AI 輔助開發導入

W3 ｜ Unity XR 基礎 × GitHub Copilot

如何用 AI 協作開發 Unity？

今天要回答的問題

如何用 AI 協作開發 Unity？　又如何確定它是對的？

一個常見的失敗

學生把 Copilot 生成的腳本整包貼上 → 編譯錯誤、頭盔上沒反應。

- 原因：AI 用了舊版 API、幻覺方法、或沒設好 Inspector 綁定。
- **教訓：AI 產出 ≠ 完成。**

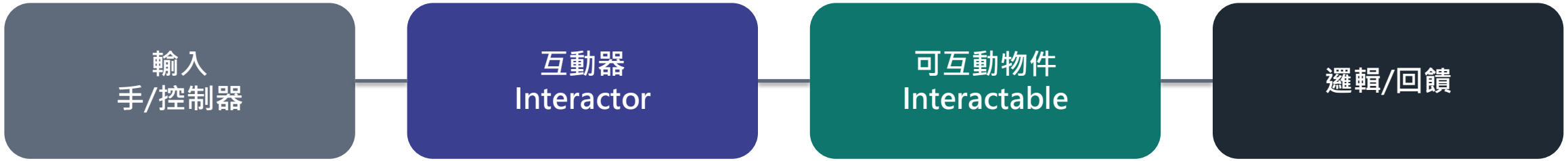
本課的預設開發方式

AI 產出 → Unity Editor 驗證 → 真人修訂 → (迭代)



頭盔與編輯器實測，才是真相來源。

Unity XR 互動資料流



底層：Meta XR SDK（Rig/Passthrough）+ Interaction SDK / XR Interaction Toolkit。

Copilot 的三種用法

行內補全：寫到哪補到哪。
- Copilot Chat：解釋、生成、重構、除錯。
- CLI / Agent：建立專案、跨檔批次修改、跑指令。

AI 協作紀律

每次協作記錄：目標 → prompt → 產出 → 驗證 → 修訂 → 結果。

- 放在 repo 的 /AI-Log/，隨作業繳交。
- **停止條件：連兩次修不好 → 回頭人工查場景綁定與 SDK 版本。**

課堂判斷題

Copilot 生成的腳本一跑就編譯錯誤，說某方法不存在。你先做什麼？

- 以編譯錯誤為準，把訊息回貼給 Copilot、指定最新版 SDK；不是盲目換一段全新程式。

本週行動

用 Copilot 做出「射線抓取並投擲方塊」。
- 在頭盔實測、真人修訂。
- 交 APK/錄影 + GitHub + /AI-Log/。