

專案名稱：基於 Web 與機器視覺的裸眼 3D 展示平台

(Web-based Glasses-free 3D Model Viewer with Head Tracking)

1. 專案目標 (Project Objectives)

本專案旨在開發一個整合「動態網頁技術」與「計算機圖學」的線上 3D 模型展示平台。系統將允許使用者上傳自定義的 3D 模型 (.gltf)，並透過一般的網路攝影機 (Webcam) 捕捉使用者的頭部位置，即時調整渲染視角與投影矩陣，在 2D 螢幕上模擬出具有「運動視差 (Motion Parallax)」的裸眼 3D 窗戶效果。

2. 系統架構 (System Architecture)

本系統採 B/S (Browser/Server) 架構，分為客戶端 (Client) 與伺服器端 (Server)。

技術堆疊 (Tech Stack)

- 前端 (Frontend):
 - UI/Logic: HTML5, CSS3, JavaScript (ES6+)
 - 3D Rendering: Three.js (WebGL wrapper)
 - Shaders: Custom GLSL (Vertex & Fragment Shaders for Blinn-Phong)
 - Computer Vision: face-api.js 或 MediaPipe (Client-side execution)
- 後端 (Backend):
 - Server Language: PHP 8.0+
 - Web Server: Apache
- 資料庫 (Database):
 - RDBMS: MySQL (儲存使用者資訊與模型 metadata)

3. 人員分工表 (Team Roles & Responsibilities)

角色	職責範圍	詳細工作內容
組員 A (圖學工程師)	3D 渲染與 Shader	<ol style="list-style-type: none">搭建 Three.js 場景與 GLTF Loader核心任務：撰寫 GLSL 實現 Blinn-Phong 光照模型 (Ambient + Diffuse + Specular)實作相機視錐體變形 (Off-axis Projection) 邏輯
組員 B (全端工程師)	LAMP 架構與網站功能	<ol style="list-style-type: none">架設 Apache + PHP + MySQL 環境實作會員系統 (註冊/登入/Session)實作檔案上傳功能 (過濾 .gltf, 儲存至 Server)

4. 建置「模型藝廊」頁面供使用者瀏覽

PM (你)	專案管理與視覺追蹤	1. 負責人臉追蹤 API (MediaPipe/face-api) 的研究與實作
(整合與演算法)		2. 將追蹤到的 (x,y) 座標標準化並傳遞給 Three.js
		3. 系統整合：將 3D Canvas 嵌入 PHP 網頁
		4. 進度追蹤與技術支援

4. 系統流程圖 (System Flow)

階段一：使用者上傳 (User Upload Flow)

1. [使用者] 登入網站。
2. [前端] 進入「上傳頁面」，選擇本地端的 .gltf/.glb 檔案。
3. [後端 PHP] 接收檔案 -> 驗證副檔名與大小 -> 將檔案移動至 /uploads/user_{id}/ 目錄。
4. [資料庫] 在 models 資料表中新增一筆紀錄 (包含檔案路徑、上傳者 ID)。

階段二：3D 互動展示 (Interactive Viewing Flow)

1. [使用者] 在「藝廊頁面」點擊某個模型縮圖。
2. [後端 PHP] 導向至 viewer.php?model_id=xyz。
3. [前端 JS] 從後端接收模型路徑 (File Path)。
4. [前端 Three.js] 啟動 WebGL Renderer，載入該 .gltf 模型。
5. [前端 Webcam] 瀏覽器請求攝影機權限 -> 啟動臉部追蹤模型。
6. [互動循環 (Loop)]：
 - 偵測人臉座標 (x, y, z)。
 - 計算相對位移，更新 Camera 的 View Matrix (模擬窗戶視角)。
 - Shader 根據光源計算 Blinn-Phong 顏色。
 - 渲染下一幀畫面。

5. 功能需求詳細列表 (Functional Requirements)

A. 動態網頁部分 (Dynamic Web)

- **R1. 會員機制：** 支援使用者註冊、登入、登出。密碼需經過 Hash 處理 (如 password_hash)。
- **R2. 檔案管理：** * 限制上傳檔案類型為 3D 模型格式 (.gltf, .glb)。
 - 防止惡意檔案上傳。
- **R3. 藝廊展示：** 使用 Grid 佈局顯示所有已上傳的模型，並標註上傳者名稱。

B. 圖學部分 (Computer Graphics)

- **R4. 模型載入**：能正確解析並顯示 GLTF 格式模型。
- **R5. 自定義著色器 (Custom Shader)**：* 不使用 Three.js 內建的 MeshStandardMaterial。
 - 需使用 ShaderMaterial 並在 Fragment Shader 中手寫光照算法。
- **R6. 互動視角 (Motion Parallax)**：* 當使用者頭部向左移，畫面中的物體應相對向右轉 (透視改變)。
 - 需處理 Webcam 影像輸入並轉換為座標數據。