

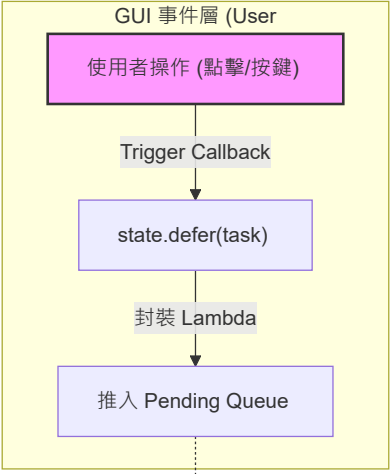
# MuJoCo 3D Editor 全功能系統管線圖 (The Grand Pipeline)

本文件匯總了編輯器所有核心功能的資料流向圖 (Data Flow Diagrams)。

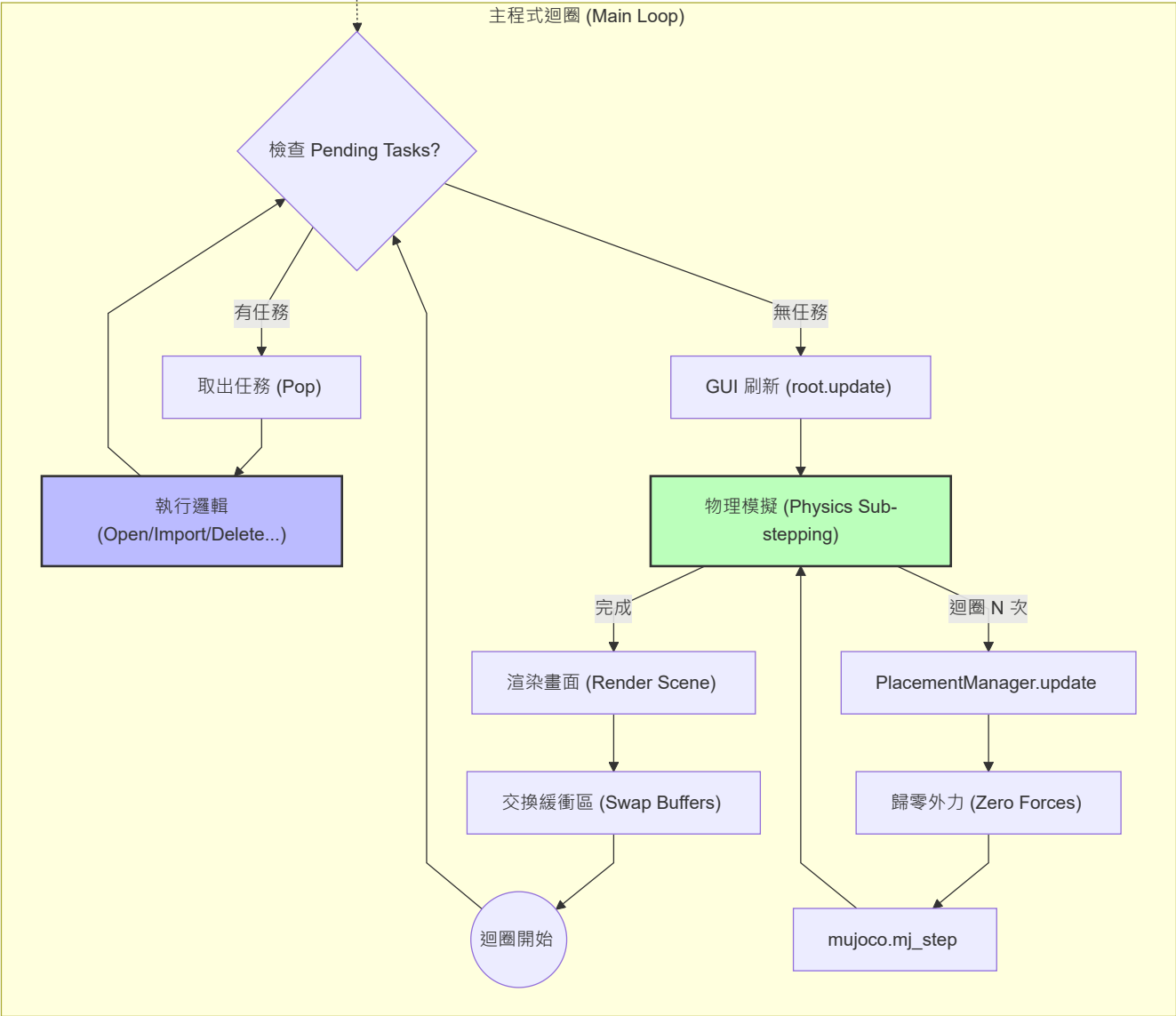
## 1. 系統核心架構 (System Core)

