國立臺灣師範大學
資訊工程研究所碩士論文

指導教授：　陳伶志　博士

GWAP 系統之設計策略研究－以 ESP 為例

Playing GWAP with strategies - using ESP as an example

研究生：　朱文園　撰

中華民國　九十八　年　六　月

摘要

「人智運算」在電腦科學中是一種創新的概念。其概念目前是有許多事情是人能夠輕易解決而電腦做不到的事情。尤其在 2004 年 ESP 遊戲被推出之後開啟 Games With A Purpose (GWAP)這塊新的領域，其目的是在玩家進行遊戲的時候也在幫系統標記圖片。這些圖片的標記能夠用在影像辨識和影像搜尋上，而且圖片標記的正確性和圖片標記產生的速率都非常高。ESP 遊戲的成功使得人智運算和 GWAP 變成大家所矚目的焦點。GWAP 是讓玩家在進行遊戲時不知不覺地幫系統進行運算，並產生對系統有用的資料。因為玩家覺得遊戲很好玩，所以玩家會一直沉迷或是狂熱地進行遊戲，並且幫系統進行運算產生有用的資料。本論文是探討如何增加 GWAP 系統的效能，此處針對 ESP 遊戲進行分析，並定義測量系統效能的函數，且提出能夠增加系統效能的圖片選擇演算法 Optimal Puzzle Selection Algorithm (OPSA)。為了驗證 OPSA 確實能夠增加 ESP 遊戲的效能，實作實驗系統 ESP Lite 進行驗證。在二個月的實驗過程中，實際證實 OPSA 是能夠增加系統效能的圖片選擇演算法，並且發現 OPSA 的行為和玩家的行為有著相當大的關係。從此研究可以發現有策略地運行 ESP 遊戲系統就能增加系統的效能；同理其他的 GWAP 系統只要有策略地運行也同樣能增加系統的效能。

關鍵字：人智運算、Games With A Purpose、系統評估、人因工程、遊戲策略

ABSTRACT

"Human Computation" is an innovative concept in computer science. The idea is there is a lot of things that human can easy do that computers can not yet do. In 2004, the ESP game has been launched and it created an emerging field, Games With A Purpose (GWAP), in computer science. The objective of GWAP is creating difficult metadata when players are playing the game voluntarily. This thesis presents how to enhance GWAP systems. In this thesis, we use the ESP game as an example and propose a puzzle selection algorithm, Optimal Puzzle Selection Algorithm (OPSA), to enhance the ESP game system. For validating our proposed algorithm, OPSA, which actually enhances the ESP game system in real world, we implement a quasi ESP game, called ESP Lite. During a two-month experiment, we investigated the inner properties of the OPSA, and verified that the OPSA scheme achieves the best system gain for the ESP game system. The results of this thesis also confirm that GWAP systems are more efficient if they are designed and played with strategies.

Keywords: Human Computation, Games With A Purpose (GWAP), System Evaluation, Human Factor, Play Strategy

致謝

本篇論文得以完成是靠著身邊許多人的支持和幫助才得以完成。在兩年的碩士生涯中，我必須非常感謝父母親的支持，讓我能夠安心完成碩士學業。同時我也要非常感謝我的指導教授陳伶志博士，在學業上鼓勵我積極進取；在研究上教我以嚴謹的態度和方法做研究。恩師除了學業之外也教我許多做人的道理，並對我的表達給予大量的建議，讓我受益良多。我也必須感謝口試委員國立政治大學資訊科學系蔡子傑教授和國立臺灣師範大學資訊工程系柯佳伶教授給我許多寶貴的建議，讓本論文能夠更趨於完善。另外在實驗室中，總是有學長姐和同學們的鼓勵和支持讓我得以順利完成學業。最後必須再次感謝恩師的指導和大家的鼓勵與支持，才得以讓我順利完成碩士學業。

朱文園

於國立臺灣師範大學

中華民國九十八年六月

目錄

附表目錄

附圖目錄

第一章　緒論

第一節　研究動機與研究目的

人智運算(Human Computation) [11]在電腦科學中是一門新興的領域，有別於以往運算是交由 CPU 執行，人智運算是以人腦為主體進行運算。使用人腦當作運算主體乃由於在目前的電腦領域還無法做到強人工智慧，也就是電腦並不會推理和思考。所以遇到要推理和思考問題時，電腦並無法勝任此類的工作，只好將這類工作交給人腦進行處理。例如影像辨識和語音辨識這類問題，電腦不擅長這些類別的問題，但是人腦並沒有這個問題。雖然人腦處理問題的精確度比電腦處理的精確度還要高上許多，不過在運算速度上電腦遠遠超越人腦。所以如何讓人智運算更有效率成為一個值得思考的問題。

Games With A Purpose (GWAP) [39]是 Dr. Luis von Ahn 所提出，其目的是讓玩家在進行遊戲時，順便產生系統所需要的資料。玩家在進行遊戲的時候可能並不知道系統的目的，但是玩家會因為遊戲很好玩而一直進行遊戲，所以玩家在進行遊戲的同時為系統進行運算，進而產生有用的資料。換句話說 GWAP 是利用遊戲當作刺激的工具，刺激玩家在進行遊戲的同時也為系統產生有用的資料。這有點類似生物中互利共生的概念，所以 GWAP 是屬於人智運算中一種用來取得運算結果的方法。

在所有人智運算的系統中，在 2004 年所建立的 ESP 遊戲[41]是第一個成功且廣為人知的 GWAP 系統。ESP 遊戲以有趣的遊戲過程使得大家樂意投注時間和精力在遊戲上面。在遊戲過程中收集到的資料能夠對影像作註解，而這些被註解過的影像就能應用在像影像辨識和圖片搜尋或者是幫助盲胞了解圖片中的內容[18]等方面。而在 2006 年 Google 買下授權並且建立自己的 ESP 遊戲，稱為 Google Image Labeler [8]，Google 用此增加 Google 本身 Google Image Search [9]對影像的搜尋能力。

在目前人智運算的研究中，大多數的研究都集中在系統上的開發，而在理論分析上的研究比較少，因此希望能夠增加 GWAP 系統的效率，並且在本論文中以 ESP 遊戲為例，討論如何增加 ESP 遊戲系統的效能。在 ESP 遊戲中，被標記過圖片的數量和平均每張圖片標記的品質影響系統的效能，所以使用上述兩個因素做為測量系統效能的依據。為了增加 ESP 遊戲的效能，使用數學建立 ESP 遊戲的模型，並且對此數學模型進行分析，最後發展出透過系統有策略地送出圖片給遊戲者進行遊戲，藉以增加系統的效能。本篇論文提出圖片選擇演算法 Optimal Puzzle Selection Algorithm (OPSA)使得系統效能達到最佳。為了驗證 OPSA，所以設計一個實驗系統 ESP Lite 實際測試 OPSA 和其他兩種圖片選擇演算法作比較，分別是 Random Puzzle Selection Algorithm (RPSA)和 Fresh-first Puzzle Selection Algorithm (FPSA)。為了驗證 OPSA 確實能夠增加系統的效能，所以實作類似 ESP 遊戲的實驗系統 EPS Lite。ESP Lite 實作三種圖片選擇演算

法 OPSA、FPSA 和 RPSA，並且系統會紀錄所有遊戲過程。在為期二個月的實驗中，對 ESP Lite 的實驗結果進行分析，得到 OPSA 確實能夠增加 ESP 遊戲系統的效能，並且了解 OPSA 的行為和使用者行為是有相當大的關係。在本論文得知透過有策略地運行 EPS 遊戲系統以增加遊戲系統本身的效能是可行的，由此可知只要有策略地運行 GWAP 系統就能增加系統本身的效能。並且在最後提出未來可以加強和改進的地方。

第二節　論文架構

本篇論文會在第二章介紹相關的研究和文獻。第三章介紹 ESP 遊戲運作的方法，並且使用數學模型對 ESP 遊戲進行描述和分析。在第四章根據第三章分析的結果提出能夠增加 ESP 遊戲系統效能的圖片選擇演算法 OPSA，並且使用電腦對系統進行模擬。在第五章會介紹實驗系統 ESP Lite 的設計和架構。第六章是 ESP Lite 實際實驗的結果。最後在第七章得到結論並且討論未來該如何加強和改進。

第二章　相關研究

人智運算興起的原因在於光靠電腦的運算無法順利地解決許多事情，像是影像辨識和文字辨識；不過對人腦來說這些是能夠輕易解決的問題，因此有許多的人智運算系統被推出。從系統中收集到的資訊能夠當作機器學習中的訓練資料，或是經過整理之後給一般使用者使用。人智運算系統大致可以分成兩種類型，一類是較早被開發出來的人智運算系統，這是單純為了收集資料而設計出來的系統，所以沒有考慮以遊戲當作媒介。另一類是使用遊戲當作媒介的人智運算系統，此類型的全稱為 Games With A Purpose 簡稱 GWAP。除了有許多人智運算的系統被提出來之外，還有一些是分析人智運算系統的特性和架構等方面的研究。

第一節　非屬於 GWAP 的人智運算系統

因為電腦無法處理許多事情，所以就有許多各式各樣的人智運算系統被開發出來。這些系統可以依照是「單人完成一個問題」還是「多人完成一個問題」分成兩類。單人完成一個問題通常是使用者能在很短的時間內解決的問題，這類型的系統有[21][40][47]。多人完成一個問題代表需要許多人才能夠完成一個問題，這類型的系統有[1][3][14][15][33][35]。

在「單人完成一個問題」類型的系統中，最廣為人知的為 CAPTCHA [40]，CAPTCHA 是一個程式，能夠將文字轉換成為扭曲文字的圖像，讓機器沒有辦

法辨識圖像中的文字，而人可以輕易地識別圖片中的字串，所以 CAPTCHA 被廣泛使用在防止機器人大量註冊或是大量發言的地方，像是各大網站的電子信箱註冊系統，例如 Google、Yahoo 和 MSN，還有各大討論區留言板的發文系統上，另外也廣泛使用在電子投票系統上，藉此有效地杜絕機器人大量灌票的行為，也是第一個將人智運算運用在網路安全領域的系統。reCAPTCHA [47]是 CAPTCHA 的延伸，其目的是為了改進 Optical Character Recognition (OCR)的結果，其中 OCR 是將圖片中的字串抽取出來的技術。reCAPTCHA 提供含有兩組字串的圖片讓使用者輸入，其中一張是從網路中得到含有字串的圖片，此圖片必須靠使用者將圖片轉成文字；另外一張和 CAPTCHA 一樣從文字轉換成的圖片，藉此驗證使用者輸入是否正確。所以使用此機制就能讓使用者在進行驗證時，同時也能進行 OCR 的工作，而且準確度比傳統程式還要準確。KA-CAPTCHA [21]是對 CAPTCHA 的機制進行延伸，目的是讓 CAPTCHA 的機制變成取得資訊的媒介。一般的使用者在使用 CAPTCHA 的時候輸入答案的正確性是很高的，這是因為使用者希望看到被保護的網頁，所以不得不提供正確的答案以通過驗證。根據使用者的心理可以提出一個問題讓使用者回答，藉此取得電腦無法回答的資訊。

在「多人完成一個問題類型」的系統中，Vipul's Razor [14]是一個防止垃圾郵件的工具。其工作的原理是使用者回報垃圾郵件位址並且根據使用者投票決定是否為垃圾郵件地址。Distributed Proofreaders [3]是使用人力校對和更正

Project Gutenberg [12]中的錯誤。其中 Project Gutenberg 是一個將書本電子化的計畫，不過其內容有許多的錯誤，不外乎 OCR 的程式辨識錯誤或是人工打字上的錯誤，而 Distributed Proofreaders 就是為了校正這些錯誤而產生的系統。Distributed Proofreaders 會提供一張內文的圖片和一段經由 OCR 所產生的文字，使用者只要將錯誤的地方更正，系統將只要將一段文字經過許多人的驗證即可知道 OCR 所產生錯誤的文字。Wikipedia [15]是一個網路上的百科全書，任何使用者都能建立條目或者更改條目的內文。因為傳統的百科全書無法包含非常廣泛，所以提供一個平台讓任何人都能增加自己想要看的條目，或是使用自己的知識讓條目中的內容更加豐富，所以這也讓 Wikipedia 所包含的內容是傳統百科全書所不及的。Amazon Mechanical Turk [1]是一個平台提供使用者以金錢請其他使用者進行協助，使用者能提出一個工作而其他使用者幫忙即可獲得金錢。LabelMe [33]是一個網頁工具，讓使用者自行圈選圖片中的物件並且進行標記，系統藉此得知圖片中物件的位置和描述。Shenoy 和 Tan 的研究[35]是使用腦波進行圖片上的分類，因為使用者觀看不同種類圖片的時候也會發出不同的腦波，所以收集足夠的訓練資料就能根據使用者看到圖片所發出的腦波進行圖片分類。

第二節　屬於 GWAP 的人智運算系統

自 Dr. Luis von Ahn 在 2004 年提出 ESP 遊戲[41]之後，讓許多人把目光投注在此遊戲上，並且提出 GWAP 的概念[42]。有別於其他種類的人智運算系統，

使用 GWAP 可以讓許多玩家在進行遊戲之時，替系統產生許多有用的資料，而且因為玩家熱愛遊戲的感覺，所以玩家幾乎是以著迷的方式進行遊戲。由此可知玩家是很樂意地為幫系統進行運算，而且玩家也能在遊戲過程中得到滿足。在 GWAP 系統中大略可以分成四種類型，對影像進行處理[8][20][24][25][41][43][44][46]、對語言文字進行處理[29][45]、對聲音進行處理[27][28][30][38]和收集有關地理資訊的 GWAP 系統[17][19][23][31][32]。

對影像進行處理的 GWAP 系統中最早被開發出來是 ESP 遊戲[41]，ESP 遊戲主要是為了對影像進行標記，遊戲中得到的資訊能夠用在圖片搜尋等方面。遊戲中系統會隨機選擇兩個玩家進行遊戲，系統會讓雙方看同一張圖片而玩家則使用單字描述圖片，若是雙方輸入相同的單字則可以得到分數。Google Image Labeler [8]是 Google 在 2006 年向 ESP 遊戲購買版權，其目的是為了增強 Google 本身 Google Image Search [9]對影像搜尋的能力。其遊戲的方式和 ESP 遊戲大致相同，不過有些不太一樣，像是遊戲時間、遊戲回合數和得分系統等。Peekaboom [46]是為了將 ESP 遊戲中所得到的資料進行更進一步的處理而產生的遊戲，其目的是為了得到圖片中標記所在的位置。遊戲也是兩人一起進行遊戲，其中一個玩家為出題者，系統會給此玩家一個標記讓玩家標出標記在圖片中所在的位置；另一個玩家則根據出題者在圖片中所標出的物件選擇正確的標記，選擇正確則可以得到分數，而遊戲中每回合玩家雙方會輪流當出題者和解題者，讓玩家都能當到出題者和解題者。PHETCH [43][44]則是找出圖片和句子

