首頁保留頁面

目錄

[一、 研究計劃背景 1](#_Toc470711797)

[二、 重要參考文獻之評述 2](#_Toc470711798)

[三、 研究方法與進行步驟 3](#_Toc470711799)

[3.1 將遊戲的任務與空間分離 3](#_Toc470711800)

[3.1.1 任務圖 4](#_Toc470711801)

[3.1.2 空間圖 4](#_Toc470711802)

[3.2 使用改寫系統來操作語法模型的轉換 5](#_Toc470711803)

[3.2.1 以圖形語法來生成任務 6](#_Toc470711804)

[3.2.2 鎖與鑰匙機制 7](#_Toc470711805)

[3.2.3 以形狀語法來生成空間 8](#_Toc470711806)

[3.2.4 基於任務圖與空間語法來生成同構空間 8](#_Toc470711807)

[3.3 套用 Machinations 框架，拓展至遊戲機制 9](#_Toc470711808)

[3.3.1 Machination 框架 9](#_Toc470711809)

[3.3.2 遊戲中的反饋結構促使系統穩定 10](#_Toc470711810)

[3.3.3 基於 Machinations 的改寫系統 11](#_Toc470711811)

[3.4 預計可能遭遇之困難及解決途徑 12](#_Toc470711812)

[3.4.1 複雜系統的呈現與關卡的評估問題 12](#_Toc470711813)

[3.4.2 3D 空間的呈現問題 12](#_Toc470711814)

[3.4.3 使用者操作介面設計 12](#_Toc470711815)

[四、 預期完成之工作項目及成果 14](#_Toc470711816)

[4.1 預期完成之工作項目 14](#_Toc470711817)

[4.2 預期完成之研究成果 14](#_Toc470711818)

[4.3 對於學術研究、國家發展及其他應用方面預期之貢獻 14](#_Toc470711819)

[4.4 參與人員，預期可獲得之訓練 14](#_Toc470711820)

[五、 執行進度 16](#_Toc470711821)

[六、 參考資料 17](#_Toc470711822)

**圖片目錄**

[圖一、自動化關卡生成工具之框架 4](#_Toc470707536)

[圖二、Joris Dormans 提出的關卡生成步驟 5](#_Toc470707537)

[圖三、基本任務圖之範例 5](#_Toc470707538)

[圖四、基本空間圖之範例 6](#_Toc470707539)

[圖五、改寫系統用途之一，將任務模型轉換空間模型 7](#_Toc470707540)

[圖六、圖形語法的範例 7](#_Toc470707541)

[圖七、改寫系統用途之二，將任務逐一增添完整內容 8](#_Toc470707542)

[圖八、分支結構的任務圖，e=入口、T=小任務、K=鑰匙、L=鎖、g=終點 9](#_Toc470707543)

[圖九、形狀語法的規則範例 9](#_Toc470707544)

[圖十、能夠將任務終端符號轉為空間規則的建造指令 10](#_Toc470707545)

[圖十一、Machinations 流程圖，應用於遊戲「Pac-Man」之範例 11](#_Toc470707546)

[圖十二、於遊戲「Monopoly」中，以 Machinations 框架所表示的反饋結構 12](#_Toc470707547)

[圖十三、基於 Machinations 的模型轉換應用於週期性結構之釋例 12](#_Toc470707548)

[圖十四、Machinations 經過多層次的改寫系統，能夠以更複雜的機制呈現關卡 13](#_Toc470707549)

1. 研究計劃背景

本計劃預計研發一套通用型遊戲關卡生成編輯器，利用語言學領域的生成語法 (generative grammars)，以計算機科學迭代的方式部分自動化關卡生成。希冀增進關卡設計師的關卡設計速度，並同時提升關卡內容的豐富度與耐玩性。