### A. 主迴圈與排程機制 (The Main Loop & Defer Mechanism)

解釋：這是程式的心臟，展示了如何解決 GUI 與物理引擎的執行緒衝突問題。



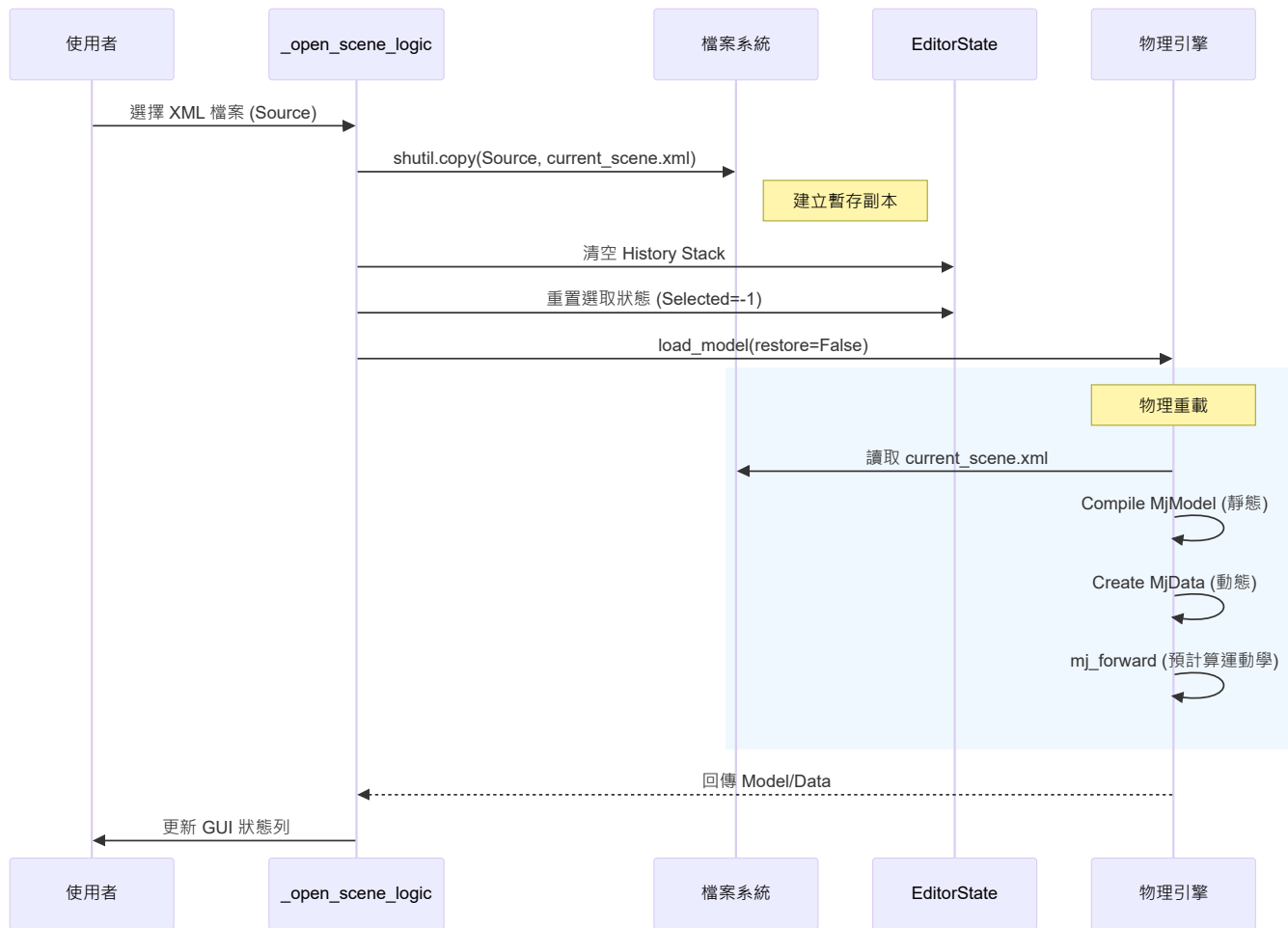
等待下一幀



## 2. 場景管理 (Scene Management)

### B. 開啟場景 (Load Scene Pipeline)

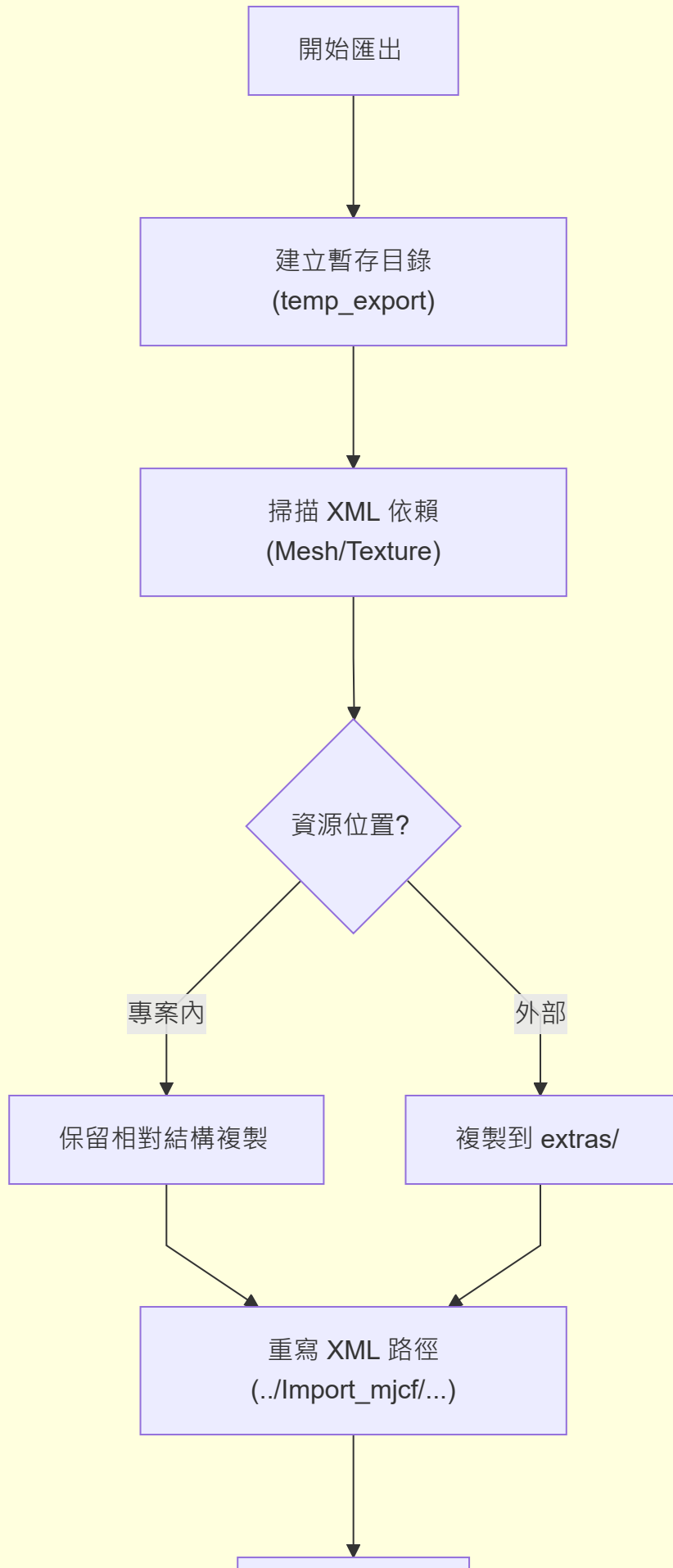
解釋：展示了「運行時副本 (Runtime Copy)」機制，確保原始檔安全。

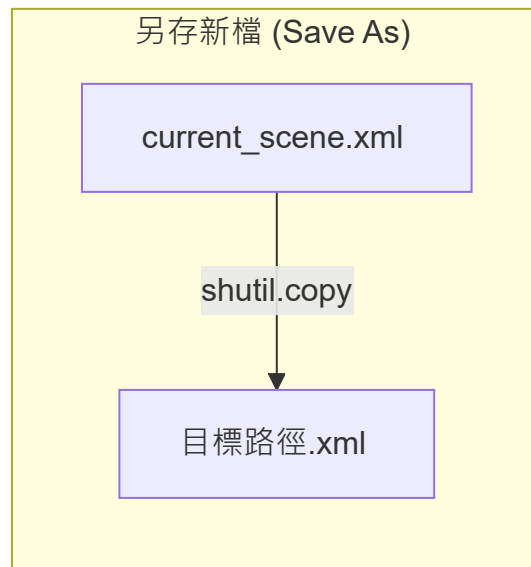