的關係。和 Peekaboom 遊戲的方式類似，遊戲是兩人一起進行，其中一個玩家為出題者，系統會給此玩家一張圖片，且此玩家用句子描述此張圖片；而另外一個玩家則根據出題者所描述的句子選擇正確的圖片，若是選擇和出題者所看到的圖片相同則可以得到分數。Matchin [24]是為了得到人對圖片的喜好程度所創造的遊戲。遊戲是兩人一起進行，玩家會看到兩張圖片並且選擇玩家所喜好的圖片，若雙方選擇相同的圖片則可以得到分數。PhotoSlap [20][25]是為了將圖片進行分類而創造的遊戲。遊戲由四個人進行，每個玩家手中都有若干圖片，玩家依照順序丟出手中的圖片到桌上，若是有相同種類的圖片在桌上則 slap，先 slap 的人可以得到分數，不過若是不同種類的圖片 slap 則會扣分。PhotoSlap 特殊的地方在於其設計系統時採用賽局理論對系統進行分析，確認玩家的行為和遊戲設計者所設計的方向一致。

對語言文字進行處理的 GWAP 系統中 Verbosity [45]是透過遊戲取得人對單字的形容。遊戲有兩個玩家一起進行，其中一個玩家是描述者會對系統提出的單字進行描述，另一個玩家根據描述猜出系統所提出的單字。遊戲時間是 4 分鐘，在遊戲中雙方的角色每一個回合都會互換，如果這一回合是描述者則下一回合變成猜謎者，若是猜謎者成功猜到系統提出的單字則描述者和猜謎者都能得到一樣的分數。遊戲得到單字的描述能夠用在自然語言處理上。Common Consensus [29]是一個網頁的遊戲，用來取得一般人對問句直覺的答案。遊戲時系統會秀出一個問句然後玩家會回答這個問題，系統根據玩家本身的答案和其

他人的答案作比較並給予分數。Common Consensus 和 Verbosity 一樣，遊戲中所得到的結果都用在自然語言處理上。

對聲音進行處理的 GWAP 系統中。TagATune [27][28]是一個為了得到音樂描述和性質的遊戲。遊戲是由一對玩家進行遊戲，玩家雙方都會聽到系統所提供的音樂，玩家對本身所聽到的音樂進行描述，最後玩家猜對方所聽到的音樂和自己所聽到的音樂是否相同，若是雙方的選擇都相同就能得到分數。MajorMiner [30]是一個網頁上的遊戲，其目的也是取得人對音樂的標記並且根據這些標記對音樂進行分類。遊戲中系統會撥放 10 秒的音樂，玩家會根據所聽到的音樂進行標記，並且系統根據玩家本身輸入的標記和其他玩家的標記進行比較然後給予分數。Listen Game [38]也是一個網頁上的遊戲，其目的也是試圖建立音樂和單字之間的關係。遊戲是由線上所有玩家一起進行遊戲，系統會讓玩家聽 15 秒的音樂，在聽音樂的同時玩家也同時回答系統的問題並選擇玩家本身認為正確的選項，遊戲是採取多數決，也就是愈多人和玩家本身選擇相同的答案則得到的分數愈多。玩家在遊戲中能夠得到系統中和玩家本身有著相同音樂喜好的人，另外玩家也可以在遊戲中得到玩家本身音樂喜好的資訊。

有關地理資訊的 GWAP 系統和前面三者些許不同，此類型的系統大部分建構在行動裝置上面，藉此在行動中順便得到和地理有關的資料。Gopher [19]是一個建立在有 GPS 和拍照功能手機上的遊戲，其目的在收集地理上的標記和圖片。遊戲分成兩個部分，一部分是在手機上面進行遊戲，玩家可以建立一個新

的任務，任務包含去某地標記或是照相，若是玩家覺得無法完成此任務則可以讓其他玩家接手，任務完成時則讓網路使用者判斷是否完成；另一方面是網路上的玩家在網頁上觀看手機玩家任務執行的狀況，若是許多網路上的玩家都認定此任務已經完成，則所有參與此任務的手機玩家們就能得到分數。系統可以用遊戲中所收集的標記和圖片得到地區的描述。MobiMissions [23]和 Gopher 的目的和遊戲機制幾乎相同，不同的地方在於 MobiMissions 傳送地理座標不是使用 GPS 而是使用基地台座標，所以 MobiMissions 只要在擁有拍照功能的手機上即可進行遊戲。遊戲過程 MobiMissions 和 Gopher 類似，除了任務最多由 5 張圖片或是 5 段文字訊息所組成，其餘幾乎都和 Gopher 類似，像是一個任務可以由許多玩家共同完成和透過網頁讓網頁使用者驗證玩家的任務是否已經完成。CityExplorer [31][32]是一個在有 GPS 和拍照功能手機上的遊戲，其目的是用來標記地理上像是酒吧或是公園之類的設施。其遊戲方式為系統會將地圖分成若干區域，並且系統會告知玩家要找什麼設施，玩家找到後就會上傳圖像和地區的資訊，一段時間後遊戲停止，系統會計算每個區域哪個玩家有比較多的標籤或是圖片則那塊區域就屬於該玩家，最後擁有最多區域的玩家則為優勝。Eyespy [17]是一個在有 GPS 和拍照功能手機上的遊戲，其目的是為了收集地理上的標記和圖片。系統會提供一個任務地圖給玩家，地圖中有許多的任務，玩家可以選擇靠近自己位置的任務，而任務就是對此地區拍照或是標記，玩家拍完照片後將照片傳送給附近的玩家，讓其他玩家驗證所拍的圖片是否為任務所

需，若是玩家被驗證完成任務則可以得到分數。Eyespy 和 Gopher 與 MobiMissions 最大的不同在於其認證玩家是否完成任務不是透過網頁靠網路使用者負責檢驗，而是使用附近的玩家幫忙進行驗證的工作。

第三節　分析人智運算系統的相關研究

除了建立實際的人智運算系統之外，還有許多的研究是討論人智運算的架構和特性[22][26][34][36][37][42][48]。

在討論人智運算的安全性上，[22]是指出人智運算中系統安全性和可靠性的問題，像是在一個系統中能夠容許多少個惡意使用者而不影響系統的運行。並且在文中說明系統使用 Bayesian inference 決定答案時可以避免多數決中常發生的問題，像是大多數的答案未必是最佳解。

在分析人力質量的方面，[37]指出人智運算只要靠專家即可完成工作，而不需要浪費大量的人力在運算上，並且使用 Yahoo! Answers 驗證其觀點。不過[36]是指出專家和非專家對同樣問題的答案在品質上是沒有許多差異，並且使用 Amazon Mechanical Turk 進行驗證。在實驗中發現專家和非專家對同一問題所回答答案的品質；一個專家大約等同於四個非專家。[34]是指出只要透過重複地標記就能夠提升標記的品質，雖然會浪費些許人力不過在可以大量提升標記品質的條件下浪費些許人力是值得的。

[42]是屬於對 GWAP 系統進行分類的研究，其研究將 GWAP 系統分成三種類型，Output-agreement games、Input-agreement games 和 Inversion-problem

games。Output-agreement games 像是 ESP 遊戲，遊戲雙方輸入相同單字後即得到一個 agreement。Input-agreement games 像是 TagATune，遊戲雙方根據互相溝通描述後進行選擇，當雙方選擇相同時即得到一個 agreement。Inversion-problem games 像是 PHETCH、Verbosity 和 Peekaboom，遊戲雙方其中一方當成出題者或是描述者，另一個玩家根據描述進行猜測或是選擇，若是輸入和題目正確的話即得到一個 agreement。

針對 ESP 遊戲的研究中，[26]是使用賽局理論對 ESP 遊戲進行分析，其將 ESP 遊戲歸類為 Bayesian game，即玩家雙方並不知道彼此的行為和策略，並且得到雙方玩家會選擇簡單的單字並且會優先輸入常見的單字為一個 Bayesian-Nash equilibrium，換句話說，玩家的行為會傾向選擇簡單的單字並且會優先輸入常見的單字。[48]是指出 ESP 遊戲中所產生的標記過於簡單且關聯性太高，若是關聯性太高則不需要浪費人力產生這些關聯性高的標記，用程式即可產生這些關聯性較高的單字。為了驗證此論點，實作機器人在 Google Image Labeler 上進行遊戲，並且根據這些經驗提出新的得分系統用以改進 ESP 遊戲，讓 ESP 遊戲產生的標記能夠比較有價值。

第三章　系統描述

本章節主要是對 ESP 遊戲進行描述和說明如何進行 ESP 遊戲，並建立測量標準用來測量 ESP 遊戲系統的效能，且依此對 ESP 遊戲系統進行理論上的分析。

第一節　ESP 遊戲簡介

ESP 遊戲[41]是在 2004 年 Dr. Luis von Ahn 所創造的第一個 GWAP 系統，目前 ESP 遊戲和 von Ahn 創造的許多 GWAP 系統都在 gwap.com [10]中。在 ESP 遊戲中，玩家進入系統之後，伺服器會隨機配對一個夥伴和玩家一起進行遊戲。一場遊戲的時間是兩分半鐘。在遊戲開始後伺服器會隨機選擇一張圖片秀給雙方玩家，而玩家會輸入在圖片中看到的任何東西。若是雙方在同一張圖片輸入同樣的單字則此時成功得到一個 agreement。玩家若是看到系統秀出的圖片太難猜或是已經輸入好幾個單字但是卻得不到 agreement 時則可以送出 pass 訊息，若是遊戲夥伴也同意則可以跳過此張圖片。無論得到一個 agreement 或是遊戲雙方都 pass 此張圖片，系統都會隨機秀另外一張新的圖片給玩家繼續進行遊戲，這段過程即稱為一個回合。在兩分半鐘的遊戲中，玩家最多能玩 15 張圖片；換句話說就是一場遊戲最多只有 15 個回合。而每得到一個 agreement 就能得到 100 分，並且每 5、10、15 個 agreement 就會有額外的紅利分數。目前在遊戲中是得到一個 agreement 則變成此張圖的 taboo word，即此張圖片的標記，玩家不能輸入已經變成 taboo word 的單字。此外在遊戲中是無法和遊戲夥伴進行交

談，同時也不會知道一起遊戲夥伴的身分。在遊戲結束之後，玩家能夠看到遊戲紀錄，紀錄中顯示玩家本身和夥伴所輸入過的單字；除此之外，玩家還能透過對話視窗和夥伴討論遊戲經驗。圖 1 是 ESP 遊戲進行中的畫面。在圖中能看到系統隨機秀出的一張圖圖片，taboo words、已經輸入過的單字、目前的分數和遊戲所剩餘的時間。

Google 為了增強 Google Image Search [9]對圖片搜尋的效率，所以在 2006 年向 von Ahn 購買授權並且創造自己的 ESP 遊戲 Google Image Labeler [8]。ESP 遊戲和 Google Image Labeler 有著些許的差異。在名稱的差異上，ESP 遊戲中的 taboo words 在 Google Image Labeler 中變成 off-limits，不過在 Google Image Labeler 中只會秀出最多 5 個 off-limits。除了名稱有些不同之外，Google Image Labeler 中一場遊戲中沒有限制可以標記圖片的數量，而 ESP 遊戲中一場遊戲

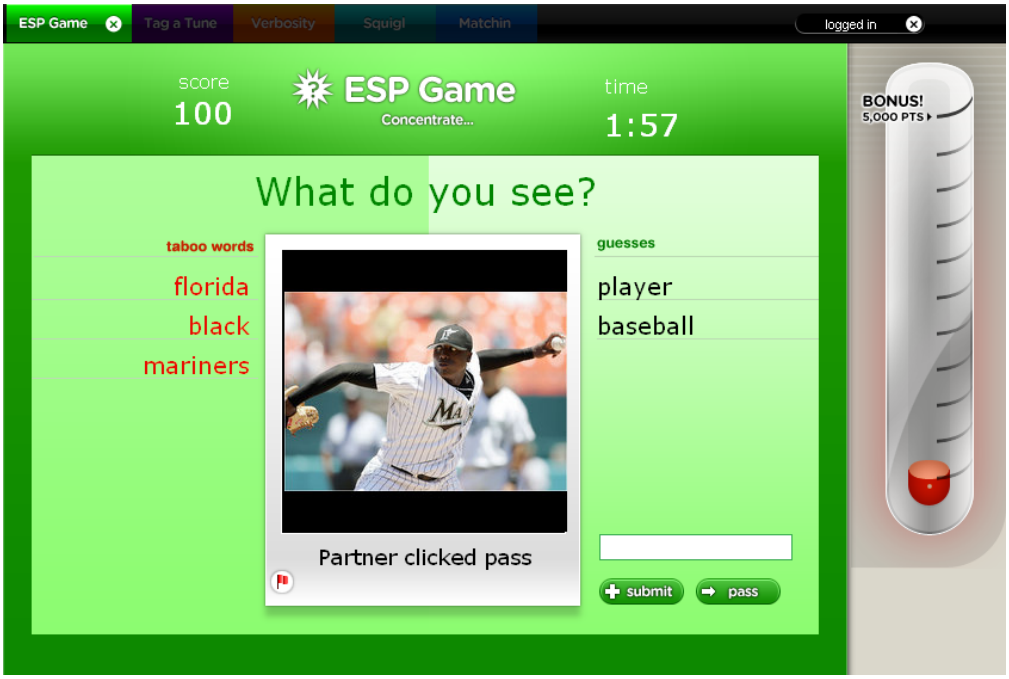

圖 1 ESP 遊戲進行中的畫面

最多只能標記 15 張圖片。Google Image Labeler 中一場遊戲時間也只有兩分鐘，而 ESP 遊戲為兩分半鐘。除此之外，在得分系統的方面也有很大的不同，ESP 遊戲固定每個 agreement 為 100 分，除了每 5、10、15 個 agreement 會有額外的紅利分數；而 Google Image Labeler 中每個 agreement 的分數根據其描述詳細的程度從 50 到 150 不等。例如 sky 可能只有 50 分，bird 有 60 分，soaring 有 120 分而 frigate bird 有 150 分。

