

逢甲大學資訊工程學系

基於變異測試之遊戲化程式學習開發

指導教授：薛念林教授

學生姓名：莊鎮維

摘要

隨著現代資訊科技的發達，程式語言的種類也越來越多，每種程式語言都有著各自的優缺點，有的跨平台性高、能在多數作業系統上執行、但功能複雜容易佔據大量資源，有的容易學習適合初學者、但執行速度緩慢，但是，這些程式語言最終還是逃不離一件事情，那就是「測試」，測試有很多種，驗收測試、系統測試、整合測試、等等種類多繁的測試，而在這之中用來測試程式最底層的單位則是被稱為單元測試。透過單元測試，能夠在開發週期的早期發現錯誤進行修復並節省成本，並且幫助程式開發人員了解程式代碼，而為了確認單元測試了正確性

通過單元測試達成過速修改的目標，但單元測試並非總是有效的，因為單元測試皆為程式開發人員自行撰寫而成的，

第一章 緒論

1.1 研究背景

隨著現代資訊科技的發達，程式語言的種類也越來越多，而為了確保程式執行正確，程式開發者會進行所謂的「測試」，測試的種類很多，比如說：驗收測試、系統測試、整合測試、等等種類繁多的測試，而在這之中用來測試程式最底層的單位則是被稱為單元測試。透過單元測試，程式開發者能夠在開發週期的早期發現錯誤進行修復並節省成本，並且幫助程式研發團隊了解程式代碼。

1.2 研究動機

透過學校老師的教導、以及在網路上所接觸到與單元測試及變異測試有關的資訊後，我了解到了單元測試對於一個程式開發者的重要性，但是單元測試也是並非總是有效的，程式開發者一不小心就有可能寫出不夠完善的單元測試，不完善的單元測試自然無法找出程式有問題的地方，到最後整份程式完成後才發現程式出錯了，就必須花更多的精力去修正這些問題，而為了避免以上的事情發生，就必須要提高單元測試的正確性，這時候可以藉由變異測試來檢測單元測試的正確性，以提高單元測試的正確性，而為了能讓更多人去認識到這件重要的事情，我希望能透過開發出一個遊戲系統，可以讓人們更加理解變異測試是如何運行的以及理解它的原理。

第二章 相關文獻回顧

2.1 單元測試

在電腦編程中，單元測試 (Unit Testing) 又稱為模組測試，是針對程式模組（軟體設計的最小單位）來進行正確性檢驗的測試工作。程式單元是應用的最小可測試部件。在程序化編程中，一個單元就是單個程式、函式、過程等；對於物件導向程式設計，最小單元就是方法，包括基礎類別（超類）、抽象類、或者衍生類別（子類）中的方法。通常來說，程式設計師每修改一次程式就會進行最少一次單元測試，在編寫程式的過程中前後很可能要進行多次單元測試，以證實程式達到軟體規格書要求的工作目標，沒有程式錯誤；雖然單元測試不是必須的，但也不壞，這牽涉到專案管理的政策決定。每個理想的測試案例獨立於其它案例；為測試時隔離模組，經常使用 stubs、mock 或 fake 等測試馬甲程式。單元測試通常由軟體開發人員編寫，用於確保他們所寫的代碼符合軟體需求和遵循開發目標。它的實施方式可以是非常手動的（透過紙筆），或者是做成構建自動化的一部分。

2.2 變異測試

變異測試 (Mutation Testing) 是一種在細節方面改進程序源代碼的軟件測試方法。這些所謂的變異，是基於良好定義的變異操作，這些操作或者是模擬典型應用錯誤（例如：使用錯誤的操作符或者變量名字），或者是強制產生有效地測試（例如使得每個表達式都等於 0）。目的是幫助測試者發現有效地測試，或者定位測試數據的弱點，或者是在執行中很少（或從不）使用的代碼的弱點。

2.3 Django

基於 Python 寫成的免費而且開放原始碼的 Web 應用程式框架，可以快速開發安全和可維護的網站。有活躍繁榮的社區、豐富的文檔、以及很多免費和付費的解決方案。可以（並已經）用於構建幾乎任何類型的網站—從內容管理系統和維基，到社交網絡和新聞網站。它可以與任何客戶端框架一起工作，並且可以提供幾乎任何格式（包括 HTML、RSS、JSON、XML 等）的內容。

2.4 Python

Python 是一種易學、功能強大的程式語言。它有高效能的高階資料結構，也有簡單但有效的方法去實現物件導向程式設計。Python 優雅的語法和動態型別，結合其直譯特性，並且其本身擁有一個巨大而廣泛的函式庫，除了內建的庫外，Python 還有大量的第三方函式庫，使它成為眾多領域和大多數平臺上，撰寫腳本和快速開發應用程式的理想語言。

第三章 研究架構

本研究為遊戲系統的開發與建置，將利用 Django、HTML、CSS、JavaScript、Python 開發一遊戲系統。在本章節中主要分為兩個部分。第一點為使用者與遊戲系統之互動關係，第二點將闡述遊戲系統之研究架構。

3.1 使用者與遊戲系統之互動關係

圖 3.1 為互動關係圖，在此系統中，使用者只需要輸入測試資料，遊戲系統會根據使用者輸入自動產生相對應的驗證函式，並自動執行變異測試，最後輸出執行結果供使用者查看。

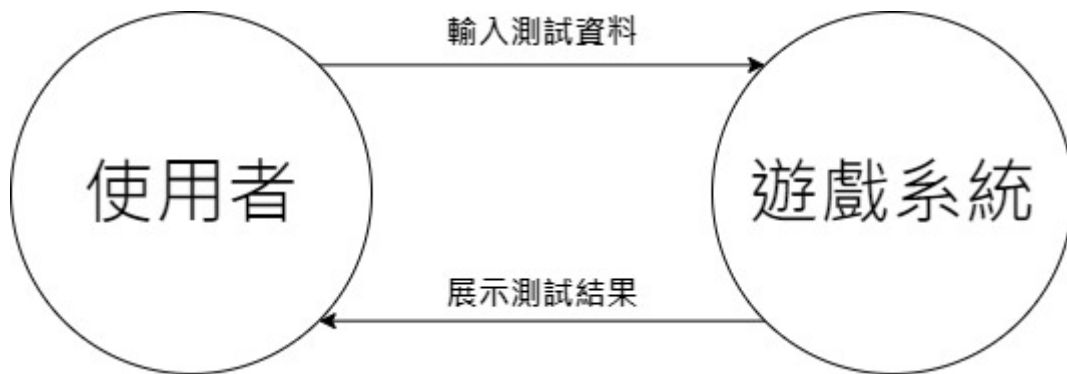


圖 3.1 使用者與遊戲系統之關係

3.2 研究架構