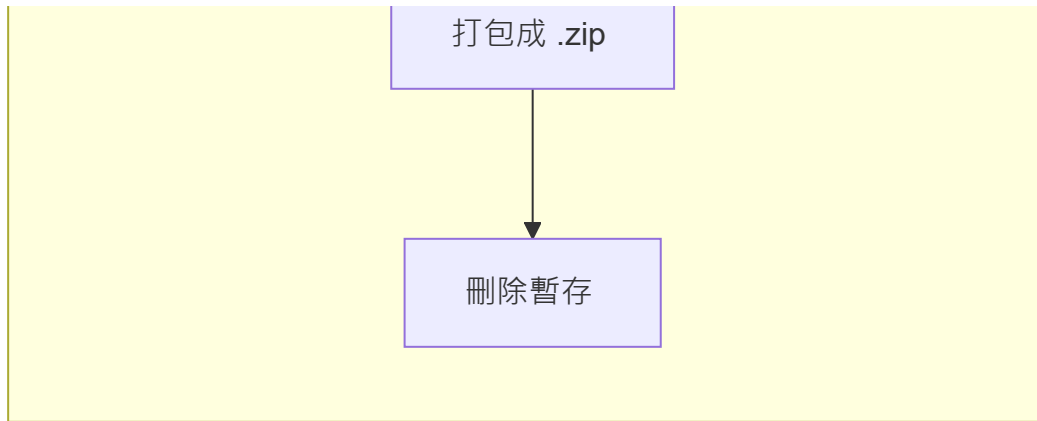


### C. 儲存與匯出 (Save & Export Pipeline)

解釋：展示了如何處理 XML 依賴的路徑問題。

## 匯出專案 (Export Project)





### 3. 物件操作 (Object Manipulation)

#### D. 匯入物件 (Import Object / Add OBJ)

解釋：展示「先轉換、後合併」策略。

階段一：格式轉換

選取 OBJ

subprocess

執行 obj2mjcf

計算慣性/碰撞

生成 MJCF(.xml) +  
Assets(.obj/.stl)

搬移至 Import\_mjcf/

階段二：XML 注入 (Loader)