第二節　系統模型

在 ESP 遊戲中，定義一個理想的 ESP 遊戲系統應該是質量兼備。在「質」的方面代表系統中平均每張圖片所擁有的標記數，在「量」的方面代表系統中被標記圖片的數量。不過在考量 ESP 遊戲和 Google Image labeler 之後，發現每個標記的品質應該是不同。在此使用分數表示標記的品質，所以「質」是代表系統中平均每張被標記過圖片所擁有的分數。所以一個理想的 ESP 遊戲系統在「質」的方面平均每張被標記過圖片所擁有的分數愈多愈好，在「量」的方面被標記圖片的數量要愈多愈好。不過這兩項指標互相衝突，因為以「量」而言，系統會希望系統中的圖片都有一個標記，這樣可以最大化「量」的部分；以「質」而言，系統會希望所有標記都集中在同一張圖片上，這樣可以最大化「質」的部分，所以必須透過分析找出「質」和「量」的平衡點，使 ESP 遊戲系統的效能達到最佳。

為了簡化系統，在本章中設定每玩一張圖片就一定會有一個 agreement 產生，不會有失敗的情形出現，也就是每一回合就會有一個 agreement 產生，並且每回合都會產生新的標記。所以根據上述的描述，設定系統中有 M 張圖片，其中有 N 張圖片被標記過。S 是一個隨機變數代表每個標記的分數，而 $S_i \in S$ 代表第 i 個標記的分數。定義系統測量函數 G 如下所示：

$$G = \ln(N)\ln(\bar{S}^*) \tag{1}$$

由等式 1 得知 G 由兩個部分所組成，$\ln(N)$ 和 $\ln(\bar{S}^*)$，分別代表系統的「量」和「質」。其中 $\bar{S}^*$ 代表平均每張被標記過圖片所擁有的分數，其表示式為

$$\bar{S}^* = \frac{\sum_i S_i}{N} \tag{2}$$

所以 $\ln(\bar{S}^*)$ 為測量系統的「質」；而 N 為被標記圖片的數量，所以 $\ln(N)$ 為測量系統的「量」。在此篇文章會使用此函數測量系統的效能。

在定義 G 之後，就可以根據系統中的變數和 G 最佳化系統。設定在系統運行的某段時間點系統中總共運行 T 個回合，且平均每張被標記過的圖片有 r 個標記。因為根據設定每回合都有一個標記產生，所以可以將等式 1 改寫如下所示：

$$\begin{aligned} G &= \ln(\frac{T}{r})\ln(\bar{S}\cdot r) \\ &= (\ln(T)-\ln(r))(\ln(\bar{S})+\ln(r)) \\ &= -(\ln(r))^2 + (\ln(T)-\ln(\bar{S}))\ln(r) + \ln(T)\ln(\bar{S}) \\ &= -(\ln(r) - \frac{\ln(T)-\ln(\bar{S})}{2})^2 + C \end{aligned} \tag{3}$$

其中 C 為常數，其值為

$$C = \ln(T)\ln(\bar{S}) + (\frac{\ln(T) - \ln(\bar{S})}{2})^2$$
$$= (\frac{\ln(T) + \ln(\bar{S})}{2})^2 \quad (4)$$

$\bar{S}$ 為標記平均的得分，其值為

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i \quad (5)$$

根據等式 2，所以在

$$\ln(r) = \frac{\ln(T) - \ln(\bar{S})}{2} \quad (6)$$

時G 有最大值C 。將等式 6 兩邊同時代入e^x 中，則得到當

$$r = e^{\frac{\ln(T) - \ln(\bar{S})}{2}} \quad (7)$$

時G 有最大值C 。

因為當時間增加時，T 只會增加不會減少，所以時間和T 是成正比的關係。

由此得知當時間趨近於無窮時，T 也會趨近於無窮。根據大數法則

$$\bar{S} \to E[S] \text{ as } T \to \infty \quad (8)$$

所以可以將等式 7 改寫成

$$r = e^{\frac{\ln(T) - \ln(E[S])}{2}} \quad (9)$$

由等式 2、等式 4 和等式 8 可以得到

$$G(T) = (\frac{\ln(T) + \ln(E[S])}{2})^2 \quad (10)$$

等式 10 代表系統在穩定狀態時理論上的最大系統效能，而且只有一個變數T 。

換句話說，任何時候只要取得系統當時的T 就能得知當時理論上最大效能。

所以在此節中得到等式 7 和等式 9 這兩個重要的式子，得到能使 G 最大最大化的 r，也代表著理想狀態下一張圖片應該要有多少個標記才合適。其中等式 7 適用在系統實作上，而等式 9 適合用在系統穩定狀態時的理論分析。

第四章　系統策略

在第三章等式 7 中得到理論上系統中一張圖片應該要有 r 個標記之後，本章根據此發展能夠增加 ESP 遊戲系統效能的圖片選擇演算法 Optimal Puzzle Selection Algorithm (OPSA)。為了和 OPSA 比較，所以提出 Random Puzzle Selection Algorithm (RPSA)和 Fresh-first Puzzle Selection Algorithm (FPSA)。並且最後以此三種演算法進行系統模擬並且分析其行為。

第一節　圖片選擇演算法 RPSA 和 FPSA

在此節提出二個圖片選擇演算法 Random Puzzle Selection Algorithm (RPSA)和 Fresh-first Puzzle Selection Algorithm (FPSA)，分別表示為演算法 1 和演算法 2。其中 RPSA 為每回合隨機從圖片集合 P 中隨機取出圖片 p ，函數 $Select\_Random(P)$ 即是表示此行為。RPSA 主要目的在當作 ESP 遊戲系統效能的基準。

而 FPSA 則是每回合從圖片集合 P 中選出最少標記數量的圖片 p 給玩家進行遊戲，函數 $Select\_Fresh(P)$ 即是表示此行為。FPSA 主要目的在最大化 $\ln(N)$，也就是最大化 G 中「量」的部分。在此研究中，FPSA 用來了解只重視

演算法 1 Random Puzzle Selection Algorithm(RPSA)	
1:	**Function RPSA**
2:	$p \Leftarrow Select\_Random(P)$
3:	Return p

演算法 2 Fresh-first Puzzle Selection Algorithm(FPSA)

1: **Function FPSA**
2: $p \Leftarrow Select\_Fresh(P)$
3: Return p

一個變數會對系統有何影響。

第二節　最佳化效能的圖片選擇演算法 OPSA

從第三章得知欲使系統效能最大，則理想的r值為等式 7 所表示，在提出理想的圖片選擇演算法之前，先提出 *Refresh_Group*() 為演算法 3。*Refresh_Group*() 是為了將圖片分成三個集合P_0、P_1和P_2。其中P_0是還沒有被標記過圖片的集合，即在P_0中的圖片 $p.num\_of\_label = 0$；P_1是圖片標記數大於 0 小於r的集合，即在P_1中的圖片$0 < p.num_of_label < r$；P_2則是圖片標記數

演算法 3 Refresh_Group

1: **Function Refresh_Group**
2: $r = e^{\frac{\ln(T)-\ln(\bar{S})}{2}}$
3: $P_0 = \varnothing$
4: $P_1 = \varnothing$
5: $P_2 = \varnothing$
6: **for** each p in P **do**
7: **if** $p.num\_of\_label = 0$ **then**
8: move p to P_o
9: **else if** $p.num\_of\_label < r$ **then**
10: move p to P_1
11: **else**
12: move p to P_2
13: **end if**
14: **end for**
15: Return

大於等於 r 的集合，即在 P_2 中的圖片 $p.num\_of\_label \geq r$ 。這是為了方便後續處理所以先將圖片進行分類。

為了最大化 ESP 遊戲系統的效能，所以每張被標記的圖片應該要有 r 個標記。所以在概念上應該要優先把 P_1 中每個圖片的標記數增加到 r 以上。根據這概念，提出理論上能夠最大化效能的圖片選擇演算法 Optimal Puzzle Selection Algorithm (OPSA)，演算法 4 描述其細節。

OPSA 在 P_1 中有圖片的時候則會優先選擇擁有最多標記的圖片，因為優先選擇這些圖片給玩家遊戲，就能比較早將這些圖片的標記數大於等於 r ，即愈早讓此圖擁有理想的 r ， $Select\_Played(P_1)$ 描述此情形。若是 P_1 為空集合代表所有圖片不是還沒被標記就是被標記過的圖片擁有標記的數目超過 r ，此時則會檢查是否還有沒被標記的圖片。若 P_0 中還有圖片代表還有圖片沒有被標記過，則此時以 RPSA 的模式隨機選擇 P_0 中的圖片進行遊戲。最後一種情形是 P_0

演算法 4 Optimal Puzzle Selection Algorithm(OPSA)

```
1:  Function OPSA
2:   Refresh_Group()
3:  if |P1|>0 then
4:      p ⇐ Select_Played(P1)
5:  else
6:      if |P0|>0 then
7:          p ⇐ Select_Random(P0)
8:      else
9:          p ⇐ Select_Fresh(P2)
10:     end if
11: end if
12: Return p
```

和 P_1 均為空集合，這代表所有圖片所擁有的標記數都超過 r，則此時運行 FPSA 的模式選擇 P_2 中標記最少的圖片進行遊戲。所以 OPSA 在系統一開始的時候是以 RPSA 的模式在運行，而當系統飽和也就是圖片所擁有的標記數超過 r 時，OPSA 是以 FPSA 的模式運行。

第三節　系統模擬

在此節將會對此三種演算法 RPSA、FPSA 和 OPSA 進行系統模擬。不過在進行系統模擬之前先觀察 r 和 $E[S]$ 的關係，因為這和得分系統的設計有所關係。$E[S]$ 這個參數非常重要是因為 r 的大小直接影響到一張被標記過的圖片應該要有幾個標記，而 $E[S]$ 影響 r 的大小，r 和 $E[S]$ 的關係可以用等式 7 表示，若是系統為穩定狀態則可以用等式 9 表示。若是 r 的值太大則不切實際，因為有可能一張圖片根本沒有包含那麼多資訊；若是 r 的值太小則 OPSA 會變成

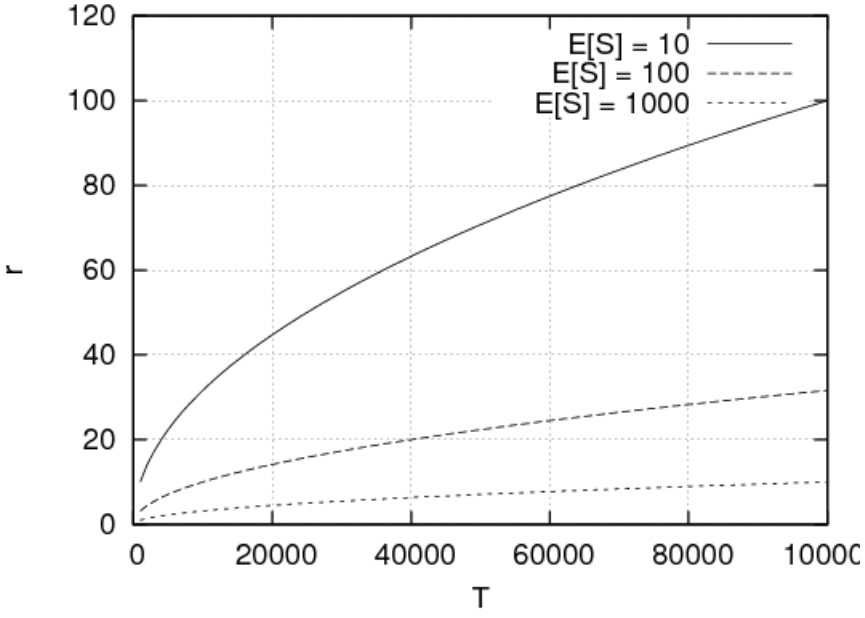

圖 2 不同 $E[S]$ 的 r 和回合數總合 T 之間的關係

FPSA，因為所有圖片都在P_2中。從等式 9 得知r和T、$E[S]$有關。因為T和時間是成正比的關係，所以廣義來說T的增加也代表著時間的進行。觀察在不同的$E[S]$之下T和r的關係。圖 2 表示這個關係，在圖中可以清楚地表示隨著$E[S]$的增加，在T固定的情形下$E[S]$愈小則r愈大。在$T=100000$時，$E[S]=10$的r和$E[S]=1000$的r相差 10 倍左右，而$E[S]=10$的r和$E[S]=100$的r相差 3 倍左右。代表不同的$E[S]$對r的影響非常大。

接下來進行系統模擬，其中設定系統中圖片的數量M為 100,000；並且和 ESP 遊戲一樣，設定系統中得到一個 agreement 即得到一個標記。而系統中每個 agreement 的得分為從 60,70,…,150 中以均勻分佈隨機選取一個分數當作 agreement 的得分，所以系統中的$E[S]=105$。得分系統會這樣設計的原因是因為希望能夠每個標記的得分不會差距太大，而且任一 agreement 得到每個分數

