	2.5	9	0	4	0	3.6

表 5 是否願意使用本論文之工具總表

	願意使用	預計可降低功能性測試比例
P1	是	40.0%
P2	是	40.0%
P3	是	66.6%
P4	是	30.0%
P5	是	43.0%
P6	是	50.0%
P7	是	25.0%
平均		42.1%

依據訪談結果，來了解目前遊戲業界的需求，以及對於本論文提出的自動化功能性測試工具的看法，本論文統整出以下結論：

- 遊戲的功能性測試並未進行自動化測試

依據訪談結果，目前業界在開發遊戲過程之所以沒有自動化測試，主要是缺少相關的自動化測試工具，若自行開發測試工具又太耗費成本，因此仍以人工方式進行功能性測試。

- 遊戲的功能性測試為所有測試項目中最耗費成本者

依據訪談結果，由於目前遊戲的功能性測試是採人工方式進行測試，測試人員需要重複進行相同

的測試項目來驗證遊戲功能的正確性，因此會花費相當高的成本在進行功能性測試。訪談統計結果顯示，功能性測試平均占了總測試成本 20.9%，為所有測試項目中最高者。

- 本論文之自動化功能性測試工具可以應用在開發後期，或整合在開發流程之持續整合測試中
受訪者認為在遊戲開發前期，測試腳本可重複利用性不高，因此大部分的受訪者表示會在開發後期使用。部分受訪者則認為，可以整合在開發流程之持續整合測試中。

- 使用本論文之自動化功能性測試工具可以降低人工測試的成本

受訪者願意使用本論文之自動化功能性測試工具，原因為：(1)可以有效降低人工測試成本，平均可減少 42.1% 的功能性測試成本；(2)遊戲在進行人工測試時，經常出現當測試人員發現遊戲錯誤時，由於是人工操作，回報開發人員時卻無法重現該錯誤。百分之百重現測試執行結果，可以有效地協助開發人員進行遊戲錯誤修正；(3)可以及早發現 Regression error。

五、個案研究

為了解本論文中提出的自動化功能性測試工具是否能夠幫助遊戲開發，減少人工測試的成本，本論文徵求遊戲開發人員，實際使用本論文提出的自動化功能性測試框架，並對遊戲開發人員進行訪談，了解對於自動化功能性測試工具有什麼看法與意見。參與個案研究的對象為臺北科技大學資訊工程系 104 學年度第二學期物件導向程式設計實習課程的修課學生，該課程內容主要為讓學生在學期中，以分組方式進行遊戲開發。我們徵求其中三組學生，實際在開發過程中使用本論文的自動化功能性測試工具，並了解他們的使用情況。

透過自動化功能性測試工具的自動上傳測試結果功能，蒐集了各組的測試執行結果，並分析其工具使用情況，表 6 為開發人員資料，表 7 為統計結果；結果顯示，錄製腳本數和執行次數與可減少之測試成本有關，重複執行越多次，可減少的測試成本比例越高。

表 6 開發人員資料

組別	組員 人數	遊戲名稱
組別 1	1 人	東方永夜抄 (Imperishable Night)
組別 2	2 人	刀塔傳奇 (Dota Legend)
組別 3	2 人	幼蟲英雄 (Larva Hero)

表 7 腳本執行統計結果

組別	錄製腳本總數	總播放腳本次數
1	10	75 次
2	11	70 次
3	16	103 次

我們訪談實際使用自動化功能性測試框架的開發人員，了解工具實際使用情形，並詢問本框架是否能夠帶來幫助，以下為訪談統整結果。根據表 8-10 的訪談統整結果顯示，本論文之自動化功能性測試工具，平均可減少開發人員約 15.7% 的測試時間；而且開發人員認為，此工具有助於開發過程中及早發現 regression error 以進行修正；最後，實際使用過的開發人員皆表示願意再次使用本工具。

