



自動化遊戲生成系統：大語言模型與擴散模型之整合應用

AUTOMATED GAME GENERATION SYSTEM BASED ON INTEGRATING LLMS AND DIFFUSION MODELS

B-4組

指導教授: 李同益

組員: 曾冠耘



目錄

01 研究動機

02 系統介紹

03 實作技術

04 DEMO 展示

05 總結



01 研究動機



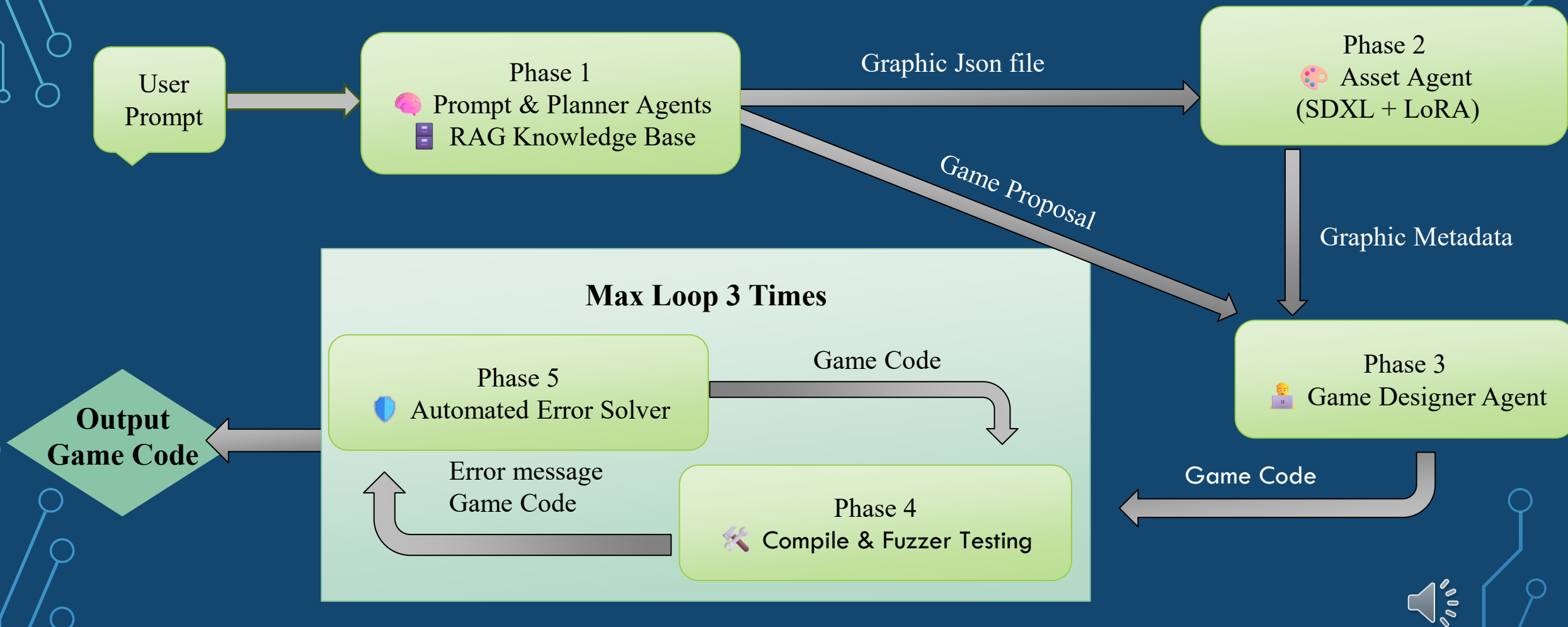
- 近年來，生成式人工智慧已經能夠在圖像與影片方面具有相當不錯的產出。然而，將自然語言直接轉化為能夠與之互動的遊戲，仍是目前較為少見的一個領域。
- 傳統的遊戲開發流程耗時長且步驟繁瑣，需要具備程式邏輯與美術設計能力，往往需要有一個多人團隊進行開發。
- 希望此系統能夠加速遊戲開發的整體流程。