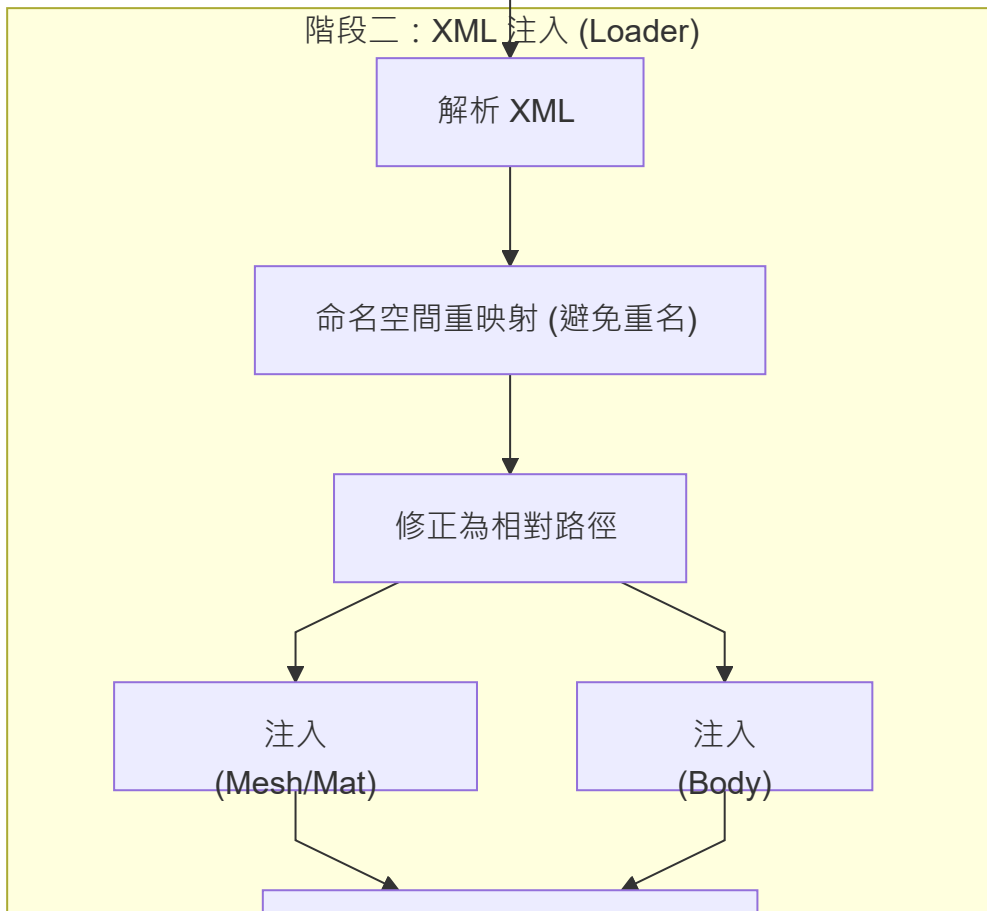
解析 XML

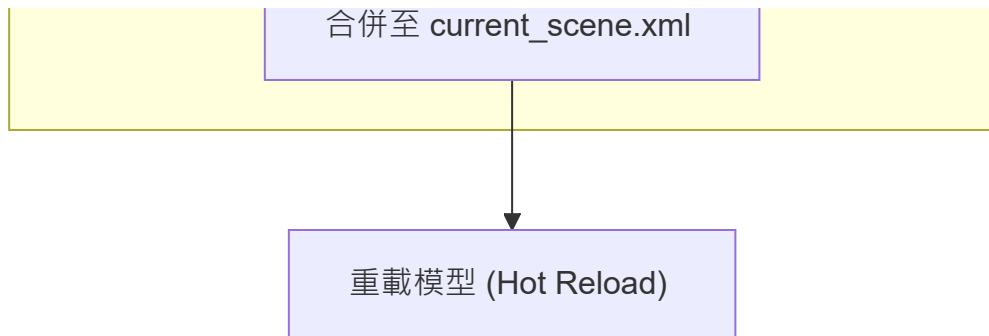
命名空間重映射 (避免重名)

修正為相對路徑

注入  
(Mesh/Mat)

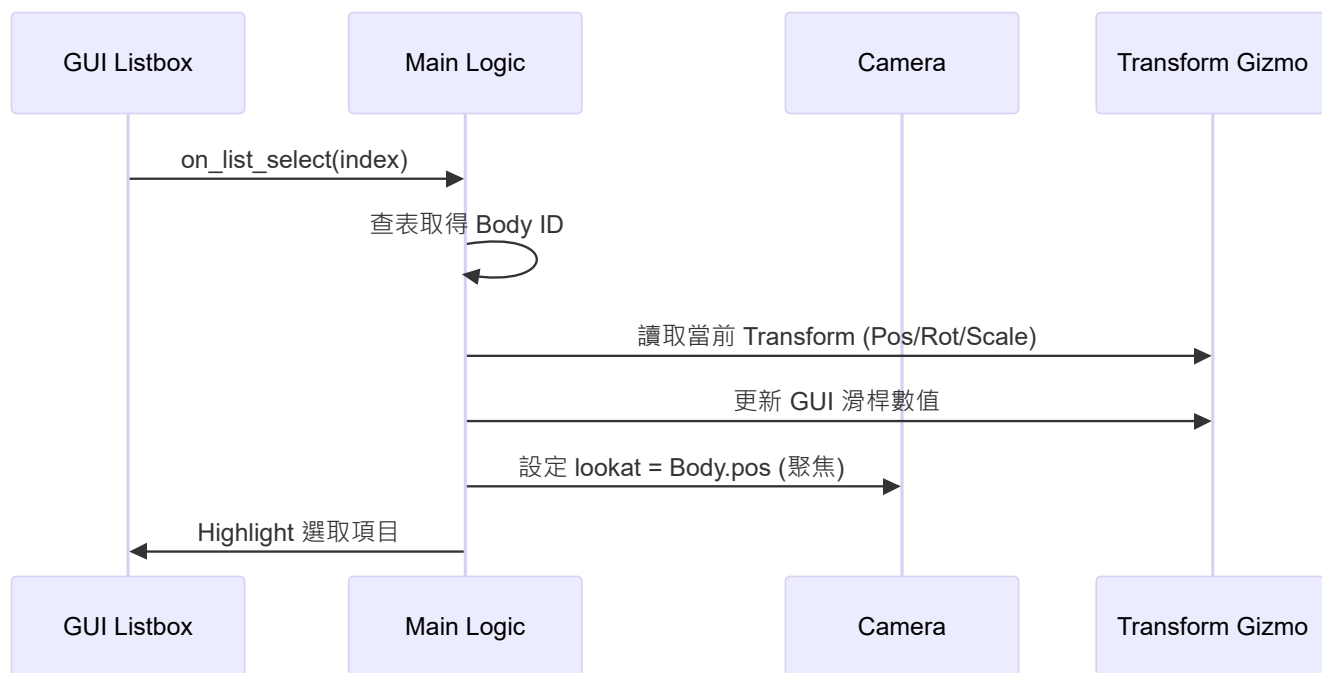
注入  
(Body)