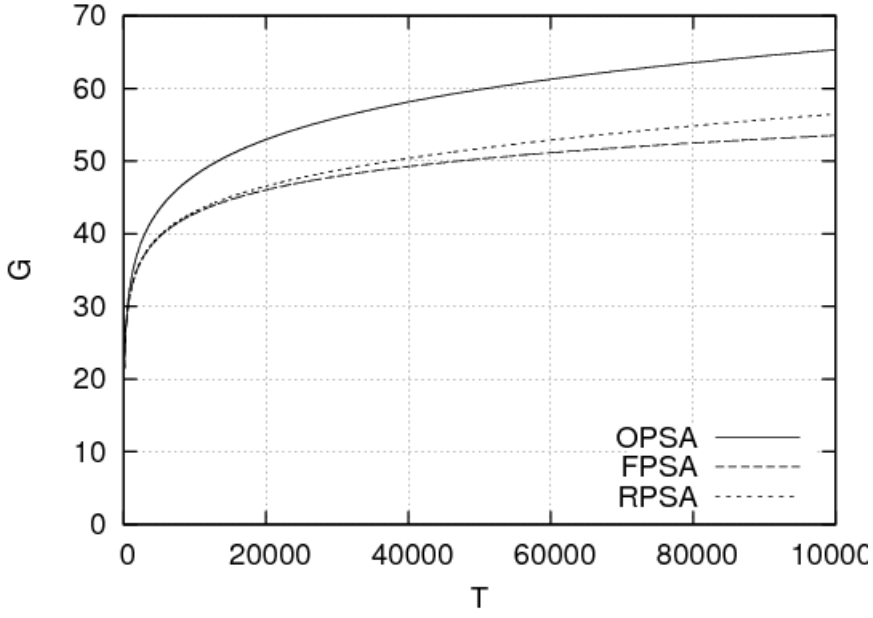

圖 3 在模擬中，三種演算法的系統效能和回合數總合T的關係

的機率均為相等，不會有某種分數特別多 agreement 的情形發生，而 $E[S]$ 約為 100 是因為在圖 2 分析的結果得到 $E[S]$ 等於 100 左右的時候 r 的值相較之下不會太高。在第三章得到任意時間系統的最大效能為等式 10，而等式 10 只有一個變數 T。所以在此以回合數總合 T 觀察系統的效能，圖 3 顯示三種演算法的系統效能和 T 之間的關係。此圖是系統進行二十次模擬平均的結果。此圖表現 FPSA 因為極端地重「量」而不重「質」導致系統效能為最低，而 OPSA 是根據最佳化的 r 而產生出的圖片選擇演算法，所以 OPSA 的表現為最佳。RPSA 則是沒有偏好「質」和「量」，所以 RPSA 的效能在 OPSA 和 FPSA 之間，不過 RPSA 的效能是比較接近 FPSA，這代表 OPSA 能夠大大地提升 ESP 遊戲系統的效能。

第五章　系統實作

為了實地驗證在第四章所提出的圖片選擇演算法 OPSA 能夠實際增加 ESP 遊戲系統的效能，所以建立一個名為 ESP Lite 實為類似 ESP 遊戲的實驗系統。在本章主要討論實驗系統 ESP Lite 設計的細節、架構。並且也會在此章定義 ESP Lite 的得分系統。

第一節　系統架構

ESP Lite 主要是沿用 ESP 遊戲的設計，遊戲進行的方式大致和 ESP 遊戲相同，像是伺服器每三十秒會建立遊戲，系統會隨機配對兩人進行遊戲，也有 pass 圖片的機制，並且得到 agreement 和產生 taboo words 的機制也是相同。另外為了考量遊戲時間不要太久而導致玩家對遊戲產生厭惡感，所以遊戲設定和 Google Image Labeler 一樣均為兩分鐘，而不是像 ESP 遊戲為兩分半鐘。和 ESP 遊戲和 Google Image Labeler 不同的是為了研究上的方便，所以玩家能輸入的單字只能為英文和數字所組成的字串，如果用正規表示法表示則為 /^[A-Za-z0-9]+$/或是/^[\w\d]+$/。而為了鼓勵玩家進行遊戲[42]，所以和 ESP 遊戲與 Google Image Labeler 一樣設定計分板的功能。讓玩家在遊戲結束後若是分數達到總共或是當天的前十名，就可以讓玩家自己的名字放在計分板上，使全部的玩家都能看到。藉此機制刺激玩家還會回到遊戲中進行遊戲，並且對系統產生貢獻。

ESP Lite 在系統設計上採用主從式的架構，其中用戶端是以 Flash 實作，伺服端是以 Java 實作。ESP Lite 比較特殊的地方為此系統是為了驗證 OPSA 是否能夠實際地增加 ESP 遊戲系統的效能而實際開發出來的系統。為了避免隨機選擇三種演算法其中之一當作遊戲選擇圖片選擇演算法的依據，造成其中一種演算法的 agreements 有過少的情形，所以不採用遊戲數總合當作開啟遊戲選擇圖片選擇演算法的標準。所以伺服器建立遊戲時選擇圖片選擇演算法是根據演算法的回合數總合決定，選擇最少的當作此遊戲所使用的圖片選擇演算法。換句話說，就是使用回合數總合當作遊戲選擇何種圖片選擇演算法演算法的標準。原本是以每個演算法所擁有的 agreement 個數當作遊戲開啟時選擇圖片選擇演算法的依據，不過此方法在正式實行後被玩家抗議下就更改回使用回合數總合當作標準。

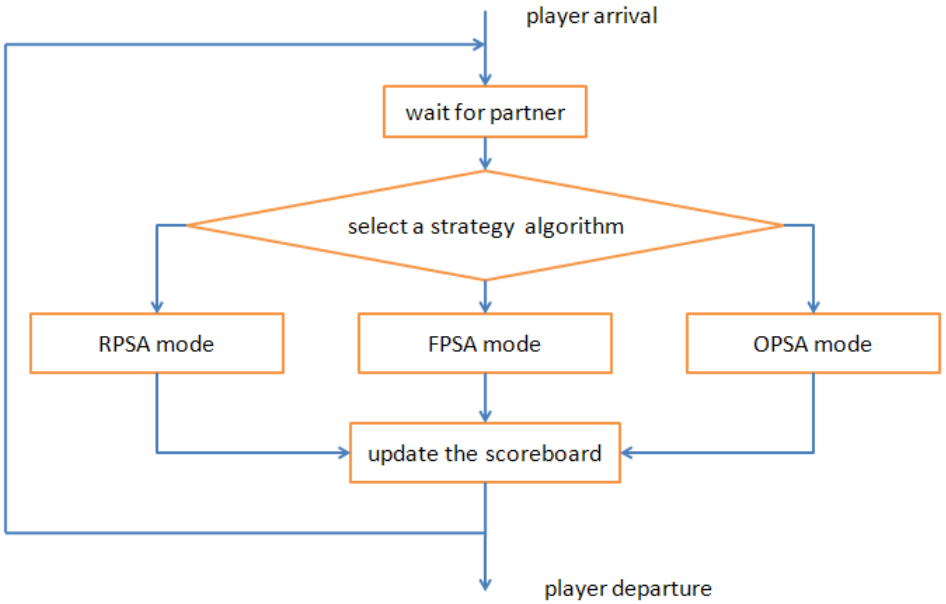

圖 4 ESP Lite 的系統流程圖

若是玩家剛好找不到夥伴或是遊戲在進行時其中一方發生斷線，此時就會使用機器人代替真實玩家進行遊戲。為了能夠讓機器人輸入的單字能夠像真人一樣，所以系統在收集圖片時會順便收集和此圖片相關的標記，讓機器人能在遊戲中所輸入的單字能夠和真人輸入的單字相近。

圖 4 是 ESP Lite 系統的流程圖，玩家進入伺服器後會等待至多三十秒內然後判斷是否有其他玩家和玩家一起遊戲，如果沒有則使用機器人代替玩家。系統創造遊戲時會判斷要使用何種演算法，此時伺服器會選擇回合數總合最少的演算法當作此遊戲的圖片選擇演算法。若是在遊戲中玩家中途斷線則由機器人接手，不過當兩個玩家都斷線的話則系統會關閉遊戲，所以不會存在兩個機器人在同一場遊戲的情形發生。若是玩家在此次遊戲的得分在總排行或是當天排行前十名，則能讓玩家的名子留在計分板上。遊戲結束後系統會儲存此次遊玩的資料進入資料庫。

圖 5 是 ESP Lite 客戶端實際的畫面，左邊是遊戲進行中的畫面。和 ESP 遊

圖 5 (左)ESP Lite 遊戲中的畫面(右)ESP Lite 遊戲結束時的畫面

戲與 Google Image Labeler 一樣，有顯示中的圖片、目前的分數、遊戲剩下的時間、此張圖片的 taboo words (Off-limites)和玩家在這回合所輸入過的單字。右邊是遊戲結束後遊戲紀錄的畫面，和 ESP 遊戲與 Google Image Labeler 一樣，玩家可以在此紀錄中看到每回合的圖片與每回合玩家本身和夥伴所輸入過的單字。讓玩家知道自己夥伴輸入單字的喜好，藉此可以增加自己的遊戲技能，以達到更高的分數。因為玩家在一場遊戲得到愈高的分數即愈有機會在計分上輸入自己的名子，讓玩家有目標地繼續進行遊戲。

第二節　資料庫

為了順利地運行系統，所以需要一定數量的圖片。而且為了遊戲中機器人的需要，系統必須尋找圖片和圖片相關的標記。圖片和標記的來源分別使用過 Flickr [6]、Fotki [7]和 ESP dataset [5]。其中 Flickr 和 Fotki 是圖片分享的網站，用戶可以上傳圖片和給圖片一些標記，所以系統可以在收集圖片時順便收集這些標記當成機器人輸入單字的依據。而 ESP dataset 是 ESP 遊戲所釋出的 100,000 張圖片和這些圖片各自的標記。

Flickr 的圖片標籤是由圖片擁有者所給予，而圖片擁有者時常會使用只對於他個人有意義的標籤，可是這些標籤並不是其他人能直接從圖片中所得到。使得機器人在使用這些標記當成單字輸出時，對方玩家在看遊戲紀錄時會感覺他的夥伴老是輸入一些很難得到標記的字。這個現象在[48]實驗進行中也有此發現。Fotki 圖片中的標籤含有比較多的網站廣告或是怪異的單字，和 Flickr 的

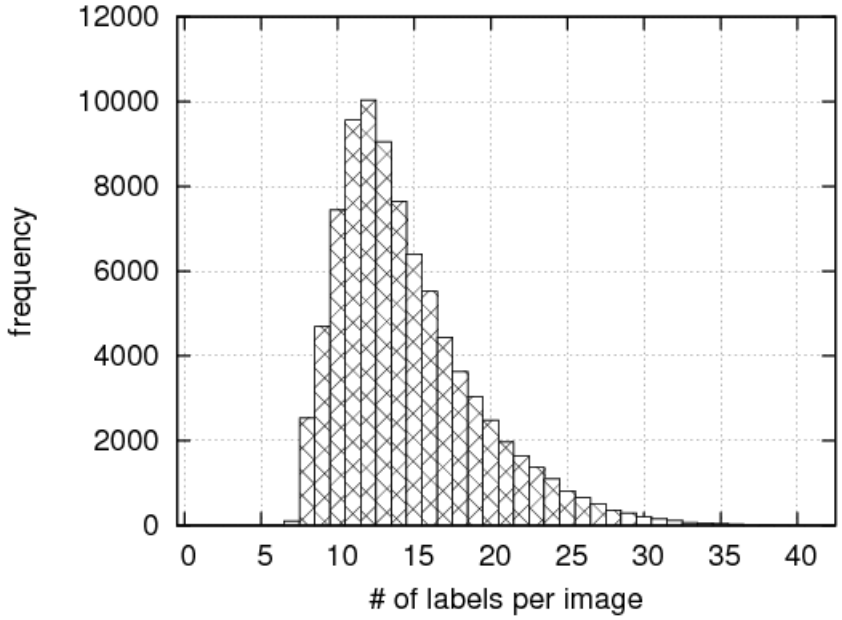

圖 6 ESP dataset 中每張圖片所有擁有標記數的統計

標籤同樣也會造成遊戲上發生問題。所以 Flickr 和 Fotki 中標記的品質都無法滿足系統的需求。

最後採用 ESP dataset 的原因是因為 ESP dataset 是由 ESP 遊戲中所得到真實遊戲的資料，而且每張圖片也擁有足夠的標記量，所以非常合適當成系統中機器人輸入的參考依據，不會有上述的情形發生。ESP dataset 中有 100,000 張圖片和 1,531,487 個標記，圖 6 是系統圖片所擁有標記數的統計，平均一張圖片約有 15 個標記。

第三節　得分系統

當遊戲中雙方對同一張圖片輸入同樣的單字時，即得到一個 agreement，並且得到分數。在 ESP 遊戲中是採用固定分數，分數固定為 100。Google Image Labeler 則是根據此 agreement 的得到分數，分數從 50 到 150 不等。不過 Google

Image Labeler 並沒有公佈其得分系統。分數的定義在第三章系統模擬中所定義的隨機變數 S 為從 60,70,…,150 的均勻分佈，在此節必須根據此精神定義實驗系統所使用的得分系統。一般來說，單字分數的高低是依據字出現的頻率。愈少見的單字給予愈高分；反之，愈常見的單字則給予愈低分。也就是認為出現頻率愈高的單字品質愈低；相反地，出現頻率愈低的單字品質愈高。為了制定得分系統，系統採取 edict virtual language centre [4]中的資料當作分數表的基準。edict virtual language centre 整理 Brown Corpus [2]中前 5,000 個頻率最高的單字，並且列出其頻率占 Brown Corpus 中字數的百分比。Brown Corpus 是研究自然語言領域的人常使用的一個工具，用來得到單字的頻率。

因為希望系統所使用的得分系統為每個分數等級的權重盡量相近，所以將分數的等級假設為均勻分佈；並且鼓勵一場遊戲中有愈多的標記產生，所以分數的差距不會太大。根據這理由而將系統的得分系統詳細定義如下段所示。

系統中的分數表有排序過的單字 $w_1, w_2, \cdots, w_n$ ，而每個字有其頻率 $f_1, f_2, \cdots, f_n$ ，且排序的標準是採取每個單字的頻率 $f_1 \geq f_2 \geq \cdots \geq f_n$ 。為了防止同樣字根但是不同詞性的單字有不同的分數，像是 determinant 和 determine；或是同樣單字但是不同的狀態有不同的分數，像是 experiments 和 experiment。所以為了避免上述的情形發生，系統採用 Porter Stemming Algorithm [13]處理單字，在此使用 $stem(\cdot)$ 表示此行為。Porter Stemming Algorithm 是研究自然語言的人常常使用的演算法，用來去除單字的字尾。系統採用此演算法避免上述的情

形發生，例如 study 和 studied 經過處理之後都會便成 studi。根據上述原因，系統中分數表中任兩個單字必須經過 Porter Stemming Algorithm 處理過後都不會一樣，這樣才能避免上述情形的發生。系統的得分系統有 k 個等級，每個等級的分數距離為 S_{offset}，基準分數為 S_{base}。所以系統中所能得到的分數為 $0 \cdot S_{offset} + S_{base}, 1 \cdot S_{offset} + S_{base}, \cdots, (k-1) \cdot S_{offset} + S_{base}$，而對每個系統中單字表中單字所對應的分數為

$$\begin{aligned} &\forall i \in 1,2,\cdots,n \\ &score(w_i) = L_i \cdot S_{offset} + S_{base} \end{aligned} \tag{11}$$

