即時神經渲染技術之多人生存遊戲設計與實現

Design and Implementation of a Multiplayer Survival Game Using Real-Time Neural Rendering Technology

指導教授：辛錫進

學生：毛彥惟、黃銘昱、張志威

國立聯合大學 資訊工程學系

苗栗市南勢里聯大2號

flhuang@nuu.edu.tw

摘要

本研究旨在發展即時神經渲染與現代封閉原始碼類遊戲引擎(及編輯器)之整合，研究探討神經渲染技術並實現神經壓縮紋理，並嘗試結合 Unity 進行多人遊戲連線狀態同步與互動。最後設計並實現具有系統性的遊戲架構以及模組化程式設計以實現遊戲之可擴充性、維護性與可重用性。

關鍵詞：Unity3D、Netcode for GameObec (NGO)、Procedural-Generation、Neural Shading、Neural Texture Compression。

Abstract

This project aims to develop an integration of neural shading and modern closed source real-time render engine (with its own editor). We investigate and implement the cores of neural texture compression, and try to integrate the state-of-the-art technologies into Unity with implementation of synchronized multiplayer networking and interactive gameplay. As a result, a systematic game architecture and the modular programming approach have been designed and implemented to achieve the goals of extensibility, maintainability, and reusability within the game system

關鍵詞：Unity3D、Netcode for GameObec (NGO)、Unity Relay、Procedural Generation、 Neural Shading 、 Neural Texture Compression。

一、研究動機

隨著即時渲染技術快速發展，遊戲對高擬真光照與材質呈現的需求日益提升。然新型渲染技術，如合成神經紋理[Tu et al.2024]、實時外觀神經模型[Zeltner et al.2024]、ReSTIR[Ouyang et al. 2021]等一系列先進即時渲染技術於現代商業化遊戲引擎之應用過於緩慢。

本研究希望將神經渲染技術與現代封閉原始碼遊戲引擎，探索其在即時互動與整合可行性，並以完整的多人生存遊戲設計驗證技術整合的實用價值，期望為未來遊戲開發提供更高效且逼真的呈現方式。

二、開發環境介紹

1. Unity

我們使用 Unity 6000.0.47f1 版本做為開發整合平台，Unity 是一款可跨平台發布遊戲、應用程式之封閉原始碼遊戲引擎，含有許多套件，並有提供免費版本給個人開發者。

2. Visual Studio

程式撰寫使用 Visual Studio 2022 作為開發工具，並以 C# 為主進行編碼。

3. Blender

3D 建模使用 Blender 進行開發，這個軟體可佈線式建模、雕刻、模型上色、製作 模型的骨架及動畫。

4. Nvidia Optix SDK

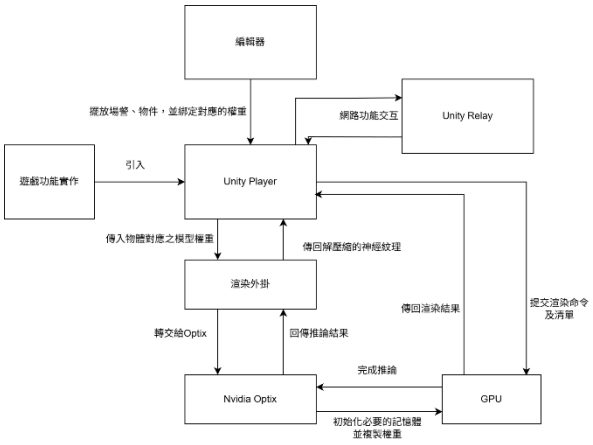
由於作為必須功能的合作向量於 DirectX 12及 Vulkan 兩種主流圖形 API 中仍屬於實驗性功能，受 Unity 底層渲染 API 架構設計所限，無法於外掛中調用，於此我們選用 Nvidia Optix 作為執行推論的工具。Optix 為 Nvidia 推出之一光線追蹤應用程式介面(API)，基於 CUDA 搭建，可提供易於學習的光線追蹤開發環境。