02系統介紹



系統架構圖



系統描述

Optimize Prompt(Phase 1)

- 先**優化使用者輸入**，之後根據優化後的輸入以及 **RAG database**生成 `game_document.txt`**企劃書**，其中，企劃書包含遊戲**整體架構**以及**重要參數**，參數會另外以`game_config.json`的方式儲存

Graphic Produce(Phase 2)

- 先根據`game_config.json`當中儲存的圖片名稱去產出**圖片 metadata**，之後再使用**Diffusion model**根據metadata去產生對應的圖片，最後根據前綴 ([sprite] or [background])去執行圖片去背

Code Produce(Phase 3)

- 根據企劃書以及`game_config.json`中的圖片名稱去**生產遊戲程式碼**

Test Code(Phase 4)

- **編譯**並且**實際執行**遊戲後，**偵測**是否有**錯誤**發生，若無則輸出當前程式碼，若有進入Phase 5處理，最多循環3次

Error Solver(Phase 5)

- 有兩個Agent，一個負責根據Phase傳入的錯誤訊息產出**修正方案**，另一個根據修正方案去**具體修改程式碼**，修改完之後傳回給Phase 4



03 實作技術



RAG DATABASE

1. 路徑演算法

- Dijkstra、A*、DFS

2. 鏡頭控制

- Player-Centered、Bounding Box、Mouse-Driven

3. 物理計算

- Collision、Gravity、Friction、Clock

4. 遊戲狀態

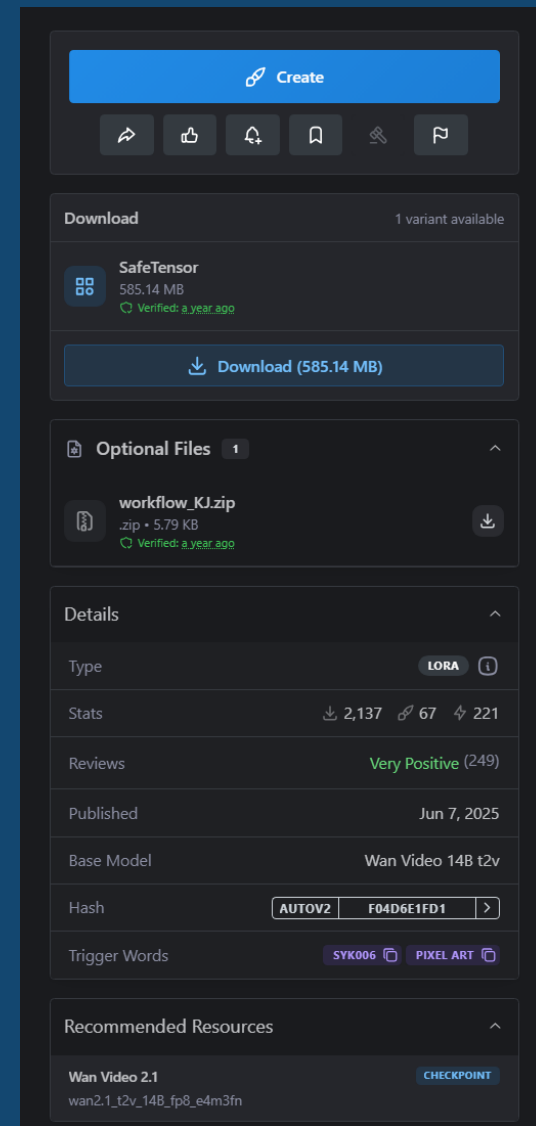
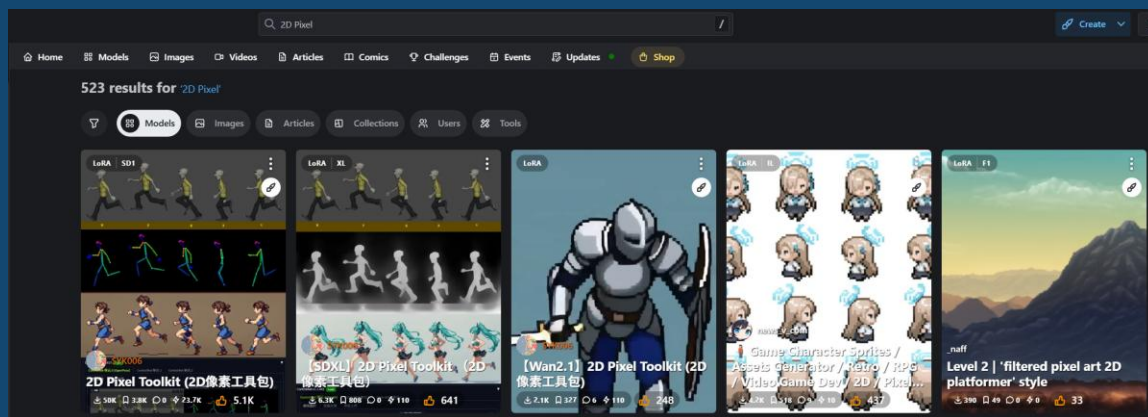
- State machine、Sprite manager

