深度強化學習於 Unity 之應用

專題編號：111-CSIE-S019
執行期限：110 年第 1 學期至 111 年第 1 學期
指導教授：郭忠義 教授
專題參與人員： 108590028 楊勤義
108590039 吳柏諭

一、摘要

本研究著重在深度強化學習在遊戲上的應用。遊戲環境是使用 Unity 進行開發，以 PVE 戰鬥遊戲作為遊戲類型，而 AI 將以玩家的角度摸索在操作時會出現的各種情況，藉此能找出那些不希望存在的狀態。

我們將研究 Double DQN、Policy Gradient 及其他深度強化學習方法，並以行為和角色間的狀態來讓 AI 尋找是否存在必勝的方法，尋找能順利且較快收斂的算法。

關鍵詞：深度強化學習、Unity、PVE 遊戲

二、緣由與目的

在現今的遊戲開發中，Boss 與角色之間的平衡需要大量玩家進行測試，若測試少了，則開發商必須端出一個可能有許多 BUG 的遊戲，測試多則需耗費大量時間成本，本專題意旨透過深度強化學習同時訓練玩家和 Boss，從而找到 Boss 某些狀況下可能會過強或過弱，例如不停施放絕對命中技能影響遊戲體驗，或是玩家可以透過地形卡點讓 Boss 無法行動的情形發生，觀察這些行為以便修改上述問題，以減少測試時所需的人力成本。

三、研究報告內容

(一)、工具說明

使用 Unity 進行遊戲環境的建置，並使用 C#開發深度強化學習與遊戲程式。

(二)、使用技術方法

Double DQN :

使用兩個神經網路分別計算估計值及實際值，依狀態獎懲及誤差再行修改神經網路的權重，進而達到學習的目的。

Policy Gradient :

使用一個神經網路根據輸入來選擇輸出動作，與 DQN 不同的是它是以回合為單位進行訓練，回合結束再根據期間表現給予獎勵回饋，而 DQN 是以每一步為單位進行訓練，Policy Gradient 優勢在於對連續動作的收斂效果較 DQN 好，因此本專題採用 Policy Gradient。

(三)、架構流程

深度學習的部分本專題依照其運行原理，以物件導向方式實作，其架構如下圖 1，我們將 Node、Layer 與 Network 分別做成 class，Layer 包含 Nodes，Network 包含 Layers，以此構建出整個神經網路。

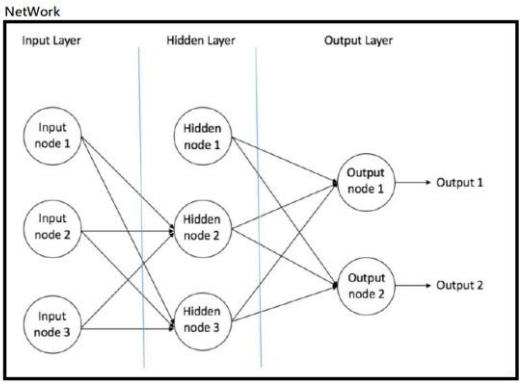

圖 1 神經網路架構圖

除了以上三個 class，深度學習依照不同使用方法會有不同的激勵函數、損失函數及優化器，本專題也有針對以上三者抽出成介面，之後若要進行擴充則只需實作介面即可。

遊戲的部分本專題使用 Unity 進行開發，玩家角色建模與動畫我們使用 Assest store 中的免費包，Boss 動畫則是使用 Unity 內部工具進行錄製，而程式核心架