關鍵字：程序化內容生成、遊戲關卡生成

為了降低遊戲的開發成本，近年業界不斷地追尋程序化內容生成之技術（以下簡稱 PCG）。希望運用 PCG 技術，進行自動且大量地生成可遊玩的關卡。可遊玩的遊戲關卡固然是最低的需求，若是僅以可遊玩為目標的話，現今已存在大量的程式與產品達成目標。倘若兼顧品質與數量，那麼符合玩家遊戲體驗期待的 PCG 方法就顯得相當稀少、甚至只能針對某些既有的遊戲產品。而我們的目標便是建置符合遊戲內容需求的地圖生成軟體，一方面既能符合任務設計邏輯，另一方面亦能生成遊戲地圖。

|  |
| --- |
| 草稿區  可以做為研究題目的幾項參考選項：   1. 以 Evolutionary algorithms 涉入改寫系統的語法建立   —「概念源自於 paper 1 section 8」  語法的品質會是影響遊戲品質的關鍵要素，而語法的建立多會需要專業的遊戲設計師涉入。   1. 涉及玩家的遊玩表現，動態生成關卡   —「概念源自於 paper 1 section 7」  最好的策略是在構築空間之前，生成一個仍有一些非終端節點的任務，而替換非終端節點的動作應在遊玩過程中進行，並且應該根據玩家的遊玩表現直接或間接地調整。   1. 除了生成有意義的地圖做為目標外，能否評估地圖的困難度及有趣性 2. under construction 尚有其他選項，整理中 |

1. 重要參考文獻之評述

Zook 和 Riedl 提出「生成與測試」的策略來自動生成遊戲機制；先行生成符合設計需求的遊戲機制，然後測試該機制以確保遊戲的可玩性 [1]。Zook 和 Riedl 將遊戲的實體，然後利用 Constraint Solver回答集編程 (Answer Set Programming)，根據需求來找出需要的轉移模式 [2]。並以 AI 來測試可玩性，能夠有效地找出存在與無效的遊玩路線。

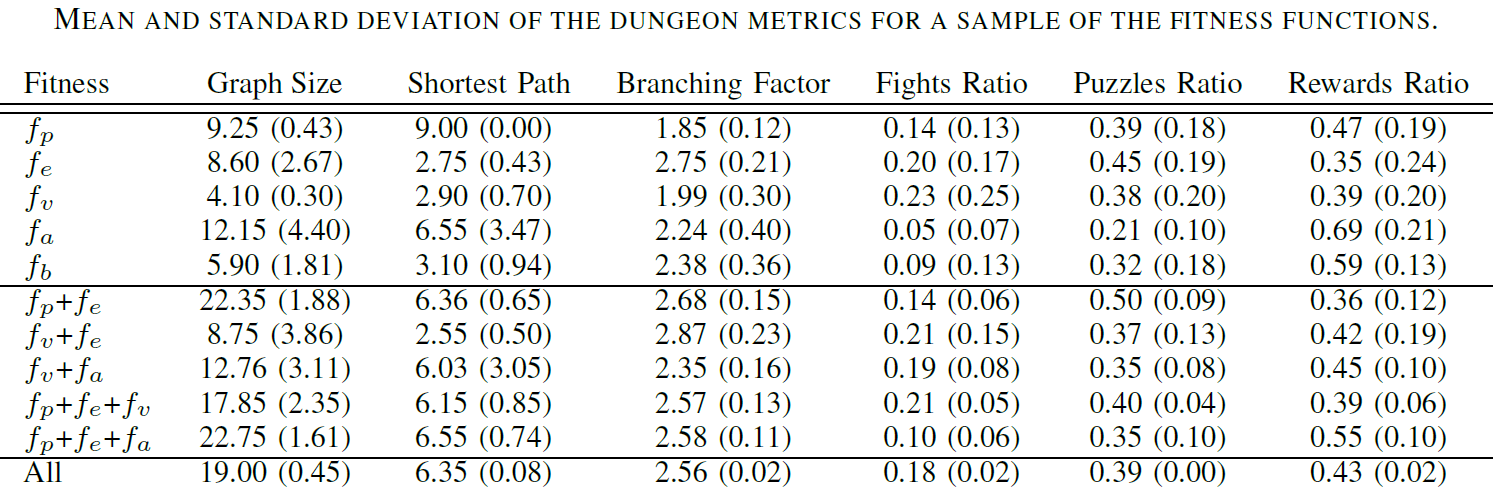
Karavolos 等人提出 search-based 的生成方法，利用演進玩家的預期動作來創建關卡 [3]。因此，任務圖中的節點都代表玩家的抽象動作，如解一道謎題、進行一場戰鬥等。他們參考了……. <下段接續>