事先寫好，並透過ChromaDB向量資料庫儲存、引用



DIFFUSION MODEL(訓練過程1)

- Civitai上找適合的Base Model
- 蒐集想要訓練的圖片素材



DIFFUSION MODEL (訓練過程2)

- 透過Kohya_ss GUI進行LoRA訓練出想要的模型
- 可自行設定參數以及Output資料夾等

The screenshot displays the Dreambooth LoRA training interface. The top navigation bar includes links for Dreambooth, LoRA (selected), Textual Inversion, Finetuning, Utilities, and About. Below this, a sub-navigation bar shows Training (selected), Tools, and Guides. The main content area is titled 'Train a custom model using kohya train network LoRA python code...'. It features several sections: 'Configuration' and 'Accelerate launch' are collapsed. The 'Model' section is expanded, showing fields for 'Pretrained model name or path' (set to 'runwayml/stable-diffusion-v1-5'), 'Image folder' (with a file selection icon), 'Trained Model output name' (set to 'last'), and 'Dataset config file' (with a file selection icon). Below the 'Model' section is a 'Training comment' field. The 'Save trained model as' section has radio buttons for 'ckpt' and 'safetensors' (selected). The 'Save precision' section has radio buttons for 'float', 'fp16' (selected), and 'bf16'. The 'Folders' section includes fields for 'Output directory for trained model', 'Regularisation directory (Optional, containing regularisation images)', and 'Logging directory (Optional, to enable logging and output Tensorboard log)', each with a file selection icon. The 'Metadata' section is collapsed. The 'Dataset Preparation' and 'Parameters' sections are also collapsed. At the bottom, there are three prominent buttons: 'Start training' (orange), 'Print training command' (gray), and 'Start tensorboard' (blue).

04DEMO



05總結



未來展望

- **突破代碼瓶頸**：針對目前的代碼生成極限(約1300行)，未來可將Code Agent拆分為更多智能體，負責不同的模組並彼此溝通，以支援更複雜的大型遊戲。
- **跨模態擴充**：計畫進一步串接 AI 音效生成模型，打造視、聽、邏輯一體的全方位遊戲生成環境。
- **GUI介面**：加上前端介面，更方便使用者使用及操作。



結語

- 本次專題希望結合了LLM、Diffusion model等模型，讓遊戲開發變得更簡單，使用者也能夠使用Kohya_ss等網站自行訓練心儀的SDXL模型，希望這些能夠讓不太懂遊戲開發流程的使用者們，也能夠創造出自己想要的遊戲。



簡報到此結束
THANK FOR LISTENING