其中 L_i 表示 w_i 所在的分數等級，其值域為 $\{0,1,\cdots,k-2\}$，且表示式為

$$L_i = \left\lfloor \frac{\sum_{j=1}^{i-1} f_j}{\sum_{j=1}^{n} f_j} \cdot (k-1) \right\rfloor \tag{12}$$

在定義分數表中的對應後，對任何單字都能找到其對應的分數，所以對每個單字的分數為

$$\begin{aligned} &\forall w_{agreement} \\ &score(w_{agreement}) = \begin{cases} score(w_i), \exists! i \in \{1,2,\cdots,n\}, \text{s.t. } stem(w_{agreement}) = stem(w_i) \\ k \cdot S_{offset} + S_{base}, \text{elsewise} \end{cases} \end{aligned} \tag{13}$$

所以在本章得到等式 11 和等式 13 之後就能對任何 agreement 找到其對應的分數。

ESP Lite 得分系統所使用的參數為 $k = 10$、$S_{scale} = 10$ 和 $S_{base} = 60$，也就是系統中所能得到的分數為 $60,70,\cdots,150$，共十個等級。而分數表的單字是從 Brown Corpus 取出前 5000 個頻率最高的單字和其頻率之後，經過 Porter

表 1 ESP Lite 和 ESP game、Google Image Labeler 細節上的比較

遊戲名稱 / 項目	ESP game	Google Image Labeler	ESP Lite
每場遊戲時間限制	2.5 分鐘	2 分鐘	2 分鐘
每場遊戲回合數限制	15	不限	不限
每個 agreement 的得分	100（不過一場遊戲中第 5,10,15 個 agreement 有額外的紅利分數）	50,60,…,150（11 個等級）	60,70,…,150（10 個等級）
幾個相同的 agreement 變成標記	1	不明	1
秀出 taboo words 個數的限制	不限	5	不限

Stemming Algorithm 處理後剩下 3,476 個單字。所以根據等式 11 得知系統中的分數表中共有 3,476 個字和其對應的分數。

第六章　實驗結果

本章為 ESP Lite 實地實驗的結果。除了對實驗的敘述之外並且包含基本的統計行為、三種演算法和玩家在遊戲中所發生的行為分析，還有三種演算法之間效能的比較和分析。本章節中大部分是以回合數總合為標準觀察不同演算法之間的行為，這是因為 ESP Lite 是以回合數總合當作開啟遊戲的依據，也就是三種演算法在同一時間的回合數總合幾乎相同，所以可以視回合數總合為時間的延伸。

ESP Lite 從 2009/3/9 開始運行。系統中共有從 ESP dataset 中的 100,000 張圖片。EPS Lite 的主機架設在中央研究院資訊科學研究所內。ESP Lite 實際運行的網址為 http://nrl.iis.sinica.edu.tw/GWAP/ESPLite/。實驗統計自 2009/3/9 開始至 2009/5/9 止，在這二個月的時間中，一共創造 3,495 場遊戲，標記 10,611 張不同的圖片和得到 13,978 個標記。其中，屬於 OPSA 的有 1,601 場遊戲，標記 656 張不同的圖片和得到 3,895 個標記；屬於 FPSA 的有 977 場遊戲，標記 5,016 張不同的圖片和得到 5,016 個標記；屬於 RPSA 的有 917 場遊戲，標記 4,939 張不同的圖片和得到 5,067 個標記。

第一節　基本統計

首先觀察第五章所設計的得分系統在實際系統運行的情形，目的在比較 ESP Lite 和 ESP dataset 所收集標記品質上的差異。圖 7 是 ESP dataset 中標記套用 ESP Lite 的得分系統和 ESP Lite 中標記分數的累積分佈函數圖。從圖中可以發現無論 ESP Lite 或是 ESP dataset 的分數都在 100 分之後，而幾乎沒有標記的分數是 60 到 90 分。這是因為 Brown Corpus 是收集一般文章作為統計的標準，所以一般文章中常常出現的介詞、冠詞和代詞幾乎不會用來形容圖片。例如，一般不會使用 the、of 和 it 等形容圖片，但是這些字卻常常在一般的文章中出現。這也代表 ESP Lite 和 ESP 遊戲一樣，都是在系統中得到有意義的標記，不會有 a、the 和 it 等之類沒有意義的單字形容圖片。EPS Lite 和 ESP dataset 做比較，可以發現 ESP Lite 分數分佈的情形和 ESP dataset 相似，不過 ESP Lite 的分

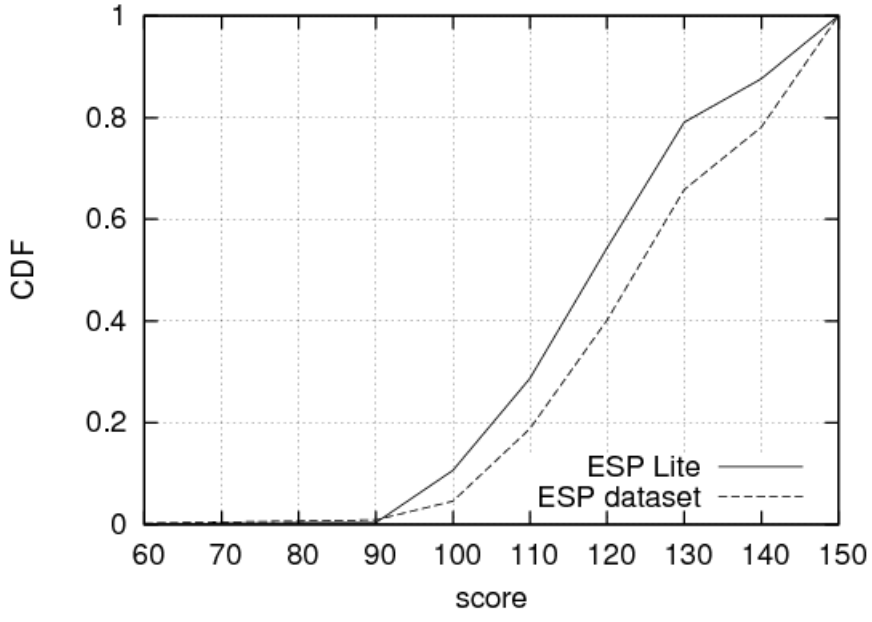

圖 7 ESP Lite 和 ESP dataset 中標記分數的分佈（採用 ESP Lite 的得分系統）

數分佈比 ESP dataset 還要往前移動。因為 ESP Lite 目前只運行二個月，所以這代表隨著時間的進行，分數會慢慢的朝向高分移動。換句話說，玩家在進行遊戲主要會輸入常出現的單字，也就是分數比較低的單字；時間愈久，因為簡單的單字都已經被輸入過，玩家不能輸入這些簡單的單字，只好輸入較難的單字，所以平均分數會愈來愈高。玩家喜歡輸入較簡單單字的現象在[48]也有相同的發現。

圖 8 是實驗中玩家進入系統時間的統計。圖中玩家進入系統中的時間和日常人的習慣上網的時間非常相近。玩家集中在 15 到 17 點和 21 到 0 點進入遊戲，這代表玩家都把 ESP Lite 當作休閒的遊戲用來在工作後放鬆身心。而 0 點到 1 點這段時間也有非常多的玩家上線。這是因為系統會在每天的 0 時更新計分板，所以玩家希望自己的名字出現在計分板上，所以造成這段時間也是有許多

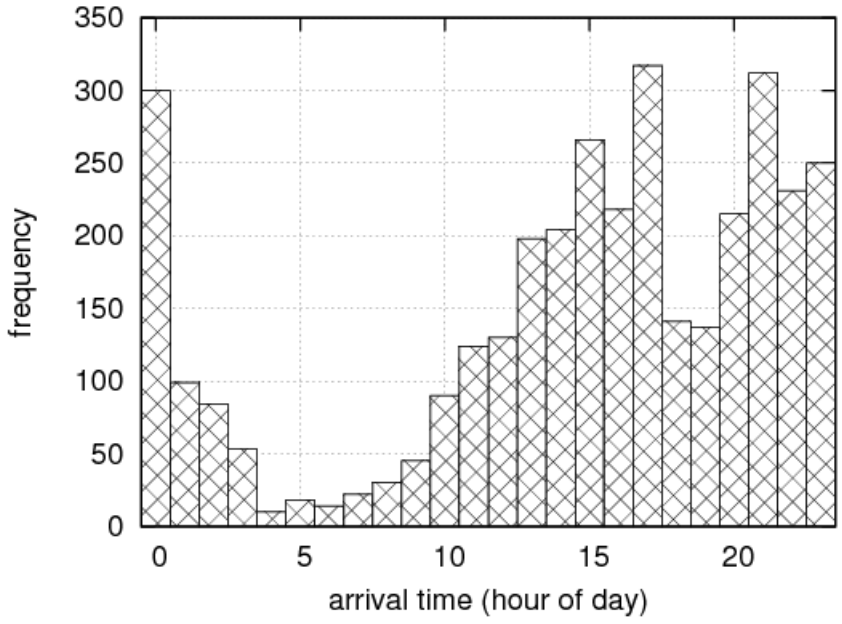

圖 8 玩家進入遊戲時間的統計

玩家上線進行遊戲。

三種演算法的 agreement 個數和回合數總合和的關係如圖 9 所示。回合數總合是創造遊戲時選擇演算法的基準，所以理論上同一時間點上三種演算法都會有相同的回合數總合。在圖中顯示 FPSA 和 RPSA 的 agreement 個數和回合數總合大致為線性的關係。不過 OPSA 的*T* 在回合數總合約為 700 的時候就開始有下降的趨勢，OPSA 的*T* 在回合數總合為 1000 時和 RPSA 與 FPSA 相差 100 左右，在 2,000 時相差 300 左右，在 3,000 的相差 400 左右，在 4,000 時相差 600 左右，5,000 時相差 900 左右而到 6,000 時相差 1,100 左右，顯示 OPSA 的 agreement 個數在回合數總合愈大的情形下和 RPSA 與 FPSA 的差距愈來愈大。所以這張圖描述在相同回合數總合的情形下 OPSA 的 agreement 個數會比 RPSA 和 FPSA 少，而且差距有隨著回合數總合的增加而增加的趨勢。換句話說，OPSA

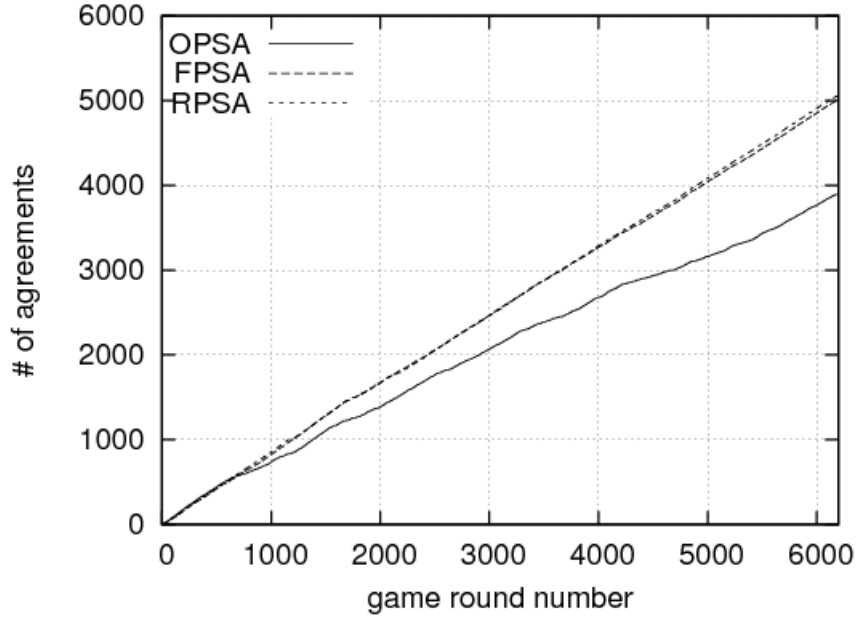

圖 9 三種演算法的 agreement 數和回合數總合的關係

中 agreement 個數的成長率比 FPSA 和 RPSA 來得較低。

三種演算法的回合數總合和開啟遊戲數量的關係如圖 10 所示。FPSA 和 RPSA 的開啟遊戲數量大致為線性關係，在回合數總合 300 之後在相同的回合數總合之下 FPSA 的開啟遊戲數量比 RPSA 多約 60 場左右。而 OPSA 在回合數總合超過 1,000 之後就開始大幅增加，在回合數總合 2,000 時 OPSA 和 FPSA 相差約為 100 場，3,000 時相差約為 170 場，4000 時相差約為 340 場，5,000 時相差約為 510 場，6,000 時相差約為 640 場。隨著回合數總合的增加 OPSA 和 FPSA 的遊戲數量差距愈來愈大，也就代表著 OPSA 的遊戲數量成長愈來愈快速，到回合數總合 6000 左右，OPSA 所開啟遊戲的數量大約為 RPSA 的 1.8 倍左右。

由圖 10 得知 OPSA 在回合數總合愈來愈多時所開啟的遊戲數量相較 FPSA 與 RPSA 也會愈來愈多。這代表 OPSA 平均每場遊戲所擁有的回合數會比 FPSA