5. Nvidia RTX Neural Shading SDK 與 Nvidia RTX Neural Texture Compression SDK

使用於早期技術驗證及前期渲染功能開發，後因確認 Unity 之底層渲染 API 架構與 SDK 不兼容而放棄，轉為 Optix 實作。

三、系統架構與功能

1. 系統架構概述



圖一、創建遊戲介面

遊戲中多種類型物件與系統間的互動，採用模組化架構撰寫腳本，從而使各項功能能夠彈性實作與整合。以下詳細說明各項功能。

2. 渲染外掛

圖(二)、一份具有複雜花紋及色塊的平坦紋理及其在2000000批次後的訓練結果

可以注意到暗處的採樣不均勻以及顏色異常加深

Credit:トキビト, 2025

2.1 技術早期驗證

由於如神經紋理壓縮、神經外觀表示等技術仍屬僅有技術演示之範疇，並未有遊戲實際選用此種方式渲染，故對本技術之可靠性實有驗證之必要。

研究小組選擇以 Nvidia RTX Neural Shading SDK (以下簡稱為 NS)及 Nvidia RTX Neural Texture Compression SDK(以下簡稱為 NTC) 提供之範本與程式庫進行驗證，得到了以下資訊：

- 對於複雜樣式紋理，僅仰賴 MLP 網路與解碼器的 NS 預設網路即便將隱藏層的神經元數量提升到128顆，也難以完整重現原圖的複雜紋理。對於紋理中之小型圖樣也經常性發生未在模型中表達的問題。
- 在同一範本中，NS 所提供的預設神經網路架構在重現原圖時會出現暗部顏色相較於原圖大幅加深的情況。由於該情況在未經微調的一般圖像生成網路也可見到，推測為神經網路自身的問題。
- NTC 目前仍需要手動編寫紋理的清單檔案，但現階段由於存在輔助用的壓縮程式及神經紋理預覽器，在開發難度上是最小的。

由於NS表現不佳，故最終決定開發方向為為將 Unity 整合 NTC，並在不影響專案進程的前提上開發用於神經渲染的新網路

2.2 架構

本外掛借助 Unity 提供之底層渲染 API，存取物件之對應模型權重，並轉遞至 Optix 由其初始化存放兩者必需的顯示記憶體，並從 Optix 處取回原始紋理傳回。

3. 遊戲系統與多人連線

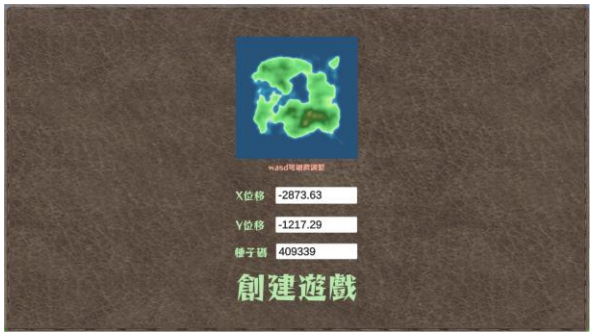
多人連線部分採用 Unity 官方提供的 Netcode for GameObject 作為基礎框架，負責玩家狀態同步、事件廣播與網路資料傳遞。為確保跨網路環境的順利連線，系統加入 Unity Relay 中繼伺服器，使玩家能在不同網路條件（如 NAT、防火牆）下正常加入遊戲。本架構將遊戲邏輯留在本地端處理，Relay 只負責資料轉送，以達到低延遲、安全且穩定的多人合作體驗。

遊戲內容以程序化生成地形技術建立世界環境，透過噪聲函數與隨機參數自動生成山丘、河流等地形特徵，使每次開局都具有獨特性。系統架構採用模組化設計，包含玩家系統、物品與背包系統、製作系統、互動系統與世界狀態管理等模組，各模組以事件驅動方式進行溝通。所有遊戲狀態與物件資料透過 Json 檔案讀寫，使玩家進度可被保存與重載，形成完整且可擴充的遊戲運作框架。