==== 以下尚未編輯 ====

參考design pattern (Bj¨ork & Holopainen, 2014)與數學公式 (Liapis, Yannakakis, & Togelius, 2013)後，提出了幾個fitness functions來協助mission graph的進化。這篇研究主要講述那些fitness functions的組合對mission graph有甚麼樣的影響以及多樣性。

本篇論文提供了一個fitness的表格，如表格 1，可以讓我們知道每項metrics產生的平均值與標準差。不同的組合都會使得mission graph產生不同的變化，進而有更多的多樣性。但是，因mission graph中裡面的node都是抽象的行為，因此還需要經過grammar來轉換成特定的動作 (這部分參考了 (Karavolos, Bouwer, & Bidarra, 2015)的mixed-initiative grammar system)。所以，容易創建出許多的”空”房間。

表格 1 fitness的表格



Lavender和Thompson (Lavender & Thompson)擴展mission and space grammars (Dormans, 2010)的研究，提出一種生成2D動作冒險遊戲(The Legend of Zelda)地圖的方法，類似經典任天堂系列＂The Legend of Zelda＂中的地圖。地圖是通過兩階段性的方法生成；在mission graph完成之後，作者基於constraint-based 的方法去建構能夠忠實代表Mission 結構的幾何空間(關卡)。本篇論文利用Metrics (Smith & Whitehead, 2010)來衡量所生成的地圖是否可行(feasibility)。然而，依然有一定的機率產生互相不連通的房間地圖，所以需要訂定好sample grammar。此外，目前該論文的生成器有一個主要限制，是它依賴Solarus引擎內的手動製造房間。

1. 研究方法與進行步驟

我們根據 [4] [5] 的提出的關卡生成方法加以改進並將其實作。

＜still under construction.＞

|  |
| --- |
| 圖一、自動化關卡生成工具之框架 |

1. 將遊戲的任務與空間分離

Joris Dormans 認為一個完整的關卡需要包含任務與空間二者 [6]，便需要有一特定的空間佈局，及一系列需要於此空間中被執行的任務。關卡任務代表玩家需要按照任務流程，來依序挑戰才能夠完成該關卡；關卡的空間由其地理佈局所組成，或者由與地圖相似的節點網絡所構成。由於任務與空間之間的交錯混雜，導致關卡設計者最終採取簡單卻有效的策略，也就是讓任務與空間同構。同構雖然在設計上不是唯一的選擇，但對於某些遊戲是非常合適的，特別是一具有線性的關卡設計。

而 Joris Dormans 亦提出了一種自動關卡設計的方法，藉由產生一個任務，在利用這個任務去產生適合此任務的空間 [4] [5] [6]。舉例來說，關卡設計者透過生成任務的介面來建立任務圖，玩家必須執行這些任務才能夠完成關卡，接下來將任務轉換為空間，並將任務依序安排至該空間中。設計者接著在地圖添加更細節的內容，直到地圖充滿任務的要素並作為遊戲的關卡。

|  |
| --- |
| 圖二、Joris Dormans 提出的關卡生成步驟 |

1. 任務圖

任務圖注重於任務與玩家的相互關係，表現出玩家距離通關的進度狀況。主要由兩種要件：「節點」和「連結線」構成，其中節點再細分為任務、起點與終點；有向連結線再依照兩節點之間的執行先後關聯，細分為薄弱條件、強烈條件與抑制。由於強烈條件或抑制的關係，某些節點會無法執行。

|  |
| --- |
| 圖三、基本任務圖之範例 |

1. 空間圖

空間圖直接呈現了關卡的空間結構，且大多數的節點能夠直接表示出玩家目前所在位置。空間圖中的任何節點能透過顏色、字母來表示不同類型。主要亦由兩種要件：「節點」和「連結線」所構成。節點細分為場所、鎖和遊戲元素所構成；有向連結線細分為通道、閥、窗、解鎖與上鎖等。

|  |
| --- |
| 圖四、基本空間圖之範例 |

1. 使用改寫系統來操作語法模型的轉換

一個模型能夠透過改寫系統來轉換成為另一模型。改寫系統由具有左側與右側的規則 (rules) 所組成。這些規則指定一符號集能夠被另一符號集所取代。而改寫規則當中所使用的符號，便是在遊戲中經常會出現一些具有代表性的物件、要素或任務目標等，用於編寫遊戲中的這些週期性結構 (recurrent construction)。

