

輔仁大學資訊工程學系
畢業專題
B03 組

Rish Engine - 2D 遊戲引擎

從零開始自幹遊戲引擎

成員

406262515 鍾秉桓 Roy
406262??? 梁博全 Icejj
406262319 黃育皓 SunTalk
406262163 黃品翰 Halloworld

指導教授
鄭進和

November 29, 2020

Contents

1	摘要	1
1.1	背景簡介	1
1.2	問題說明	1
1.3	實作結果	1
2	緒論	2
2.1	動機	2
2.2	問題描述	2
2.2.1	歷史介紹	2
2.2.2	遊戲是什麼	2
2.2.3	遊戲引擎是什麼	2
2.2.4	遊戲引擎具備的功能	3
3	系統說明	5
3.1	使用技術	5
3.1.1	使用語言	5

Chapter 1

摘要

1.1 背景簡介

本專題是以 C++ 開發之 2D 遊戲引擎，開發者可以使用引擎提供之編輯器 (RishEditor) 編輯遊戲場景 (Scene)，並且可以對遊戲物件 (Entity) 附加遊戲邏輯 (使用 C++ 撰寫，並和編輯器一同編譯)，並支援 Batch Rendering ¹、2D Lighting (Point Light, Ambient Light), Particle System, Constrain-based Physics (支援圓形、多邊形等)

1.2 問題說明

TODO

1.3 實作結果

TODO 本專題製作解決問題方法、創新所在、與實作結果等概要陳述

¹高達 100000 個 sprites

Chapter 2

緒論

2.1 動機

隨著遊戲的發展，現代的遊戲越來越趨於複雜，從數人的小型獨立製作，到數百人的大型 3A 級遊戲，遊戲的規模與以前不可同日而語，現今一個獨立開發工作室製作的 2D 遊戲，在十多年前要達到同樣的規模可能要數十人的團隊才可能達到，而這些節省下來的時間成本，就是多虧了遊戲引擎的強大之處。現代遊戲引擎的複雜度，先進的圖學技術，複雜的物理模擬，是需要一個具有規模的團隊來開發的，我們嘗試以此為目標，試著實作了一個具有一定規模的 2D 遊戲引擎。

2.2 問題描述

2.2.1 歷史介紹

最初並沒有所謂的遊戲引擎，遊戲常常是重頭開始建構 (from scratch) 的，這個概念被眾人所知最早是由 Jhon Carmack 發揚光大，他為 Doom 以及 Quake 系列遊戲開發的 3D 遊戲引擎，深深地影響了遊戲業界，為早期 (1990 末) 的遊戲業界標準，並促成引擎授權的商業模式，使得遊戲軟體的規模上升。

2.2.2 遊戲是什麼

電腦遊戲的技術本質是實時 (real-time) 可交互 (interactive) 的程式，遊戲程式會模擬出不精確但足以表現的遊戲世界，並且根據玩家的輸入做出相對的輸出 (例如操作搖桿控制角色等)，通常遊戲會實作遊戲循環 (Game Loop)，更新遊戲邏輯、物理模擬、更新 AI 等。而通常遊戲要維持在每秒更新 60 幀才能保證流暢運行 (低於 60 fps 通常會感覺卡頓)，也就是要在 16 毫秒內做完所有的遊戲更新並渲染到螢幕上頭，更何況在 VR 上要維持 90 FPS 才不會感覺暈眩，這也是為甚麼遊戲會如此要求效能。