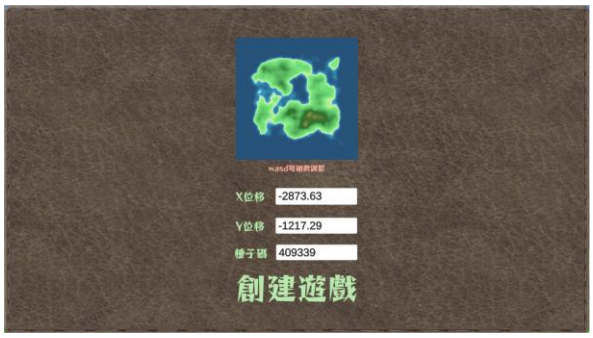
四、專題成果展示

1. 遊戲內容展示

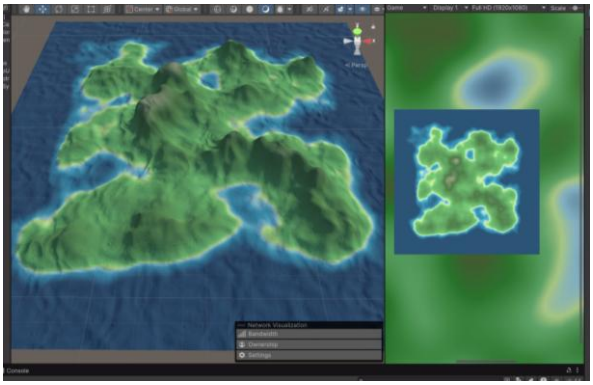
圖(三)、主選單介面



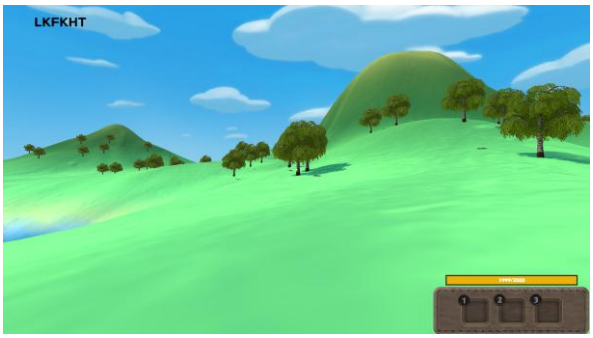
圖(四)、創建遊戲介面



圖(五)、加入遊戲介面



圖(六)、程序化地形生成



圖(七)、場景物件之隨機生成

圖(八)、多人互動

圖(九)、物品製作系統

圖(十)、背包與快捷欄

圖(十一)、砍法樹木實作與同步

圖(十二)、建造功能

圖(十三)、任務系統

圖十四、存檔功能

五、專題作品影音平台網址

https://youtu.be/CViCJ-VHDdQ

六、結論

在本次專題中，我們不僅驗證了神經渲染與現有封閉原始碼引擎深度整合的可行性，也建立了一套具備擴充性與穩定性的多人遊戲框架。此專題使我們對跨領域技術整合、系統設計與多人遊戲開發流程有更深入的理解，並為後續在遊戲開發或圖形技術領域的研究奠定基礎。

七、參考文獻

[1] *Peihan Tu, Li-Yi Wei and Matthias Zwicker. 2024. Compositional Neural Textures. Arxiv preprint arXiv:2404.12509*

[2] *Tizian Zeltner, Fabrice Rousselle, Andrea Weidlich, Petrik Clarberg, Jan Novák, Benedikt Bitterli, Alex Evans, Tomáš Davidovič, Simon Kallweit and Aaron Lefohn. Real-Time Neural Appearance Models. ACM Transactions on Graphics (TOG) 43, 3 (June 2024) Article 33.*

[3] *Yaobin Ouyang, Shiqiu Liu, Markus Kettunen,*

Matt Pharr and Jacopo Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. In Computer Graphics Forum, Vol 40, Wiley Online Library, Issue 8, 17-29.

[4] *Nvidia RTX Neural Shading SDK. Nvidia.* https://github.com/NVIDIA-RTX/Rtxns

[5] *RTX Neural Texture Compression (NTC) SDK.* *https://github.com/NVIDIA-RTX/RTXNTC/*

[6] *Sta. (n.d.). Websocket-Sharp. Github.* https://github.com/sta/websocket-sharp

[7] *Getting Started with Unity*. (n.d.). Unity. https://unity.com/cn/learn/get-started

[8] *Unity Relay (n.d.). Unity.* https://docs.unity.com/ugs/en-us/manual/relay/manual/introduction

[9] *Client-server quickstart for Netcode for GameObjects* (n.d.). Unity. https://docs.unity3d.com/Packages/com.unity.netcode.gameobjects@2.4/manual/advanced-topics/message-system/rpc.html

[10] *Unity Technologies. Unity Official Documentation.* https://docs.unity.com/

[11] *Microsoft. C# Programming Guide.* https://learn.microsoft.com/en-us/dotnet/csharp/

[12] *Getting Started with Unity*. (n.d.). Unity. https://www.json.org/