改寫系統的操作類似於生成語法中採用的規則，而生成語法通常對字串進行操作，但並不全然，下述幾種類型便是生成語法的幾種特殊形式：

* 構成任務的圖形語法 (graph grammars)。
* 構成空間的形狀語法 (shape grammars)。

二者能夠獨立生成出結果，不過建議能夠將改寫系統套用在任務圖上，使其能夠產生出空間圖。此處提及之任務圖和空間圖是經過改良後的版本，定義其規則時會有些微上的不同，但更能夠體現出遊戲的關卡結構。

|  |
| --- |
| 圖五、改寫系統用途之一，將任務模型轉換空間模型 |

1. 以圖形語法來生成任務

圖形語法由多個節點與邊所組成。圖 (graph) 比字串更利於表達任務結構，且也能夠用來表示空間。在圖形語法中，一個或多個節點與其相連邊會被取代成另一組由節點與邊所構成的新結構。圖形語法除了擴充、修改節點外，亦支援刪除節點的操作。

|  |
| --- |
| 圖六、圖形語法的範例 |

然而在生成任務的過程時，能設立多組不同功能導向的改寫系統。如下圖，第一道改寫系統將任務的數量擴充，第二道添加任務之間的關聯性以確保處於有趣的順序，第三道添加鎖與鑰匙機制來建立非線性的任務結構，最後一道則為添加額外的任務與獎勵。

|  |
| --- |
| 圖七、改寫系統用途之二，將任務逐一增添完整內容 |

1. 鎖與鑰匙機制

改寫系統可以被用於規範、編制關卡的設計原則。舉例來說，設計一套改寫系統，實作出在遊戲中常見的典型鎖與鑰匙機制。鎖與鑰匙是使用改寫系統的技術的重要例證，它們足以對一個特定的關卡任務，將其轉換成一個近似關卡空間的結構。且因為使用改寫系統的關係，便可以防止死結的問題衍生，死結意旨鑰匙出現在對應的鎖之後，產生無法開啟鎖的狀態。

鎖與鑰匙的概念讓關卡設計師能夠去設計一系列線性的任務，再經由改寫系統能夠將其轉換為等價分支結構的空間中。以下是經過改寫系統後，能夠映照至遊戲的實例之一。玩家在先前的任務單位中會遇到無法通行的鎖，於是尋覓其它通道，最後在擊敗關卡魔王後取得關鍵道具，進而聯想到該道具與鎖的相依關係。

|  |
| --- |
| 圖八、分支結構的任務圖，e=入口、T=小任務、K=鑰匙、L=鎖、g=終點 |

1. 以形狀語法來生成空間

形狀語法由三種符號所組成，分別為「牆」、「開放空間」與「連結處」。在此語法中，只有「連結處」的類別為非終端符號，它的特徵是帶有指向性三角形的正方形符號。而在規則右側中必須要有一個連接處的屬性為起始連結處，並以實心灰色表示之。形狀語法相較其它生成語法的不同點，規則左側僅只會有一個符號。以下範例圖中的符號表僅有牆與連結處，但通常我們在設計規則時，我們會先以多個四邊形相連的開放空間勾勒出空間形狀，再以牆、連結處包圍之。

|  |
| --- |
| 圖九、形狀語法的規則範例 |

1. 基於任務圖與空間語法來生成同構空間

從已生成的任務模型產生出空間，我們對形狀語法進行些微的調整，讓圖形語法的終端符號將同時作為形狀語法的建造指令。然而，為了實作出任務終端符號在形狀語法的功能，形狀語法的每項規則都必須與任務語法終端符號有所關聯。形狀語法會先搜索任務圖當中的下一個符號，接著尋找能夠實現該符號的空間規則，並基於其關聯權重隨機挑選出一個規則。當任務圖已被完整地轉換成空間後，空間語法會退回至正常的轉換步驟，並持續迭代至空間中的非終端符號皆被取代為終端符號。

