

與 AI 協作的紀錄與反思

一、我們做了什麼（協作紀錄）

1. 建立遊戲基礎 + 改滑鼠控制

我們一開始製作星際大戰遊戲，遇到 `java.lang.NullPointerException` 錯誤。

AI 幫我們發現是 `Earth` 圖片載入失敗，教我們檢查 `src/main/resources/星際大戰/` 路徑。

修正後遊戲成功執行。

2. 優化程式

我們請 AI 幫忙整理 `Earth.java` 和 `GamePanel.java`，保留 `package` 星際大戰。

AI 也提醒我們檢查其他圖片路徑，確保遊戲正常執行。

3. 加 BOSS 超強雷射 + 特效

我們想要 BOSS 發射超強雷射，AI 幫我們寫了 `BossLaser.java`，改進 `GamePanel.java`。

BOSS 每 5 秒發射，攻擊力 20，還加入爆炸特效與音效，超酷！

4. 玩家受傷要有特效

我們告訴 AI 希望玩家被擊中時也有特效。

AI 修改 `GamePanel.java`，加入爆炸圖與音效，測試無誤，畫面更生動。

5. 加排行榜 + 輸入名字

我們請 AI 幫忙加入排行榜功能，玩家過關或死亡時可輸入名字，並在封面新增「查看排行榜」按鈕。

AI 修改了多個檔案，測試後功能正常。

6. 修正錯誤

我們遇到 `ScoreEntry` 是私有類，導致 `LeaderboardPanel` 無法使用。

AI 教我們改成 `public static`，問題順利解決。

7. 再修正建構子錯誤

`GameOverPanel`、`GameCompletePanel` 出現錯誤，AI 教我們使用 `Consumer<string>`，並修改 `GamePanel` 呼叫邏輯，測試後正常。

`</string>`

二、AI 怎麼幫我們

1. 記憶功能

AI 會記住我們的專案進度，例如 Explosion 特效一致重複使用，保持連貫。

2. 程式分析 + 生成功能

AI 幫我們分析錯誤，生成像 BOSS 雷射、排行榜等功能，還提醒我們檢查資源路徑。

3. 回應快又簡單

AI 提供的程式碼清晰，步驟整理清楚，節省我們理解的時間。

4. 幫我們加錯誤處理

AI 教我們為圖片載入失敗設定預設圖，增強遊戲穩定性。

5. 自動反思

AI 主動提醒我們可優化的部分，例如特效位置或遊戲難度平衡。

三、我們的反思

優點

- ✓ 節省時間，AI 幫我們快速找錯。
- ✓ 程式碼結構清晰，說明簡單。
- ✓ AI 能配合我們的想法逐步實現功能。

挑戰

- ✗ AI 不會自動確認資源是否放好。
- ✗ 我們需要清楚表達需求，AI 才能理解。
- ✗ 多檔案改動時，AI 有時會遺漏，需我們自行測試。

學到的技能

- ✓ Java 圖片、音效載入、UI 互動。
- ✓ 排行榜邏輯、Lambda、Callback。
- ✓ 每步測試、閱讀錯誤訊息、檢查路徑。
- ✓ 與 AI 高效協作，快速修正問題。

改進

- ✓ 未來向 AI 表達需求時會更精準。
 - ✓ 整合前我們會先自行檢查與測試。
 - ✓ 繼續新增功能，如排行榜動態更新、粒子特效。
 - ✓ 利用 AI 進一步優化專案品質。
-

四、總結

這次與 AI 合作，讓我們體驗到 AI 是超強的隊友，能快速幫我們除錯、撰寫程式、擴充功能。

但 AI 無法完全理解我們的全部需求，因此我們需搭配自身的驗證與反思。

未來我們會將這次經驗應用到更多專案，AI 是我們的程式開發小幫手，但我們仍是專案的指揮官！