表 8 是否可降低測試時間

組別	是否可降低測試時間	減少比例
1	是	12%
2	是	10%
3	是	25%
平均		15.7%

表 9 可否及早發現 regression error

組別	可否及早發現 regression error	發現 次數	提早多久 (天)
1	是	1	14
2	是	3	5
3	是	4	9
平均		2.7	9.3

表 10 是否願意再使用框架及原因

組別	是否 願意	原因
1	是	可以提早發現遊戲的錯誤，提早修正
2	是	節省測試時間，記錄所有要測試的項目
3	是	可以記錄要測試的項目，可以避免人工測試時忘記測試

六、結論

本論文改善徐嘉陞 [4] 提出的支援遊戲測試之遊戲框架，在錄影模式中，支援滑鼠移動事件的錄製，使工具可支援更多類型的遊戲；同時增加重播和繼續錄影功能，使測試人員可以重播已錄製成功的腳本，並可即時中斷播放再繼續錄製，減少錄製相同腳本的時間。在播放模式中，加入了計算 Code Coverage 功能，使測試人員可以查看測試案例中未涵蓋的功能，來設計新的測試腳本。同時也增加自動上傳測試紀錄功能，在測試播放完畢時，會自動上傳測試執行結果，讓測試人員可以查看測

試執行紀錄，掌握遊戲開發與測試的狀況。

本論文並對遊戲業界進行訪談，以了解目前業界的測試過程，根據訪談結果，目前遊戲業界對於功能性測試並未導入自動化測試，主要是由於缺少相關的自動化測試工具，而且自行開發測試工具太耗費成本。由於目前功能性測試主要是以人工方式進行，導致功能性測試在所有測試項目中成為最耗費成本者。訪談結果也顯示，業界願意採用本論文之自動化錄影播放工具，在遊戲開發前期，由於遊戲功能變動性高，可重複利用的測試腳本不多，因此主要在開發後期中使用。而錄影播放工具可以減少人工測試的成本，平均可減少 42% 的功能性測試比例。

在個案研究方面，本論文徵求開發人員，實際使用本框架進行遊戲開發和測試並對其進行訪談。根據開發人員實際使用與訪談結果，相較於人工測試，使用本論文之工具進行測試，平均可以減少 15.7% 的測試時間。參與研究的人員同時認為在修改程式時，自動化重新執行測試案例可有效幫助發現 Regression Error，有助於盡早修正錯誤。綜合業界訪談與個案研究結果顯示，可減少之測試成本與執行測試頻率和開發時間有關，本論文之框架可節省之測試成本為 15.7% 到 42% 之間。

致謝

本研究由科技部計畫 MOST 104-2221-E-027-008- 所補助，特此感謝。

參考文獻

- [1] Selenium, <http://www.seleniumhq.org/>, Jun. 1, 2015
- [2] Robot Framework, <http://robotframework.org/>, Jun. 1, 2015
- [3] Calabash, <http://calaba.sh/>, Jun. 1, 2015
- [4] 徐嘉陞，支援遊戲自動化測試之HTML5遊戲框架，碩士論文，國立台北科技大學資訊工程系，台北，2015
- [5] Rido Ramadan ; Bayu Hendradjaya , “ Development of Game Testing Method for Measuring Game Quality” Data and Software Engineering (ICODSE), 2014 International Conference, 2014, pp.1-6.
- [6] Chang-Sik Cho ; Kang-Min Sohn ; Chang-Jun Park ; Ji-Hoon Kang. “Online game testing using scenario-based control of massive virtual users” Advanced Communication Technology (ICACT), 2010 The 12th International Conference, 2010, pp. 1676 – 1680.
- [7] Zhao, Huiqun; Sun, Jing; Hu, Gongzhu. “Study of Methodology of Testing Mobile Games Based on TTCN-3” Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, SNPD '09. 10th ACIS International Conference, 2009, pp. 579 – 584.
- [8] Qunit, <https://qunitjs.com/>, Jun. 1, 2015
- [9] Blanket.js, <http://blanketjs.org/>, Jun. 1, 2015
- [10] Mocha, <http://mochajs.org/>, Jun. 1, 2015
- [11] Jasmine, <http://jasmine.github.io/>, Jun. 1, 2015
- [12] Peter Lubbers, Brian Albers, Frank Salim, Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development, New York, NY, USA:: Apress, 2011.