|  |  |
| --- | --- |
| |  | | --- | | 圖十、能夠將任務終端符號轉為空間規則的建造指令 | |

1. 套用 Machinations 框架，拓展至遊戲機制

有別於前述章節的 Mission / Space 框架，Machinations 框架有助於視覺化與理解遊戲的內部經濟體系，而這裡提到的經濟並非狹義地泛指金錢，而是廣義地指遊戲中的資源流動。複雜系統中的突現現象 (emergence) 只能當系統實際運行後才會呈現，意味著同屬於複雜系統的遊戲，相當仰賴原型建置與遊戲測試。為了要驗證遊戲的可玩性，最好的方法便是以某種形式去玩這款遊戲。

1. Machination 框架

遊戲經濟的基本要素有資源、實體與四種使資源能夠進行生產、流動與消費的機制。然而遊戲機制與其結構在大多數的遊戲中並非顯而易見，絕大部分的機制都隱藏在程式碼之中，之外還有有限狀態機的狀態圖、Petri 網等表達工具，不過使用的門檻較高。因此，Machinations 框架利用了遊戲內部經濟的概念，將遊戲內的活動、互動與溝通給模組化。非程式設計師的其它使用者，像是遊戲設計師若能夠掌握 Machinations 這類的高度抽象遊戲開發工具，方能改善目前的遊戲設計環境。

以下列表為 Joris Dormans 發明的 Machinations 框架所涵蓋的部份元件說明（詳細的使用方式與範例請參考 [7]）：

* 節點：一種能夠參與資源生產、分配與消耗活動的元件。
* 資源通路：規定資源如何在各元件之間流動。
* 狀態通路：規定元件狀態改變的節點會對其它元件產生影響。
* 啟動模式：決定節點啟動的時機。
* 門的類型：能夠利用概率、條件型的輸出端來分配資源。
* 其它元件：包含了遊戲目標、模擬玩家、暫存器等。

|  |
| --- |
| 圖十一、Machinations 流程圖，應用於遊戲「Pac-Man」之範例 |

1. 遊戲中的反饋結構促使系統穩定

遊戲的內部經濟相當重要，它對於遊戲的動態特性具有相當大的影響，其中的反饋循環 (feedback loop) 甚為重要。以現實的自然生態系統為例，能自動調節並維持自身的正常功能。某一原始森林生態系統中的昆蟲族群暴發時，樹木因此受害；而昆蟲的增加為鳥禽類提供了充足的食物，鳥禽類數量隨之增加；然而因為鳥類大量的捕食昆蟲，昆蟲的生長受到抑制，森林的生態平衡將逐漸回復。在、於遊戲「Monopoly」中，以 Machinations 框架所表示的反饋結構圖十二中，以 Machinations 框架表示在 Monopoly 中所出現的反饋結構，能夠發現於資源通路間構成了循環結構。

|  |
| --- |
| 圖十二、於遊戲「Monopoly」中，以 Machinations 框架所表示的反饋結構 |

Jochen Fromm 的理論提到了多反饋循環系統能夠展現出更多的突現特性 [8]。然而為了要確立遊戲中反饋循環的數量相當困難，大部份的遊戲將數量控制在二至四個較為合適。

1. 基於 Machinations 的改寫系統

在基於 Machinations 的改寫系統中，其表達方式變得能夠比任務圖更方便地去涵蓋複雜的機制。同樣的，亦可以透過使用 Machinations 流程圖來更好地組織任務圖，用以表達各機制之間的關係。

|  |
| --- |
| 圖十三、基於 Machinations 的模型轉換應用於週期性結構之釋例 |

當設置完模型轉換的規則後，改寫系統能夠將逐次的轉換歷史紀錄作為關卡改進的基礎。利用此方法所建構出相連的關卡、挑戰等，能夠以新機制代替舊有機制，藉此創造出遊玩的多變性。