構部分如圖 2 UML 圖。

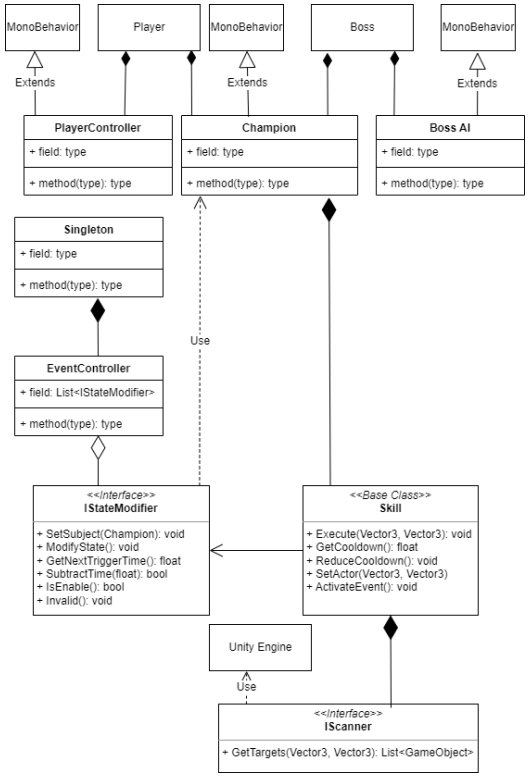

圖 2 遊戲程式 UML 架構圖

Player 分成兩個部分，一個是 PlayerController，是用來供 AI 操作移動及施放技能，另一個是 Champion，用來記錄 Player 的能力值、擁有的技能以及帶有的異常狀態。

IStateModifier 是用來實作異常狀態的介面，運作方式是將目標角色血量、速度等數值持續或定時改變，並使用 EventController 控制計時施放，直至效果時間歸零。

Skill 是技能的基礎類別，運作方式是使用 IScaner 搜尋符合條件的角色，並對目標進行傷害或使用 IStateModifier 實作的異常狀態到目標。

在強化學習演算法部分，由於本專題的遊戲在進行過程中是一個連續不斷的環境變化，若使用 QLearning 不斷根據環境選出較佳的連續行為，則會比較吃不消，故我們選用 Policy Gradient，因為此演算法是可以對一個回合來選出一系列行為。

而 Policy Gradient 在反向傳播上是將一回合中的動作根據獎勵機制來提升或抑制行為出現的機率。在獎勵的選擇中，擊敗 boss 是主要目標，而攻擊到 boss 是加分項目，被攻擊到則為扣分項目。角色的行為有向左轉、向右轉、前進以及使用技能，技能的部分目前設定為兩個。

(四)、結論

在設計 AI 時，由於我們是一步一步地加深演算法，故可以看到不同強化學習演算法的適用不同，從為了改良 QLearning 需要存儲每一種狀態，而使用 NN 來降低大小而生的 DQN，再到為了處理連續資料而使用的 Policy Gradient，因此更深刻了解不同強化學習的運作模式。

由於我們從深度學習演算法就開始自製，為了實作出深度學習我們看了很多資料，其中也碰上了不少困難，如梯度消失與無法收斂，除此之外還有公式推導與程式架構設計等問題，在一一克服後，我們能夠感受到自己的進步，並且能認知到自己的不足，期望這些經驗能夠在未來帶給我們思考層面的轉變，化為養分支撐我們繼續前進。

參考文獻

莫煩(2016 年 11 月 03 日)。什麼是 Q Learning。取自:
https://mofanpy.com/tutorials/machine-learning/reinforcement-learning/intro-q-learning/。2022 年 05 月 15 日訪問

雞雞與兔兔的工程世界(2018 年 8 月 05 日)。深度學習優化器使用。取自:
[機器學習 ML NOTE]SGD, Momentum, AdaGrad, Adam Optimizer | by GGWithRabitLIFE | 雞雞與兔兔的工程世界 | Medium。2022 年 05 月 15 日訪問

深度強化學習 DQN 進化史(2019 年 2 月 28 日)。取自:
深度强化学习(三)——DQN 进化史, A2C & A3C (antkillerfarm.github.io)。2022 年 05 月 15 日訪問

UnityAPI。取自:
Unity - Scripting API: (unity3d.com)。2022 年 05 月 15 日訪問