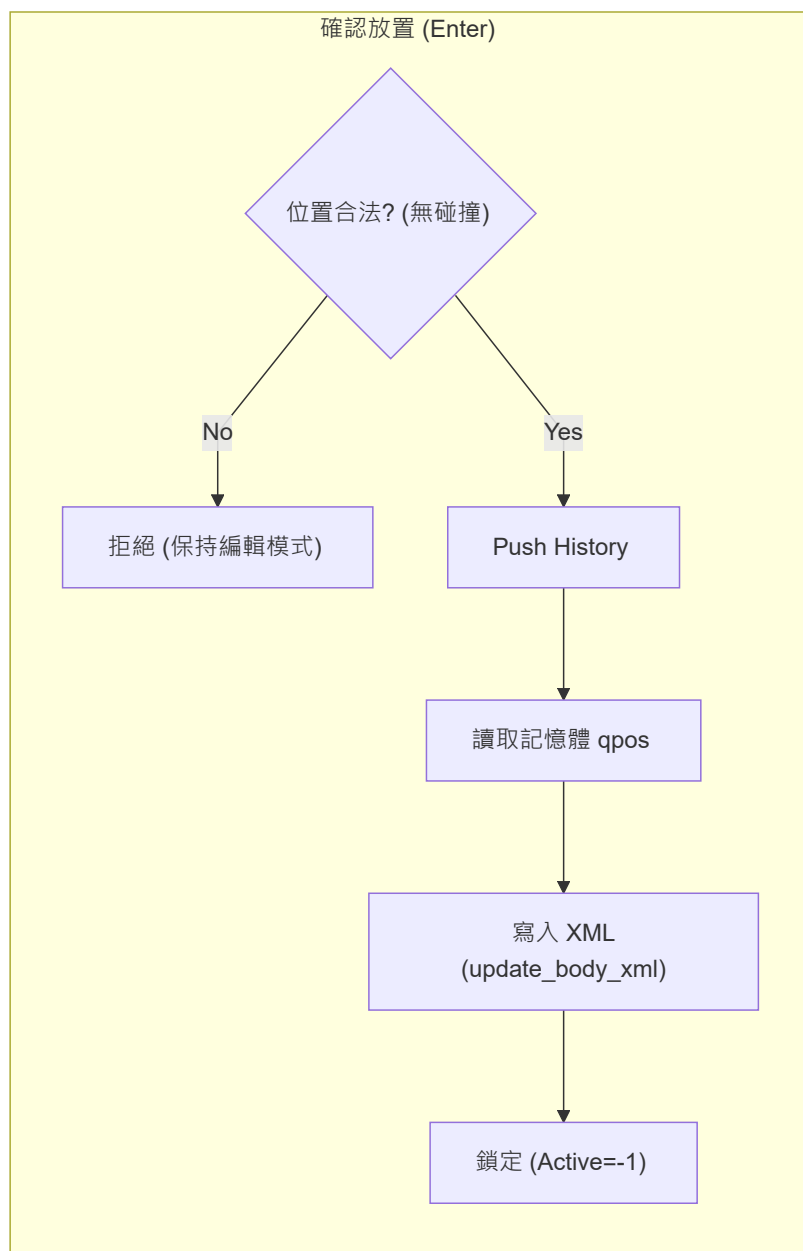
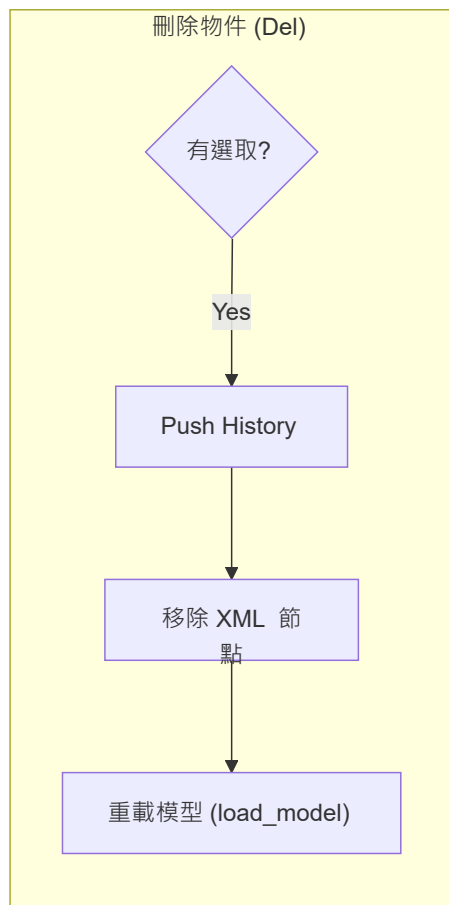
## E. 物件選取與列表 (Object Selection)