|  |
| --- |
| 圖十四、Machinations 經過多層次的改寫系統，能夠以更複雜的機制呈現關卡 |

1. 預計可能遭遇之困難及解決途徑
2. 複雜系統的呈現與關卡的評估問題

在前述提及的 Mission / Space 與 Machinations 兩種框架，Joris Dormans 提到這兩種框架並未嚴謹地考慮部份的動態系統 [5]，特別是遊戲中角色移動、角色間相對位置等，間接影響到玩家接下來會採用的戰術策略。我們預計基於以上框架來實作，倘若模擬遊戲並量化數據，其結果勢必會有一定程度的影響。

舉例來說，圖十一是 Dormans 基於遊戲「Pac-Man」的 Machinations 案例，在這份流程圖中它不以距離、相對位置作為考量條件，而以相當高度抽象的「危險度」表示之，依照目前的鬼怪數量、鬼怪存在於場上的時間來增減危險度。我們實際上也能夠加入更詳細的遊戲狀態資訊至 Machinations流程圖中，但是會與 Machinations 的初衷背道而馳，導致流程圖的易讀性嚴重下降。儘管加入前述的資訊，無法克服模擬小精靈行動策略，仍需要以遊戲設計師預先定義好的機率分配來決定角色接下來的戰術機制。

1. 3D 空間的呈現問題

＜still under construction.＞

1. 使用者操作介面設計

在圖形使用者介面中，電腦螢幕上顯示視窗、圖示、選單、按鈕等圖形，各自表不同目的之動作，使用者透過滑鼠等指標性裝置的操作與其互動。如同整合開發環境（IDE），目的在於輔助、加速程式設計師的開發時程，並減少底層的繁瑣操作等。而本次計劃中的項目「自動化關卡生成工具」，便是利用圖形化的方式呈現任務、空間語法，甚至是遊戲機制、遊戲地圖等，要如何設計介面、改良操作操作流程便是相當重要的課題。

而以土法煉鋼的方式實作互動操作指令：點擊、雙擊、拖曳等功能，儘管自由度較高，但並非有效率的做法。我們會依照所選之程式語言、環境，選擇開發程度已相當成熟的互動類別之函式庫來輔助開發。

1. 預期完成之工作項目及成果
2. 預期完成之工作項目

對於本研究計劃，目前我們預期的研究工作項目如下：

* 收集程序化遊戲關卡生成相關文獻資料，並研讀分析、比較其優缺點。
* 開發基於「Mission / Space 框架」方法之自動化關卡生成工具。
* 改良「Mission / Space 框架」使之能夠支援 3D 空間的遊戲應用。
* 結合「Machinations 框架」方法至上述關卡生成工具。
* 將上述關卡生成工具與其它遊戲銜接，實際產生遊戲地圖。
* 收集使用者回饋資料，以驗證遊戲之難易度、耐玩度等指標，並進行調整。
* 撰寫論文發表。
* 撰寫研究報告。

1. 預期完成之研究成果

本研究計畫如能順利進行，我們預期的研究工作成果如下：

* 完成自動化關卡生成工具。
* 改良關卡自動生成之演算法。
* 程序化遊戲關卡生成相關論文發表。
* 研究報告。

1. 對於學術研究、國家發展及其他應用方面預期之貢獻

若能夠完成此項研究計畫，在學術研究方面，藉由研讀相關論文、參考資料，實際讓計劃研究人員能夠學習團隊合作的開發經驗、溝通協調、程式撰寫訓練等，在人才的培育的貢獻上有助於未來進入產業界時，更能夠如魚得水；同時發表研究成果於專業論文期刊，提升國內學術研究的水平外，亦可以將相關知識、技術等，以技術轉移的方式傳授給其它公司。

在國家發展及其它應用方面，希冀藉由此項研究成果，帶領其它遊戲製作公司與工作室，更加重視程序化自動生成之技術，便能利用相關知識及技術改善舊有的遊戲開發模式，進而與國際接軌。