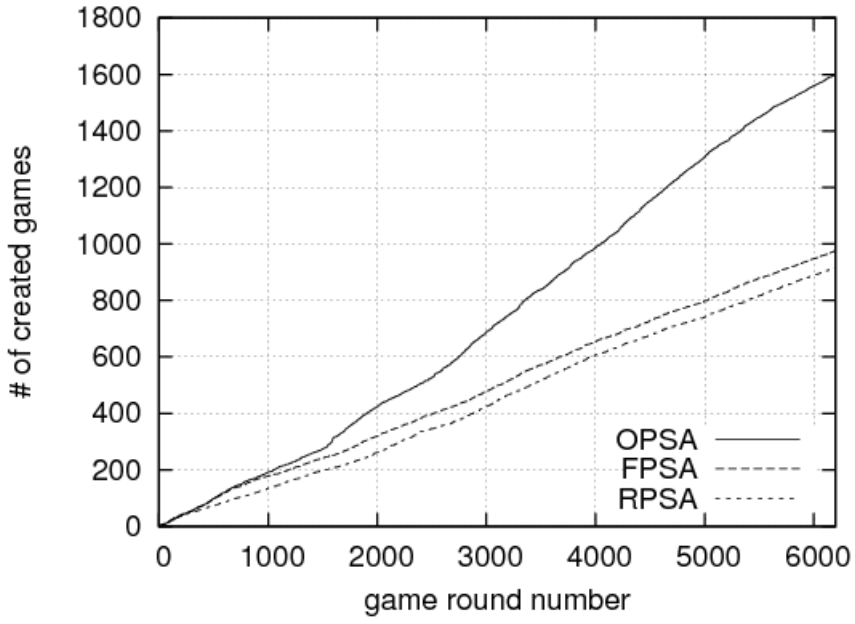

圖 10 三種演算法的開啟遊戲數量和回合數總合的關係

和 RPSA 來得多，圖 11 顯示這個現象。除了在回合數總合 300 以前時是系統極度不穩定的狀態，之後 RPSA 平均每場遊戲所擁有的回合數都是最大。RPSA 平均每場遊戲所擁有的回合數在回合數總合在約 1,700 時達到約為 7.5，之後就一直下降，直到回合數總合約 3,800 趨於穩定，此時 RPSA 平均每場遊戲所擁有的回合數約為 6.5。而 FPSA 在回合數總合 600 以前呈現下降的趨勢不過之後的趨向於 RPSA，不過 FPSA 平均每場遊戲所擁有的回合數約比 RPSA 少 0.5 左右。OPSA 平均每場遊戲所擁有的回合數在回合數總合 600 之後都是最低，而且到回合數總合 1,500 之後就急速下降，在回合數總合 4,200 時已經下降到 4 以下。到最後 OPSA 平均每場遊戲所擁有的回合數和 FPSA 相差 2.4 左右，代表著 OPSA 每場遊戲所擁有的回合數與 RPSA 和 FPSA 相較之下少了許多。

從圖 9、圖 10 和圖 11 得知 OPSA 在相同回合數總合的情形下產生的

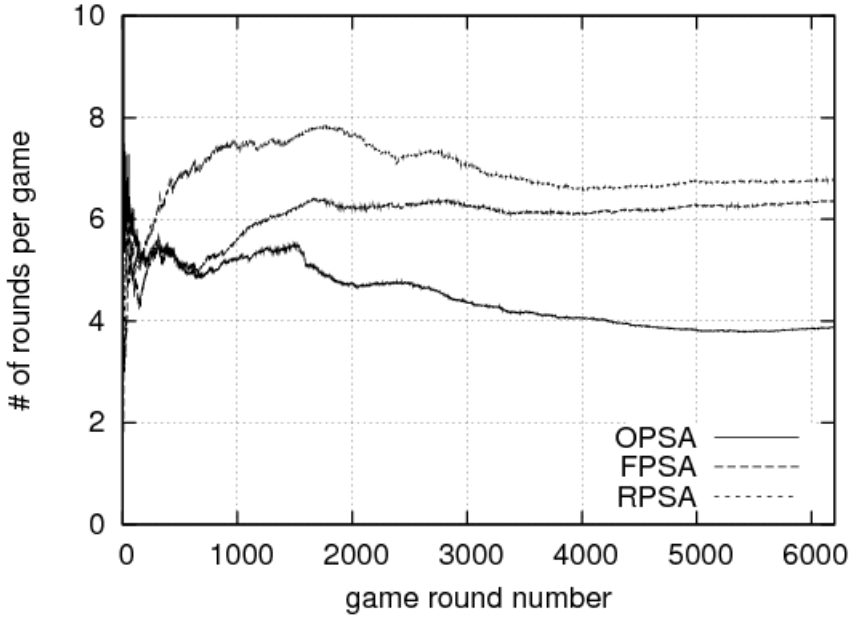

圖 11 三種演算法平均一場遊戲所擁有有的回合數和回合數總合的關係

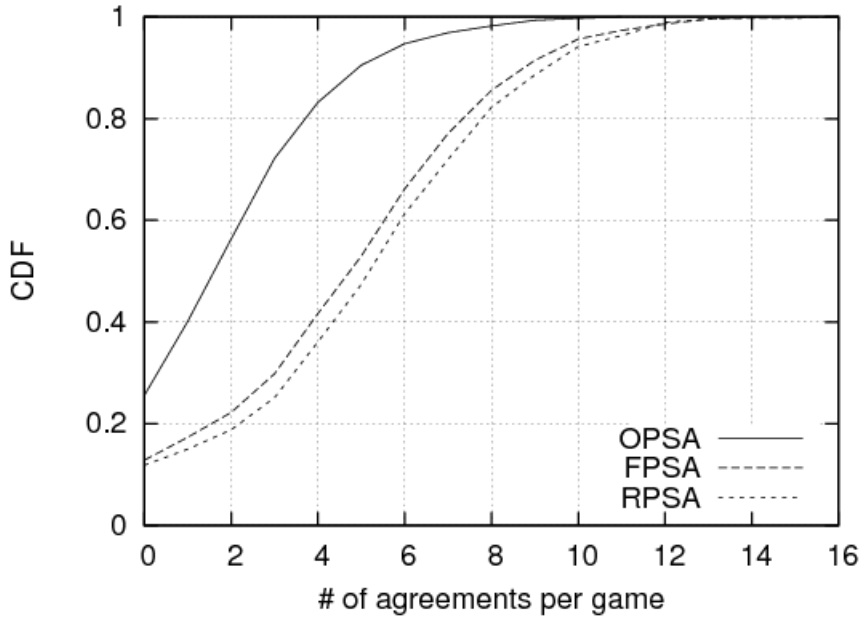

圖 12 三種演算法每場遊戲所產生 agreement 個數的分佈

agreement 個數為最少，而且 OPSA 平均每場遊戲所擁有的回合數為最少，這代表 OPSA 每場遊戲所產生的 agreement 個數也會比較少。圖 12 表示三種演算法每場遊戲平均產生多少個 agreement，OPSA、FPSA 和 RPSA 有著相同形式的分佈。不過 RPSA 每場遊戲平均產生約 5 個 agreement，FPSA 每場遊戲平均產生約 4.5 個 agreement，而 OPSA 每場遊戲平均只有產生約 1.5 個 agreement。其中 OPSA 一場遊戲沒有產生任何 agreement 的機率約為 0.25，RPSA 為 0.1 左右，相差 2.5 倍左右。OPSA 每場遊戲平均產生的 agreement 個數和 RPSA 相差大約 3.5 個，代表 OPSA 每場遊戲很難產生 agreement。

第二節　行為分析

從上一節實驗的結果中觀察 OPSA 的標記成長速率比 FPSA 和 RPSA 要來得低，OPSA 一場遊戲擁有的回合數也比 FPSA 和 RPSA 要來得少。為了得知

為何 OPSA 會有這些現象，所以必須詳細觀察每場遊戲中每回合到底發生什麼事情,這有助於了解 OPSA 為何會產生和其餘二種圖片選擇演算法不同的現象。

圖 13 表示三種演算法回合數總合和每回合被 pass 機率的關係。除了回合數總合在 700 前系統還不是穩定狀態之外，之後 FPSA 和 RPSA 被 pass 的機率趨於穩定，且行為幾乎類似，值大約是 0.1 左右。不過 OPSA 每回合被 pass 的機率高達 0.3 左右，這幾乎是 FPSA 和 RPSA 的三倍，並且還有上升的趨勢。這代表 OPSA 可能有某種特殊的原因造成 OPSA 每回合被 pass 的機率是 FPSA 和 RPSA 的 3 倍。

為了得知 OPSA 遊戲中的回合被的 pass 機率為何那麼高，所以必須了解玩家在每一回合中的行為。圖 14 表示每回合成功得到一個標記或是被 pass 所花時間的機率累積分佈圖。圖中顯示大約在 8 秒內此回合被 pass 的機率要比 8 秒

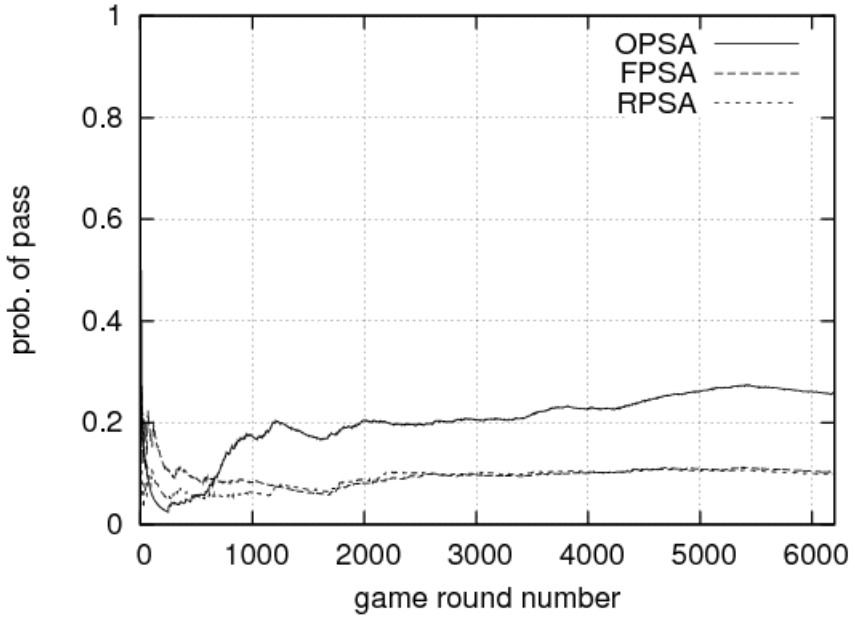

圖 13 三種演算法每回合被 pass 的機率和回合數總合的關係

內此回合得到一個標記的機率還要大，例如在 5 秒內此回合被 pass 的機率大約是 0.12，而在 5 秒內此回合得到一個標記的機率大約是 0.04，相差 3 倍左右；反之，超過 8 秒則機率相反，超過 8 秒內此回合得到一個標記的機率比超過 8 秒內此回合被 pass 的機率還要高，例如在 30 秒內此回合得到一個標記的機率大約是 0.9，而在 30 秒內此回合被 pass 的機率大約是 0.7，相差 1.3 倍左右。這代表玩家有比較大的可能是看到一張圖在 8 秒內就很快的 pass 這張圖或是一直在這回合花許多時間而得到一個標記。

有玩家在一回合中的行為，直覺地聯想到 OPSA 與 FPSA 和 RPSA 最大的不同點在於 OPSA 會盡力地將每張被標記過的圖片的標記數超過 r，代表著系統秀給玩家進行標記的圖片都是被標記過的圖片，也就是一張圖片的 taboo word 數大於 0。圖 15 表示 r 和回合數總合的關係，在系統運行一個月左右，回

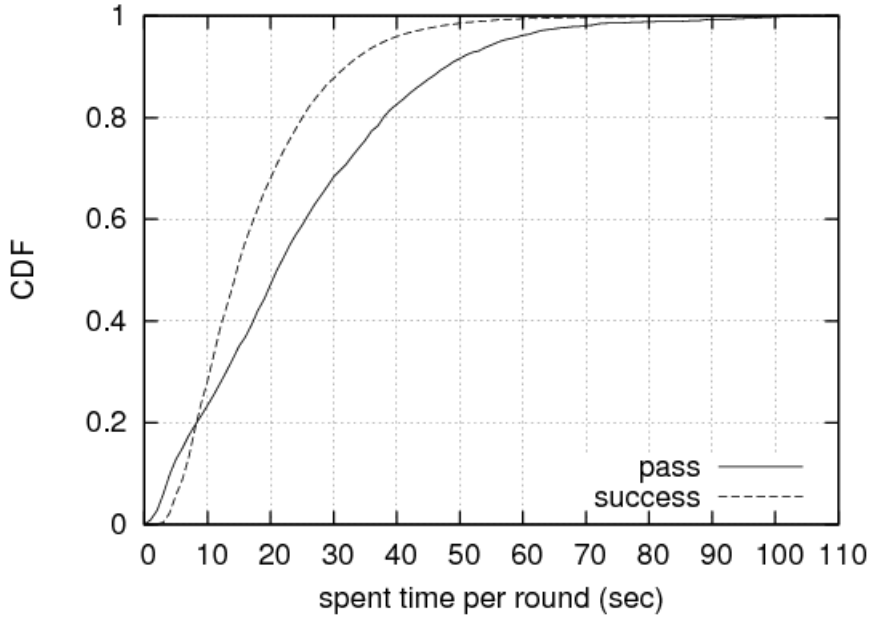

圖 14 此回合成功得到一個 agreement 或被 pass 所花時間的分佈

合數約有 6,000 的情形下，*r* 成長到 5.5 左右，代表著在 OPSA 中平均每張被標記過的圖片要有 5.5 個標記。回合數總合約 120 時 *r* 超過 1，約 550 時 *r* 超過 2，約 1,480 時 *r* 超過 3，約 2,927 時 *r* 超過 4，約 5,000 時 *r* 超過 5。這現象和圖 2 中 *T* 和 *r* 的關係類似，*r* 愈大則 *r* 每增加 1 所需要的回合數總合也要愈多。所以 OPSA 在遊戲時大部分都是秀出約有 5 個 taboo word 的圖片，所以 OPSA 的特殊現象可能跟玩家看到 taboo word 的數量有關。

在發現 OPSA 的特殊現象可能和 *r* 有關之後，也就是 OPSA 大部分時間會讓系統秀出有被標記過的圖。欲觀察玩家看到標記時的反應，首先必須得知一張圖片被標記的個數和此回合要得到一個 agreement 所花時間的關係。圖 16 描述這個關係，當圖片沒有任何標記時平均花 16 秒左右就能得到一個 agreement，當圖片有 1 標記時平均花 20 秒左右就能得到一個 agreement，當圖