解釋：展示 GUI 列表如何反向控制 3D 攝影機。



## F. 確認與刪除 (Confirm & Delete)

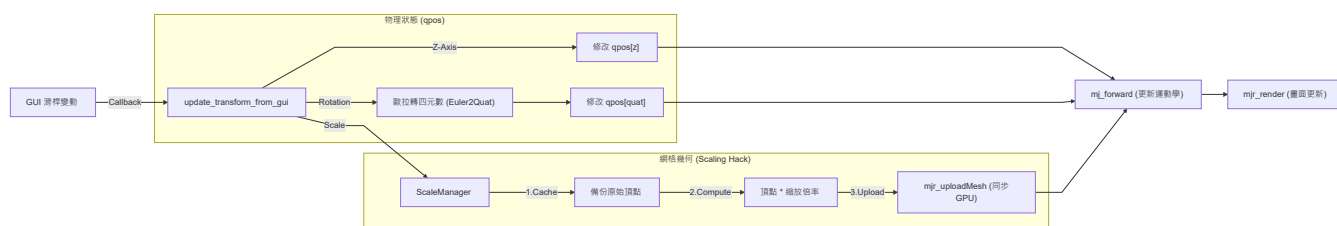
解釋：展示編輯操作如何寫入檔案。



## 4. 變換管線 (Transform Pipeline)

### G. 移動、旋轉與縮放 (Translate, Rotate, Scale)

解釋：這是最複雜的部分，涉及了「縮放 Hack」(修改網格) 與「運動學更新」。

