

## 基於強化學習遊戲機器人

專題編號：114-1-CSIE-S001  
執行期限：113 年第 1 學期至 114 年第 1 學期  
指導教授：郭忠義  
專題參與人員：  
111590010 林益賢  
111590020 陳偉瑜

### 一、摘要

本專題旨在開發一套基於深度強化學習的遊戲機器人，研究環境為《空洞騎士》。藉由使用卷積神經網路與殘差網路架構處理遊戲畫面影像資訊，模型可自動學習在不同情境下的最適動作策略。將整合畫面擷取、目標辨識、與動作控制模組，使人工智能能夠在遊戲中進行移動與攻擊等決策行為。本專題設計的 DQN 模型能在複雜的遊戲環境中自主學習並完成基本任務，證實強化學習在遊戲 AI 領域的應用潛力。

**關鍵詞：**深度強化學習、DQN 演算法、電腦視覺

### 二、緣由與目的

近年來，人工智能在遊戲領域的應用日益成熟，從棋類 AI 到能處理即時策略與動作遊戲，展現出其模擬人類學習與決策的潛力，擴展應用可能，本專題目標在於建構一套具自適應能力的 AI，使其能透過觀察遊戲畫面，自主學習操作策略並完成自動通關。

### 三、系統架構

本專題的系統架構分遊戲資訊提取模組（外部 API）、攝取影像模組、DQN 訓練模組、ResNet 預訓練模組，四大主系統，由遊戲畫面擷取開始，透過預訓練模型進行特徵提取，並輸入全連接層以輸出對應各動作的 Q 值，接著系統結合  $\epsilon$ -greedy 策略進行動作選擇，再由動作控制系統將決策轉換為遊戲操作，以下為系統架構圖。

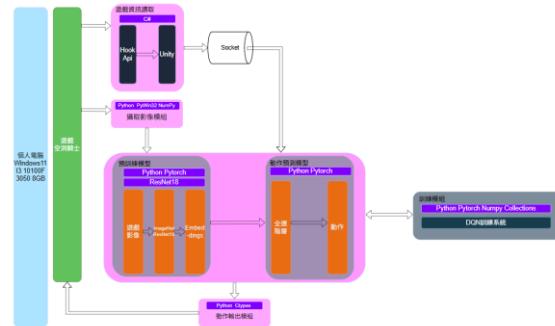


圖 1. 系統架構

### 四、模型訓練步驟及方法

#### (一) 訓練預訓練模型

本研究使用 ResNet18 架構訓練預訓練模型，透過預訓練模型辨別敵人的各個動作，並將輸出的 Embedding 向量作為主模型的輸入。

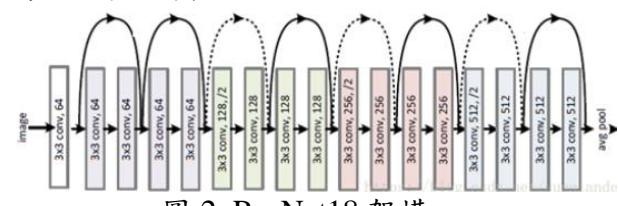


圖 2. ResNet18 架構

#### (二)收集訓練、測試資料

我們將 BOSS 的動作拆分為六類【圖 3 所示】，透過 HK Modding API[8]寫的模組即時回傳 BOSS 的座標，並且計算比例把遊戲內的座標轉換成螢幕上的座標，只要執行遊戲，透過剛剛得到的座標輸入到攝取影像模組即可獲得 BOSS 的各個動作影像，並根據類別進行分類，最後將標記好的資料拆分資料集以及驗證集，比例為 80% 資料集 20% 驗證集。