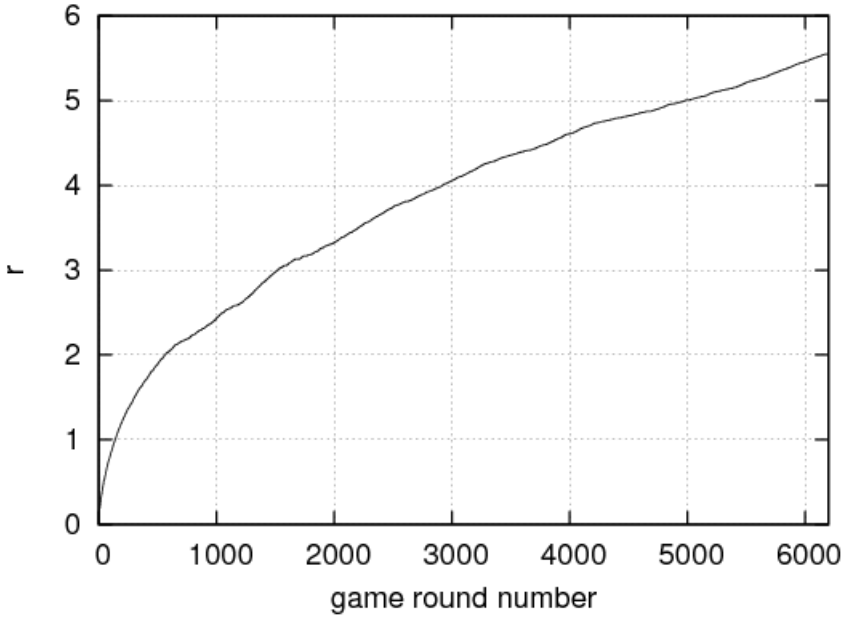

圖 15 OPSA 的 *r* 和回合數總合的關係

片有 2 標記時平均花 22 秒左右就能得到一個 agreement，當圖片有 3 標記時平均花 25 秒左右就能得到一個 agreement，當圖片有 4 標記時平均花 28 秒左右就能得到一個 agreement，當圖片有 5 標記時平均花 28.5 秒左右就能得到一個 agreement。沒有被標記過的圖片和有 5 個標記的圖片要得到一個 agreement 所花的時間相差約 12 秒，大約占遊戲時間的十分之一。由此可知一張擁有愈多標記個數的圖片則玩家所花在得到一個 agreement 的時間也愈多。這也說明 OPSA 每場遊戲所擁有的回合數比較少的原因。

圖 17 描述被標記個數和此回合被 pass 的機率關係。當圖片沒有任何標記的時候此回合被 pass 的機率約為 0.1，這與 FPSA 和 RPSA 每回合被 pass 的機率相近。當有 1 個標記時此回合被 pass 的機率約為 0.15，有 2 個標記時此回合被 pass 的機率約為 0.25，有 3 個標記時此回合被 pass 的機率約為 0.26，有 4

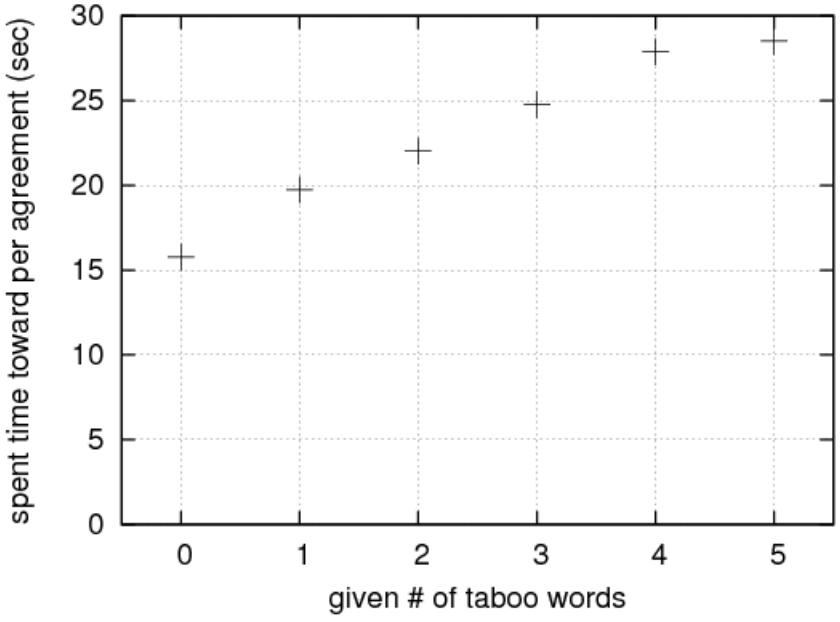

圖 16 平均此回合得到一個 agreement 所需要的時間和被標記個數的關係

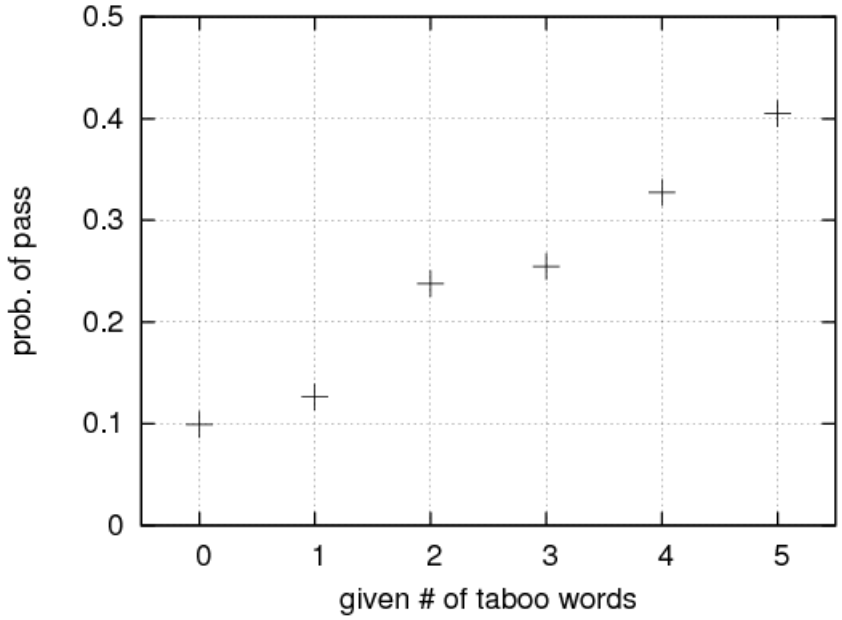

圖 17 此回合被 pass 的機率和被標記個數的關係

個標記時此回合被 pass 的機率約為 0.34，有 5 個標記時此回合被 pass 的機率約為 0.42。這代表玩家看到愈多標記則此回合被 pass 的機率就愈高，這是因為玩家想要在一場遊戲中得到較高的分數，有些玩家看到一張圖有許多的標記就不願意花太多的時間在得到一個 agreement 上，所以玩家傾向 pass 此回合以節省時間。這和圖 16 也說明為何 OPSA 每場遊戲所產生的 agreement 數會那麼少的原因。

在本節中得知 OPSA 有著和其餘二種圖片選擇演算法不同的行為是因為 OPSA 容易將已經被標記過的圖片交給玩家進行標記。不過玩家看到被標記過的圖片傾向於馬上 pass 這張圖片或是花了許多時間得到一個 agreement。由此可知 OPSA 的行為和玩家的行為有著很大的關係。

第三節　效能分析

在前面實驗的分析中得知 OPSA 的行為和玩家的行為有著相當大的關係，在此節中將討論三種演算法效能的表現，並且透過效能測量函數中「質」和「量」的角度觀察此三種演算法。在效能分析的方面，和效能相關的有兩個變數，$\overline{S}*$ 和 N 。這兩個變數分別代表系統的「質」和「量」。其中 $\overline{S}*$ 代表「質」，N 代表「量」。圖 18 顯示三種演算法被標記過圖片數量和回合數總合的關係，也就是三種演算法之間「量」的關係。在圖中可以發現 FPSA 和 RPSA 被標記過圖片的數量呈現線性的關係，且 FPSA 和 RPSA 有一樣的趨勢。而 OPSA 始終為最低，OPSA 的被標記過圖片數量大約在 500 之後就成長緩慢。RPSA 和 FPSA 的 N 相近是因為系統中圖片數太多且系統進行得不夠久才會有此現象。由此可知 OPSA 不像 FPSA 只注重「量」的圖片選擇演算法。

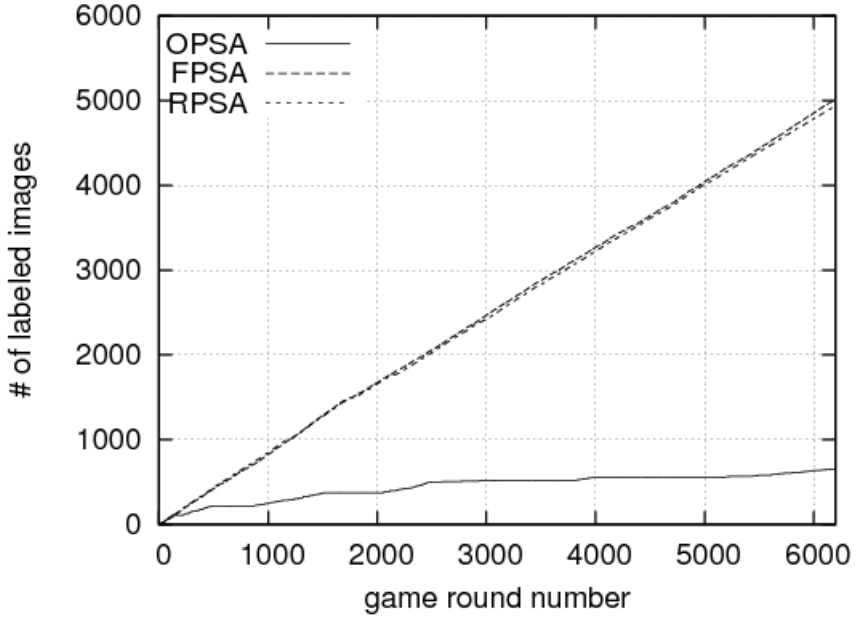

圖 18 三種演算法的被標記圖片數量 N 和回合數總合的關係

圖 19 為三種演算法中被標記過圖片所擁有標記數的分佈。OPSA 被標記過圖片的標記個數集中在 5 和 6，RPSA 幾乎都是 1 少數是 2 而 FPSA 全部都是 1。會造成這種情況是因為 OPSA 會盡量將被標記過圖片的標記數上升超過 r ，所以 OPSA 會將擁有 1 到 r 個標記的圖片優先送給玩家進行遊戲，這和設計 OPSA 時的概念一致。

圖 20 表示三種演算法的平均被標記圖片數量擁有的分數和回合數總合的關係，這是為了觀察三種演算法之間「質」的關係。被標記圖片數量擁有的分數代表了「質」，圖中可以看到 FPSA 和 RPSA 的行為幾乎一樣都維持在 120 左右，由於平均每個標記的分數約為 120，所以 FPSA 和 RPSA 中所被標記的圖片幾乎只擁有一個標記，顯示了 FPSA 極端的重「量」不重「質」。圖中 RPSA 的行為和 FPSA 幾乎一樣，這是因為系統中有太多圖片，而且系統運行得不夠

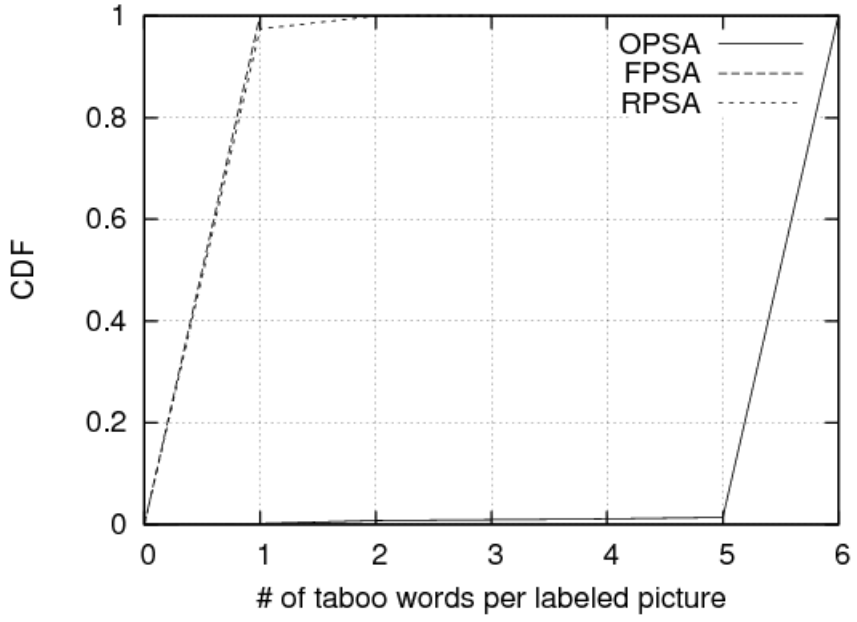

圖 19 被標記過圖片所擁有標記數的分佈

久的緣故。圖中 OPSA 從回合數 150 左右就遠遠超過 FPSA 和 RPSA，代表著 OPSA 不只注重在「量」的發展，也注重在「質」的發展。另外在圖中可以觀察到 OPSA 回合數 1000、2000、4000 等處有起伏的現象，這是因為在這些時候 OPSA 的圖片集合 P_1 中沒有圖片，所以系統會選擇沒有被標記過的圖片給玩家進行遊戲，所以被標記過圖片的數目就會增加，相對平均被標記圖片數量擁有的分數就會減少，所以會造成這些起伏的現象。

最後圖 21 顯示三種演算法的效能和回合數總合的關係。圖中可以發現 FPSA 和 RPSA 的效能差不多，而 OPSA 的效能比 FPSA 和 RPSA 來得高。不過在第四章模擬的結果 RPSA 的效能比 FPSA 來得高，這是因為目前圖片太多且系統進行得不夠久所以才會有這種情形。OPSA 的效能比 FPSA 的效能來得好是因為 FPSA 只注重「量」而不重「質」。OPSA 的效能比 RPSA 的效能來得