2.2.3 遊戲引擎是什麼

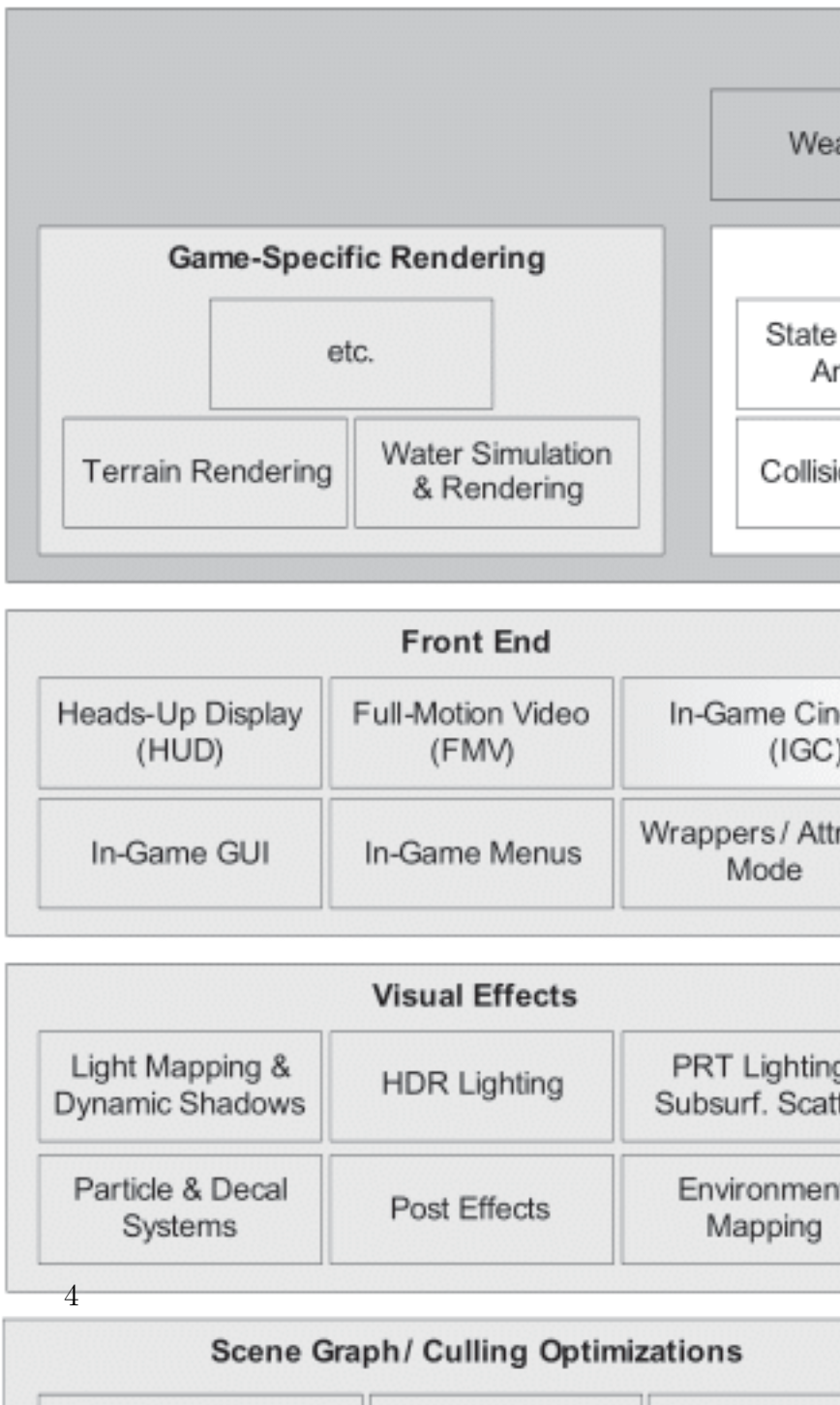
遊戲引擎 (Game Engine) 從字面上解釋是驅動遊戲的基礎程式，因此遊戲引擎須具備了窗口管理、輸出入管理、渲染 (Rendering) 系統、物理 (Physics) 系統…等子系統，

除了基礎程式之外，要是一個遊戲引擎還必須要有讓遊戲開發者使用之工具 (可視化、非可視化)，可能是單個工具或一整個工具鏈 (Toolchain)，使得遊戲開發者可以用該工具開發遊戲 (Developing)、進行測試 (Debugging)、封裝發布 (Shipping) 等，否則就只能被稱為遊戲框架 (Game Framework)。遊戲引擎會讀入自訂的資源 (Asset) 格式 [常常是為了效能才會這樣做]，並且有工具支援設計師將素材 (貼圖、音效、模型等) 轉換成引擎的格式，或是引擎工具可以直接產生，因此遊戲引擎可以說是專門開發遊戲的開發環境。

2.2.4 遊戲引擎具備的功能

TODO

1.6. Runtime Engine Architecture



Chapter 3

系統說明

3.1 使用技術

3.1.1 使用語言

使用 C++ 11 作為主要開發語言，使用 `msys2` 作為套件管理器¹，`CMake` 建置專案，使用 `doxygen` 工具將註解轉換 documentation, `cppcheck` C++ 的靜態程式碼分析器

- Slack
團隊協作軟體，用於溝通紀錄
- Trello
看板軟體，用於工作進度管理
- git, github
版本控制

TODO 寫多一點

為何選 C++ 做為開發引擎之語言？

因為 C++ 讓開發者可以自由掌控記憶體，讓開發者可以精確的控制變數的生命週期，沒有垃圾回收 (Garbage Collection)，這對於效能注重的遊戲引擎來說相當重要；比 C 來得高階但效能卻沒損失很多；C++ 歷史悠久且有許多精良的函式庫可以使用。

使用函式庫

- SFML
一個 C++ 的跨平台用於遊戲、多媒體程式開發的函式庫，於此專案中用於 Input 及窗口操作等
- OpenGL + glad

1

一個跨平台的 API，使用 glad 載入器

- [imgui](#)

一個輕量、快速的 Immediate Mode GUI 函式庫，常在遊戲開發、工具開發等使用，於此專案中用於編輯器的 UI 開發。

- [spdlog](#)

一個輕量、快速的 Logging 函式庫，於此專案中用於處理除錯訊息。

- [nativefiledialog](#)

小型的 Open File Dialog 函式庫，於此專案中用於處理開啟檔案視窗。

- [IconFontCppHeaders](#)

字體的 Helper Headers，於此專案中用於引入自訂字體。

- [fmt](#)

補足了 C++ 標準庫缺少的格式化輸出輸入。

- [glm](#)

OpenGL 數學函式庫，於此專案中用於處理向量、投影等相關數學函式。

- [cereal](#)

C++ 缺少的序列化函式庫，於此專案中用於序列化儲存資源。

- [re2](#)

來自 Google 的 Regular Expression 函式庫，標準庫的有夠慢 (確信)。

-