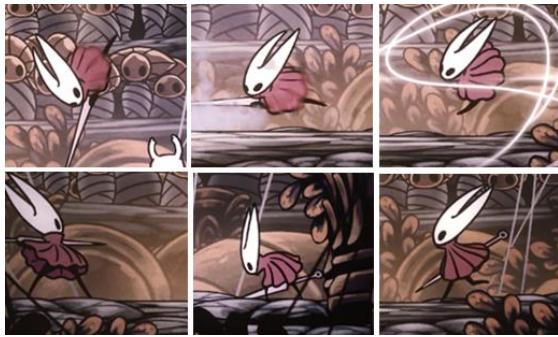


圖 3. 六類動作

### (三)獎勵設計

DQN 嘘勵系統共可分為移動獎勵【圖 4 所示】以及動作獎勵【圖 5 所示】，針對 AI 做出的行動給予正回饋或是負回饋，以引導其學習最佳策略。

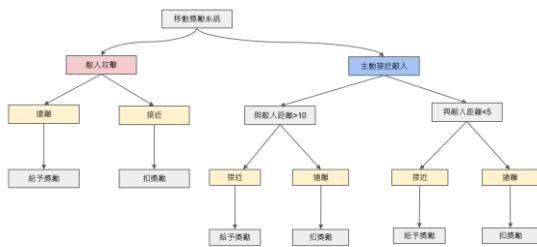


圖 4. 移動獎勵

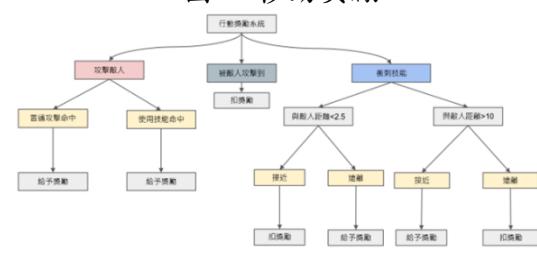


圖 5. 動作獎勵

### (四)訓練主模型

結合預訓練模型當作輸入，並在每一輪遊戲同時收集訓練資料，遊戲結束透過經驗回放訓練主模型，並使用均方誤差作為損失函數。

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

### 五、模型訓練成果

總共訓練 500 場遊戲【如圖 6 所示】可以清楚的看到原本獎勵值由負的到正的，代表 AI 有正確的再學習，驗證神經網路的有效性，可以正確地抓出人物或敵人的一些細節並做出正確的決定。

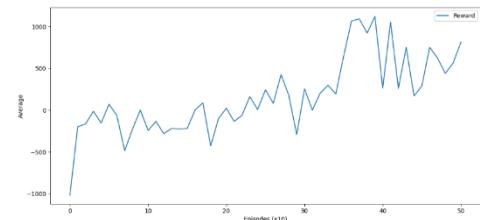


圖 6. 訓練獎勵成長圖

### 六、勝率測試

將訓練好的模型進行 100 場的遊戲【如圖 7 所示】，共獲得 83 場的勝利 17 場失敗，成功將之前模型勝率由原先的 26% 調整到 83%，驗證模型並不只是獎勵值高，勝率也很高。

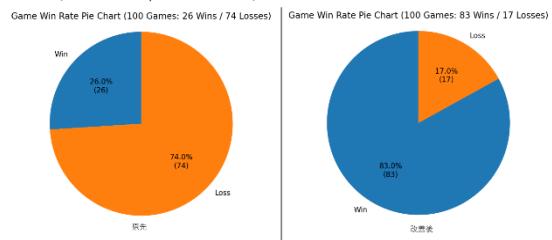


圖 7. 勝場統計

### 七、結論

未來展望可以讓模型操控的角色達到 100% 通關率，甚至在無受傷的情況下達成完美通關。同時，此模型也可移植至其他 2D 遊戲並保持穩定表現，展現跨遊戲的適用性。此外，若能將此技術結合至遊戲系統中，便能實現可自動學習的電腦對手，讓玩家在與電腦互動的過程中不斷提升其智慧水準。如此不僅能增加遊戲的挑戰性與樂趣，也能推動遊戲 AI 向更高層次的智能發展。未來研究方向可著重於強化學習效率、降低訓練資源消耗，以及結合更進階的電腦視覺技術，以使 AI 更貼近人類的反應與判斷能力。

### 八、參考文獻

- [1] Black-Myth-Wukong-AI  
<https://github.com/Turing-Project/Black-Myth-Wukong-AI>
- [2] ResNet-PyTorch  
<https://github.com/JayPatwardhan/ResNet-PyTorch>
- [3] Playing Hollow Knight with reinforcement learning.  
[https://github.com/seermer/HollowKnight\\_RL](https://github.com/seermer/HollowKnight_RL)

[4] Dueling Network Architectures for Deep Reinforcement Learning

<https://arxiv.org/pdf/1511.06581>

[5] Deep learning with PyTorch

[https://books.google.com.tw/books?hl=zh-TW&lr=&id=BjszEAAAQBAJ&oi=fnd&pg=PR15&dq=pytorch+&ots=Qr0dWEF0Fm&sig=amb50IekBIHYYHjClmD6n5pUWgQ&redir\\_esc=y#v=onepage&q=pytorch&f=false](https://books.google.com.tw/books?hl=zh-TW&lr=&id=BjszEAAAQBAJ&oi=fnd&pg=PR15&dq=pytorch+&ots=Qr0dWEF0Fm&sig=amb50IekBIHYYHjClmD6n5pUWgQ&redir_esc=y#v=onepage&q=pytorch&f=false)

[6] An Introduction to Convolutional Neural Networks

<https://arxiv.org/abs/1511.08458>

[7] Playing Atari with Deep Reinforcement Learning

<https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf>

[8] HookApi

<https://radianc.synthagen.net/apidocs/Hooks.html>