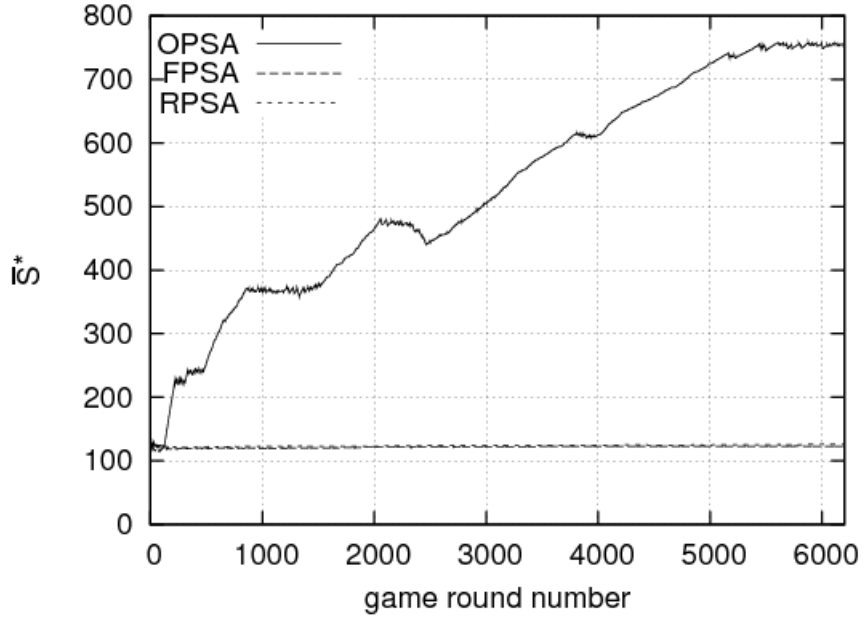

圖 20 三種演算法的平均每張被標記圖片擁有的分數 $\bar{S}^*$ 和回合數的關係

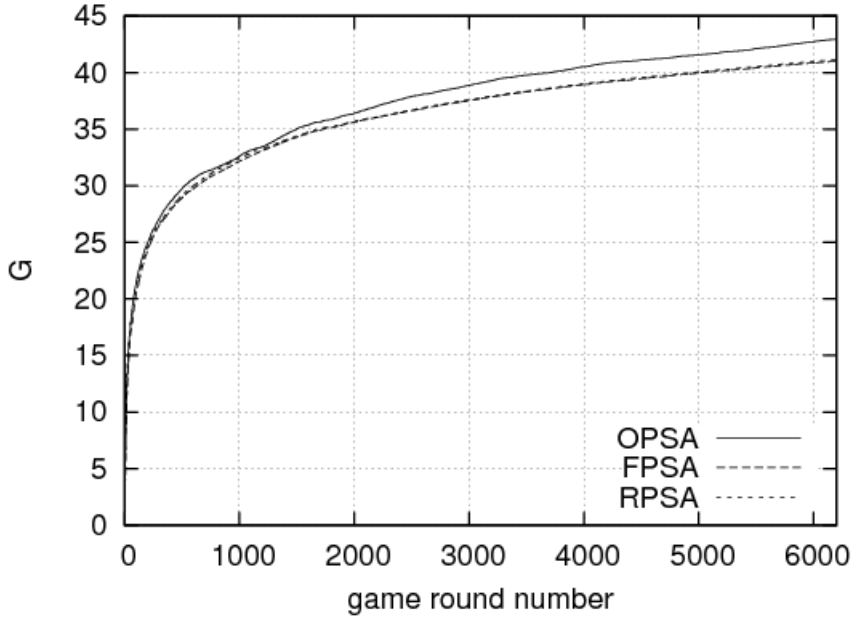

圖 21 三種演算法的系統效能和回合數總合的關係

好是因為 OPSA 是在「質」和「量」之間取得平衡的圖片選擇演算法，所以 OPSA 的效能會比 RPSA 和 FPSA 還要好。本章說明在實際的實驗中證實根據第三章分析所創造的圖片選擇演算法 OPSA 不只在模擬中能夠增加 ESP 遊戲系統的效能，在實際系統中 OPSA 也能夠增加 ESP 遊戲系統的效能。

第七章　結論與未來工作

此篇論文探討如何藉由適當的遊戲策略增加 GWAP 系統的效能。我們使用 ESP 遊戲為例，使用數學分析的方法，定義一個能夠同時考量「質」和「量」的系統效能評估依據，並且依據我們分析的結果，設計出能夠增加 ESP 遊戲系統效能的圖片選擇演算法 OPSA。同時，為了驗證 OPSA 在真實系統上之效能，我們設計並實作 ESP Lite 進行實際的驗證。實驗的過程中發現 OPSA 的結果和玩家的行為是有著相當大的關聯性，且 OPSA 確實能夠增加 ESP 遊戲系統的效能。有此可知有策略地運行 GWAP 系統確實能夠改善系統的效能。

在未來可以改進和延伸的目標上，大致有四個方面分別依序如下所示：

一、更詳細且深入的數學分析

在第三章數學模型建構和分析的部分，由於為了簡化系統起見，並沒有考慮每回合所花的時間，也沒有定義每回合玩家決定 pass 的機率。若是在分析時考慮此二項因素，則在數學分析的部分能夠更加完整。

二、針對個別圖片之 r 值設定

在目前的系統中，所有圖片都採用相同的 r 值，但在實際的系統中，有些圖片包含的資訊可能有超過 r 個單字可以描述，因此無法將圖片中所有資訊都取出來；或是少於 r 個單字可以描述，因此無論此張圖片被玩多

少次都不會擁有超過 r 個標記，造成計算資源白白地被浪費。所以能夠根據每張圖片設計出各自獨立的 r 是日後所要發展的目標。

三、遊戲玩家選擇策略研究

目前系統正在與身分認證系統結合，並且透過個別玩家的遊戲紀錄得知玩家的專長。所以系統根據玩家的背景資料配對擁有相同背景的玩家進行遊戲，像是擁有相同性別或是學歷、興趣、國籍、語言等，較隨機配對玩家進行遊戲更容易取出圖片中較難取出的資訊，藉此增加系統的效能。

四、系統模組化

目前 ESP Lite 是針對 ESP 遊戲進行設計，並且採用固定網頁的方式呈現，我們計劃持續改良系統程式碼，增加其對於模組化設計的支援，以大幅增加系統的彈性，並且透過開放原始碼的方式，讓更多使用者可以針對自身的需求，修改特定的模組即可快速建立一個新的 GWAP 系統。如此一來，對於日後 GWAP 相關的系統開發可以節省許多開發的時間和人力資源。

參考文獻

[1] Amazon Mechanical Turk. https://www.mturk.com/

[2] Brown Corpus Manual. http://khnt.aksis.uib.no/icame/manuals/brown/

[3] Distributed Proofreaders. http://www.pgdp.net/

[4] edict virtual language centre. http://www.edict.com.hk/textanalyser/wordlists.htm

[5] ESP Game Dataset. http://www.cs.cmu.edu/~biglou/resources/

[6] Flickr. http://www.flickr.com/

[7] Fotki. http://www.fotki.com/

[8] Google Image Labeler. http://images.google.com/imagelabeler/

[9] Google Image Search. http://images.google.com/

[10] gwap.com. http://www.gwap.com/gwap/

[11] Human-based computation – Wikipedia. http://en.wikipedia.org/wiki/Human-based_computation

[12] Project Gutenberg. http://www.gutenberg.org/

[13] Porter Stemming Algorithm. http://tartarus.org/~martin/PorterStemmer/

[14] Vipul's Razor. http://razor.sourceforge.net/

[15] Wikipedia. http://en.wikipedia.org/

[16] Yahoo! Answers. http://answers.yahoo.com/

[17] Marek Bell, Stuart Reeves, Barry Brown, and Scott Sherwood. *Eyespy: supporting navigation through Play.* ACM Conference on Human Factors in Computing Systems (CHI), 2009.

[18] Jeffrey P. Bigham, Ryan S. Kaminsky, Richard E. Ladner, Oscar M. Danielsson, and Gordon L. Hempton. *WebInSight: making web images accessible.* The Eighth International ACM SIGACCESS Conference on Computers and Accessibility (ASSET), 2006.

[19] Sean Casey, Ben Kirman, and Duncan Rowland, The Gopher Game: *A social, mobile, locative game with user generated content and peer review.* The International Conference on Advances in Computer Entertainment Technology (ACE), 2007.

[20] Tsung-Hsiang Chang, Chien-Ju Ho and, Jane Yung-jen Hsu. *The PhotoSlap game: play to annotate.* The Twenty-Second Conference on Artificial Intelligence (AAAI), 2007.

[21] Bruno Norberto da Silva and Ana Cristina Bicharra Garcia. *KA-CAPTCHA: an opportunity for knowledge acquisition on the web.* The Twenty-Second Conference on Artificial Intelligence (AAAI), 2007.

[22] Craig Gentry, Zulfikar Ramzan, and Stuart Stubblebine. *Secure distributed Human Computation.* The Sixth ACM Conference on Electronic Commerce (EC), 2005.

[23] Lyndsay Grant, Hans Daanen, Steve Benford, Alastair Hampshire, Adam Drozd, and Chris Greenhalgh. *MobiMissions: the game of missions for mobile phones.* The 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques(SIGGRAPH), 2007.

[24] Severin Hacker and Luis von Ahn. *Matchin: eliciting user preferences with an online game.* ACM Conference on Human Factors in Computing Systems (CHI), 2009.

[25] Chien-Ju Ho, Tsung-Hsiang Chang and, Jane Yung-jen Hsu. *PhotoSlap: a multi-player online game for semantic annotation.* The Twenty-Second Conference on Artificial Intelligence (AAAI), 2007.

[26] Shaili Jain and David C. Parkes. *A game-theoretic analysis of games with a purpose.* The 4th International Workshop On Internet And Network Economics (WINE), 2008.

[27] Edith L. M. Law, Luis von Ahn, Roger B. Dannenberg, and Mike Crawford. *TagATune: A game for music and sound annotation.* The 8th International Conference on Music Information Retrieval (ISMIR), 2007.

[28] Edith Law and Luis von Ahn. *Input-Agreement: A new mechanism for collecting data using human computation games.* ACM Conference on Human Factors in Computing Systems (CHI), 2009.

[29] Henry Lieberman, Dustin A Smith, and Alea Teeters. *Common Consensus: a webbased game for collecting commonsense goals.* The Workshop on Common Sense and Intelligent User Interfaces (CSIUI), 2007.

[30] Michael Mandel and Daniel Ellis. *A web-based game for collecting music metadata.* Journal of New Music Research (JNMR), vol. 37, pp. 151-165, June 2008.

[31] Sebastian Matyas. *Playful geospatial data acquisition by location-based gaming communities.* The International Journal of Virtual Reality (IJVR), vol. 6, no. 3, pp. 1-10, September 2007.

[32] Sebastian Matyas, Christian Matyas, Christoph Schlieder, Peter Kiefer, Hiroko Mitarai, and Maiko Kamata. *Designing location-based mobile games with a purpose - collecting geospatial data with CityExplorer.* International Conference on Advances in Computer Entertainment Technology (ACE), 2008.

[33] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. *LabelMe: A database and web-based tool for image annotation.* International Journal of Computer Vision (IJCV), vol. 77, no. 1-3, pp. 157-173, May 2008.

[34] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. *Get another label? Improving data quality and data mining using multiple, noisy labelers.* The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2008.

[35] Pradeep Shenoy and Desney S. Tan. *Human-aided computing: utilizing implicit human processing to classify images.* ACM Conference on Human Factors in Computing Systems (CHI), 2008.

[36] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. *Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks.* Conference on Empirical Methods in Natural Language Processing (EMNLP), 2008.

[37] Qi Su, Dmitry Pavlov, Jyh-Herng Chow, and Wendell C. Baker. *Internet-scale collection of human-reviewd data.* The 16th International World Wide Web Conference (WWW), 2007.

[38] Douglas Turnbull, Ruoran Liu, Luke Barrington, and Gert Lanckriet. *A Game-Based Approach for Collecting Semantic Music Annotations.* The 8th International Conference on Music Information Retrieval (ISMIR), 2007.

[39] Luis von Ahn. *Games with a purpose.* IEEE Computer Magazine, pp 96-98, June 2006.

[40] Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford. *CAPTCHA: using hard AI problems for security.* Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2003.

[41] Luis von Ahn and Laura Dabbish. *Labeling images with a computer game.* ACM Conference on Human Factors in Computing Systems (CHI), 2004.

[42] Luis von Ahn and Laura Dabbish. *Designing games with a purpose.* Communications of the ACM, vol. 51, no. 8, pp. 58-67, August 2008.

[43] Luis von Ahn, Shiry Ginosar, Mihir Kedia, and Manuel Blum. *Improving image search with PHETCH.* The 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2007.

[44] Luis von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. *Improving accessibility of the web with a computer game.* ACM Conference on Human Factors in Computing Systems (CHI), 2006.

[45] Luis von Ahn, Mihir Kedia, and Manuel Blum. *Verbosity: A game for collecting common-sense facts.* ACM Conference on Human Factors in Computing Systems (CHI), 2006.

[46] Luis von Ahn, Ruoran Liu, and Manuel Blum. *Peekaboom: A game for locating objects in images.* ACM Conference on Human Factors in Computing Systems (CHI), 2006.

[47] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. *reCAPTCHA: human-based character recognition via web security measures.* Science, vol. 321, pp. 1465-1468, 12 Septemper 2008.

[48] Ingmar Weber, Stephen Robertson, and Milan Vojnović. *Rethinking the ESP game.* Technical Report MSR-TR-2008-132, Microsoft Research, 2008.

附錄　符號對照

本章列出本論文中所使用的符號和其所表示的意義。

$E[S]$	系統中每個 agreement 分數的期望值
f_i	w_i 所對應的頻率
G	測量系統效能的函數
k	系統的分數表中有 k 個分數等級
L_i	分數表中第 i 個單字 w_i 所對應的分數等級，值域為 $\{0,1,\cdots,k-2\}$
M	系統中圖片的數量
N	系統中被標記過圖片的數量
P	系統中所有圖片的集合
P_0	系統中沒有標記圖片的集合
P_1	系統中有 1 到 r 個標記圖片的集合
P_2	系統中超過 r 個標記圖片的集合
r	系統中每張被標記過圖片所擁有的標記個數
S	隨機變數，代表每個 agreement 所得到的分數
$\overline{S}$	系統中平均每個 agreement 的分數
$\overline{S}^*$	系統中平均每張被標記過圖片所擁有的分數
S_i	隨機變數 S 的實現值
S_{base}	系統得分系統中分數的起始值
S_{offset}	系統得分系統中每個分數等級的分數間隔
$score(\cdot)$	函數，輸出所對應的分數
$stem(\cdot)$	函數，輸出經過 Porter Stemming Algorithm 處理的結果
T	系統中的回合數總合
w_i	系統的分數表中第 i 個單字