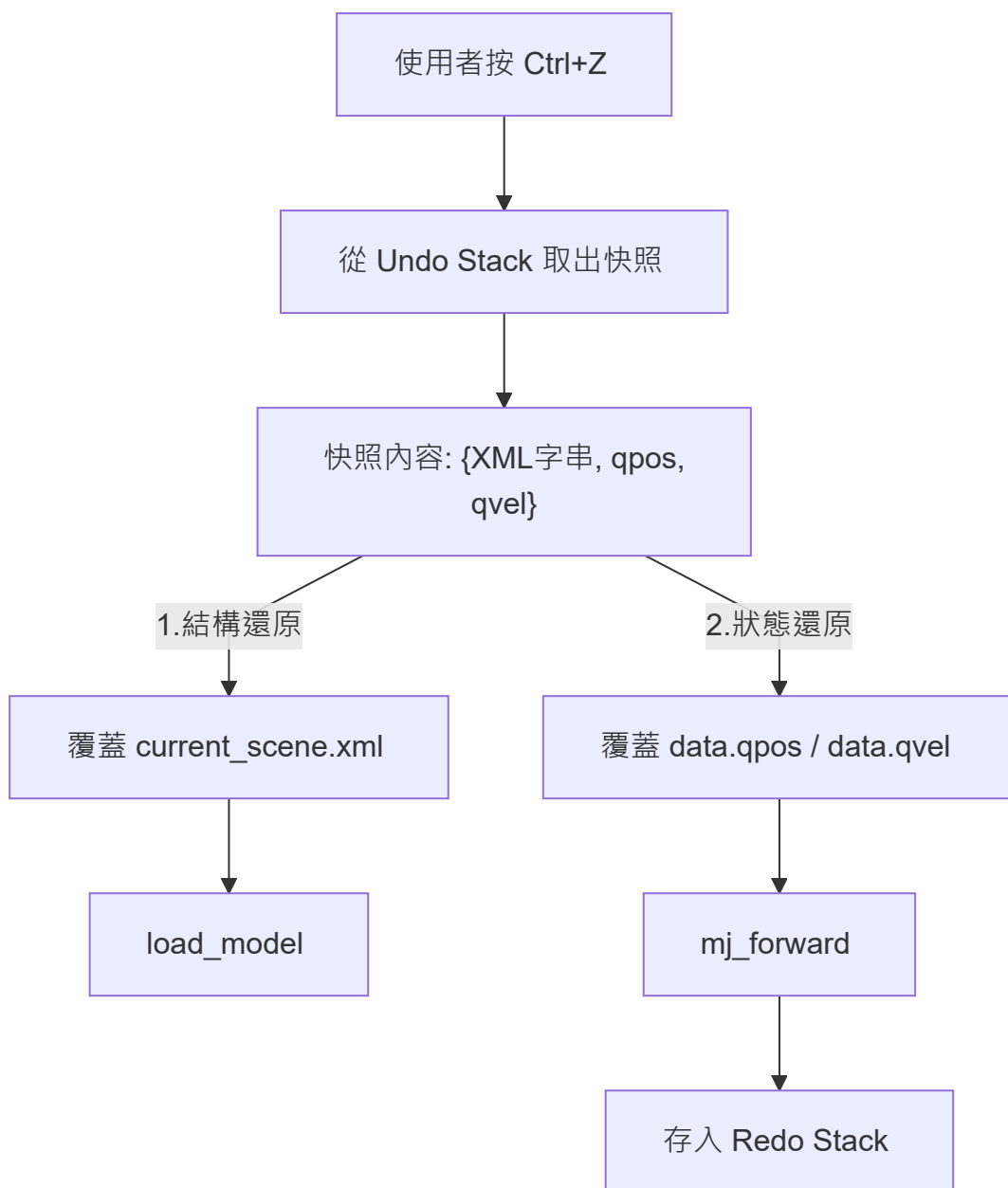




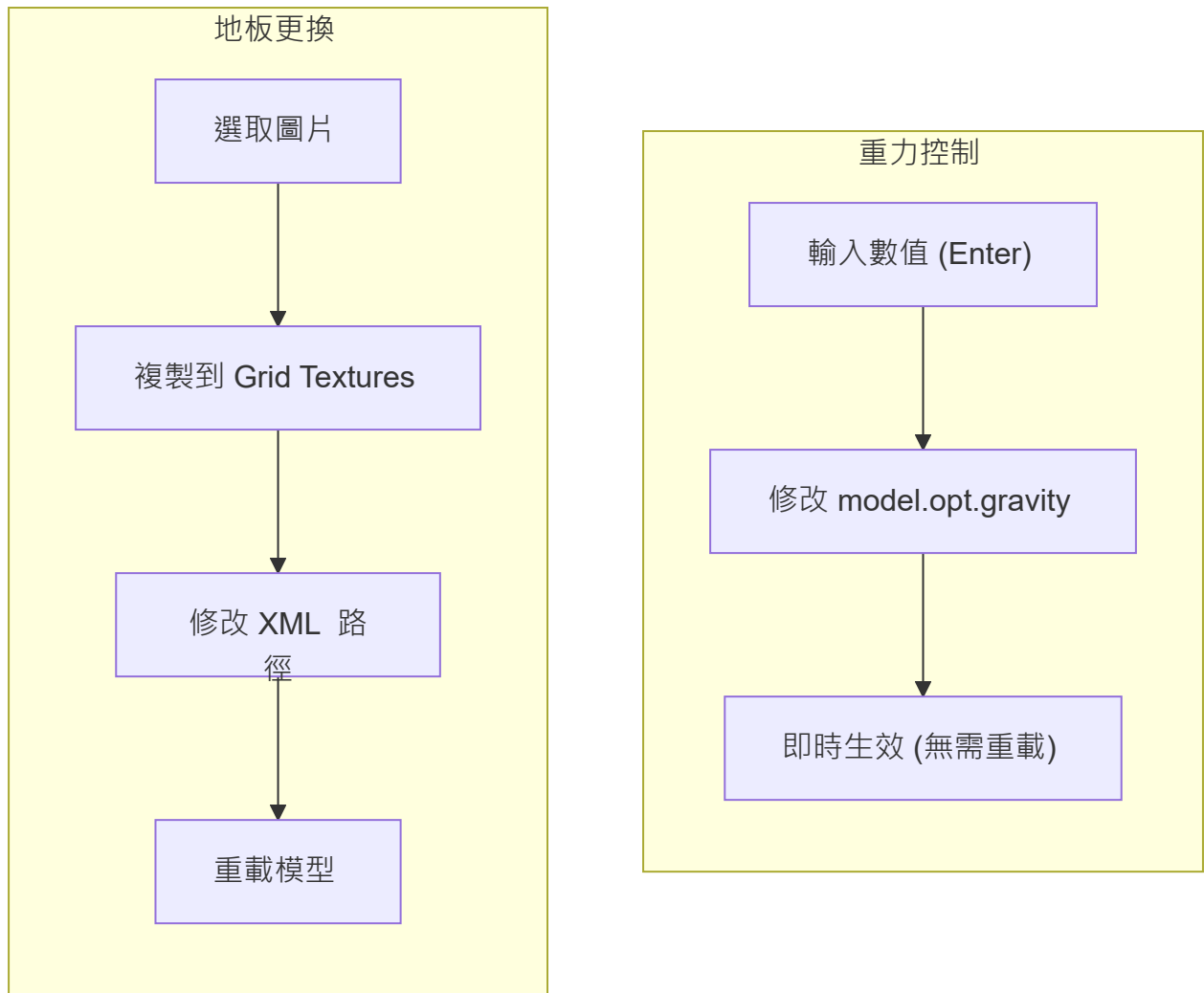
## 5. 歷史與環境 (History & Environment)

### H. 復原與重做 (Undo/Redo)

解釋：展示基於「完整快照 (Full Snapshot)」的歷史系統。



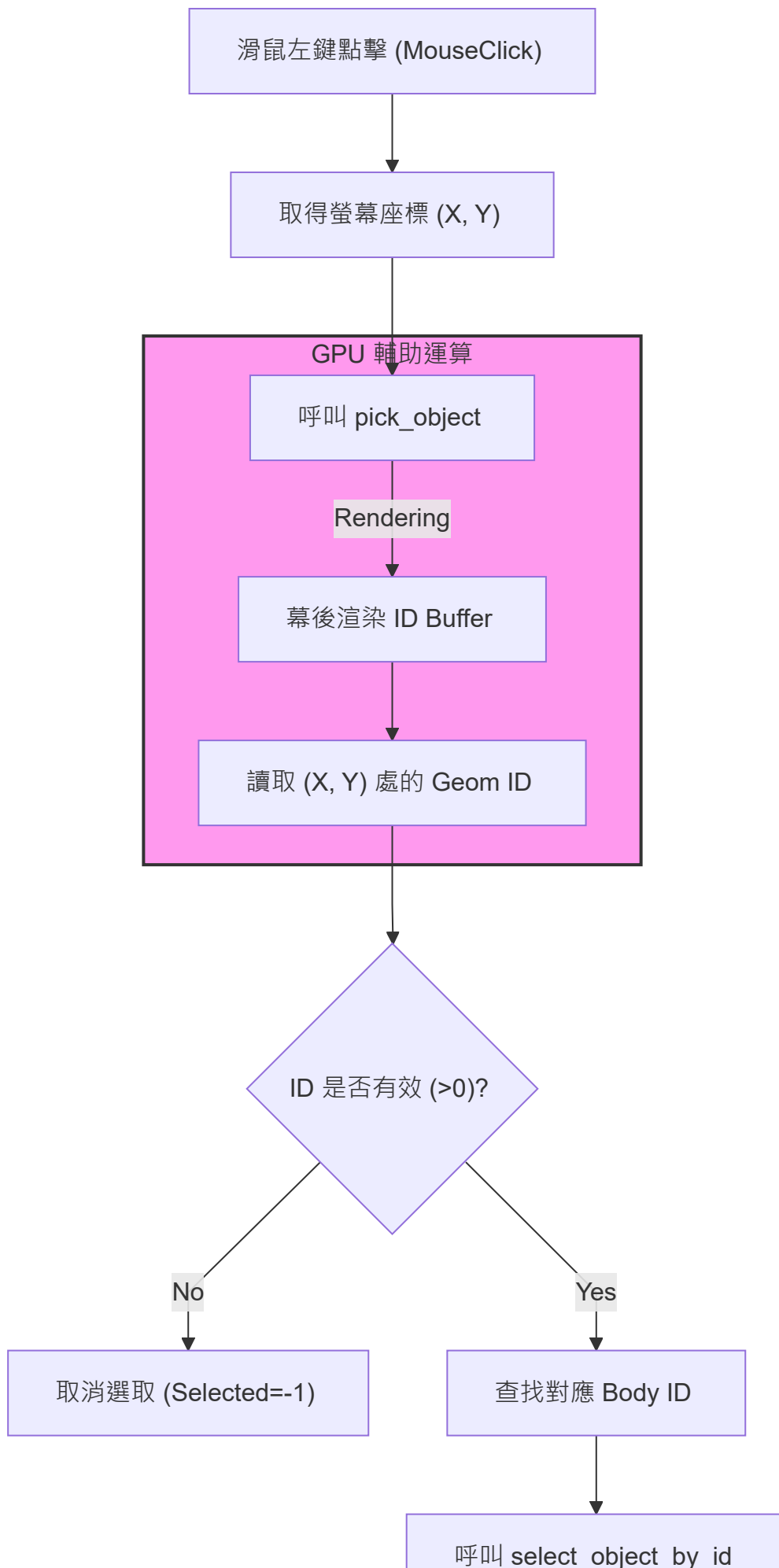
## I. 重力與地板 (Gravity & Floor)