1. 參與人員，預期可獲得之訓練

誠如我們計畫所陳列的研究方法以及預估的可能難題，在進行研究與克服挑戰解決問題的同時，我們預期參與人員將可得到學術上與軟體開發工程上的訓練效果。

學術研究訓練：

* 相關文獻與工具的搜尋、研讀，與口頭報告表達能力之培養。
* 獨立研究精神、研究價值與處事態度的培養。
* 對正規語言之了解。
* 對基於語法自動生成之研究。
* 兩種語言函數之對應域之集合、代換研究。
* 遊戲經濟模型之模擬與建立。
* 電腦繪圖學之基本知識。
* 基於 L-System 形式語法之生成。

軟體開發技術與工程訓練：

* 團隊合作與版本控制系統。
* 系統規格設計與應用開發。
* 圖形化操作介面系統開發。
* 軟體整合測試與系統測試。
* 編譯語言設計與實作。
* 物件導向程式設計之基本知識。

1. 執行進度

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 工作項目 | | 月份 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |
| 總計 |
|  | 收集程序化遊戲關卡生成相關文獻資料，並研讀分析、比較其優缺點 | 12 | T | T | T | T | T | T | T | T | T | T | T | T |
|  | 開發基於「Mission / Space 框架」方法之自動化關卡生成工具 | 6 | T | T | T | T | T | T |  |  |  |  |  |  |
|  | 改良「Mission / Space 框架」使之能夠支援 3D 空間的遊戲應用 | 4 |  |  |  |  | T | T | T | T |  |  |  |  |
|  | 結合「Machinations 框架」方法至上述關卡生成工具 | 4 |  |  |  |  | T | T | T | T |  |  |  |  |
|  | 將上述關卡生成工具與其它遊戲銜接，實際產生遊戲地圖 | 4 |  |  |  |  | T | T | T | T |  |  |  |  |
|  | 收集使用者回饋資料，以驗證遊戲之難易度、耐玩度等指標，並進行調整 | 5 |  |  |  | T | T | T | T | T |  |  |  |  |
|  | 撰寫論文發表 | 4 |  |  |  |  |  |  |  |  | T | T | T | T |
|  | 撰寫研究報告 | 2 |  |  |  |  |  |  |  |  |  |  | T | T |

1. 參考資料

|  |  |
| --- | --- |
| [1] | A. Zook and M. O. Riedl, "Automatic Game Design via Mechanic Generation," 2016. |
| [2] | C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003. |
| [3] | D. Karavolos, A. Liapis and G. N. Yannakakis, "Evolving Missions to Create Game Spaces," 2016. |
| [4] | J. Dormans, "Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games," in *In Proceedings of the 2010 workshop on procedural content generation in games*, 2010, June. |
| [5] | J. Dormans, Engineering emergence: applied theory for game design, 2012. |
| [6] | J. Dormans, "Level design as model transformation: a strategy for automated content generation," in *In Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, 2011, June. |
| [7] | Adams, E., & Dormans, J., Game mechanics: advanced game design, New Riders, 2012. |
| [8] | J. Fromm, "Types and forms of emergence," arXiv, 2005. |
| [9] | J. Dormans, "Generating emergent physics for actionadventure games," in *In 3rd Workshop on Procedural Content Generation in Games*, 2012, May. |
| [10] | J. Dormans, "The machinations framework," [Online]. Available: http://www.jorisdormans.nl/machinations. |
| [11] | K. Neil, "Game Design Tools: Can They Improve Game Design Practice?," 2015. |
| [12] | A. Liapis, G. N. Yannakakis and J. Togelius, "Towards a generic method of evaluating game levels," in *Proceedings of the AAAI Artificial Intelligence for Interactive Digital Entertainment Conference*, 2013. |
| [13] | S. Bj¨ork and . J. Holopainen, "Patterns in Game Design," in *Charles River Media*, 2014. |
| [14] | D. Karavolos, A. Bouwer and R. Bidarra, "Mixed-initiative design of game levels: Integrating mission and space into level generation," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2015. |
| [15] | G. Smith and J. Whitehead, "Analyzing the expressive range of a level generator," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 2010. |
| [16] | B. Lavender and T. Thompson, "A Generative Grammar Approach for Action-Adventure Map Generation in The Legend of Zelda," 2016. |