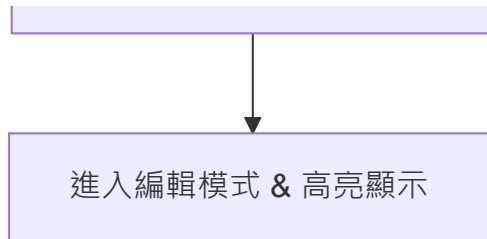


## 6. 滑鼠互動系統 (Mouse Interaction System)

### J. 點擊選取邏輯 (Picking Logic)

**解釋：**展示如何將螢幕上的 2D 像素點轉換為 3D 世界中的物件 ID。MuJoCo 使用 **GPU 輔助選取 (mjv\_select)**，這是最準確的方法。





## K. 拖曳與射線投射邏輯 (Dragging & Raycasting Logic)

**解釋：**這展示了當你拖曳物體時，程式如何計算出它在地板上的位置。這裡使用了 **逆向投影 (Inverse Projection)** 技術，即「從相機射出一道光」。

滑鼠拖曳 (Mouse Drag)

是否在編輯模式?

Yes

射線投射運算 (The 'Light

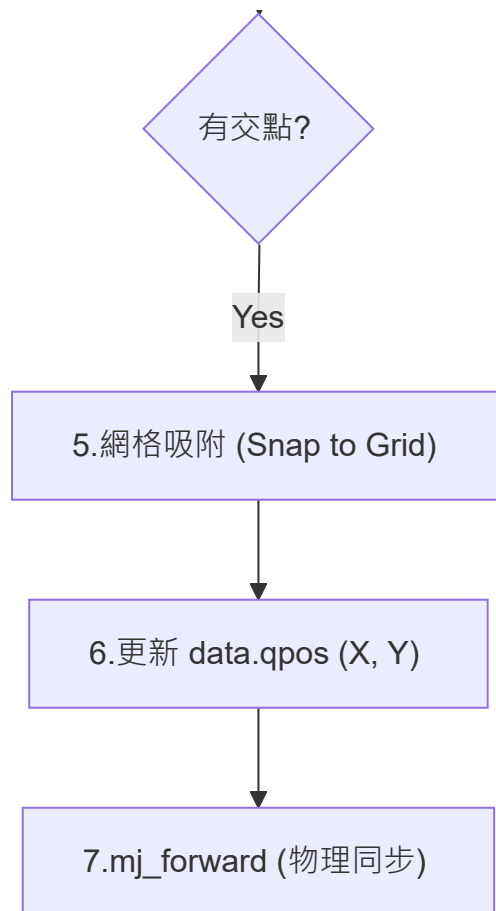
呼叫 raycast\_to\_ground

1. 像素轉歸一化座標  
(Screen -> NDC)

2. 逆向投影 (NDC ->  
Camera Space Ray)

3. 旋轉變換 (Camera ->  
World Ray)

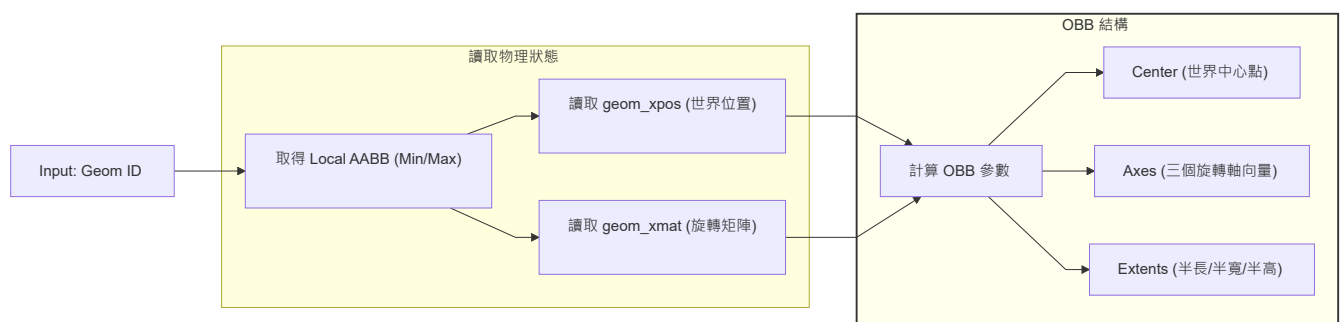
4. 計算與平面 ( $Z=0$ ) 的交點



## 7. 碰撞偵測系統 (Collision Detection System)

### L. OBB (導向包圍盒) 建構管線

解釋：在進行碰撞偵測前，必須先為不規則的物體建立一個數學上容易計算的包圍盒。OBB 是會隨著物體旋轉的盒子。



### M. SAT (分離軸定理) 判斷管線

解釋：這是 `src/managers.py` 中的核心演算法。它透過檢查 15 個軸向的投影重疊，來精確判斷兩個 OBB 是否相撞。

