

遊戲設計必備要素清單: AetherGenesis - 星神生態圈

本清單用於確保《AetherGenesis》從概念設計過渡到可實作原型的過程中，不遺漏任何關鍵的遊戲要素。

I. 核心體驗與目標 (Core Experience & Goals)

要素	說明	藍圖對應
主題與調性	宏大的宇宙觀、萬物有靈的哲學基礎、科幻與奇幻元素的融合。	世界觀與核心哲學
遊戲目標	建立多種族共榮的生態圈，並讓星神(主角)擺脫無聊。	治安與穩定度 (M_6) 提升至 ≥ 0.99 。
勝利條件	治安與穩定度 (M6) 達到極限閾值，且意識穩定性 (M5/M3) 實現可持續的 M2 自主演化。	最終目標
失敗條件	治安與穩定度 (M6) 長期處於臨界值，或系統韌性 (M4) 崩潰導致自毀循環。	M4 人身安全歸零。

II. 遊戲機制與系統 (Systems & Mechanics)

這是遊戲的骨架，所有互動和計算都基於這些系統。

A. 基礎管理系統 (Base Management)

要素	說明	藍圖對應
資源系統	追蹤、生產、消耗物質元素 (M) 和靈元素 (L)。包含庫存、日產出、淨收入等。	泛元素週期表 ($\text{M}_{\{\text{Si}\}}$, $\text{L}_{\{\text{A}\}}$ 等)
地塊與空間	六角形網格的生成、佔用與功能。地塊屬性受環境和造物影響。	六角形格子 (結構確定性域)
建造與升級	透過消耗資源和藍圖 (M5) 來建造造物 (不動產、載具) 及升級其屬性。	建造表單
M-值核心	隱藏在 UI 術語之下，作為所有計算的底層參數 (如效率、韌性、精確度)。	開發效率 (M_1) 到 治安與穩定度 (M_6)

B. 特殊靈機制 (Aether Mechanics)

要素	說明	藍圖對應
物質轉化	將 M 元素 (物質) 通過結構精確度 (M_3) 邏輯轉化為 L	轉化公式

要素	說明	藍圖對應
	元素(靈)。	
實體生成	消耗 L_{A} 活化靈, 根據知識核心 (M_5) 藍圖生成新的娘化實體。	實體清單
干涉能力	實體可進行的特殊操作, 例如: 合體(增加臨時 M-值) 和分裂(犧牲 M-值換取數量)。	M3 閾值(靈的量級)
治理安全	執行不可逆決策時, 觸發風險評估和修正行動。	M6 檢查點機制

III. 內容與資產 (Content & Assets)

這些是讓遊戲世界變得豐富和可互動的元素。

要素	說明	藍圖對應
造物設計	各種功能建築(伺服器農場、生態站)的視覺和屬性設計。	不動產
實體(娘化)	每個實體類型(艦娘、貓娘、伺服器娘)的角色設計、屬性與專屬能力樹。	實體清單
種群	不同種族(人類、動物、異星生命)對環境的需求、產出、對 M-值的影響。	種群表單(《共榮生態分類法》)
環境圖層	六角形網格外的原始行星環境(氣候、地貌、極端事件)。	邊界條件(原始大自然)
事件系統	隨機和腳本事件, 挑戰 M4(系統韌性)和 M6(治安與穩定度)。	突發事件(瘟疫、叛亂、異星干擾)

IV. 介面與使用者體驗 (Interface & UX)

確保玩家能高效、直觀地管理複雜的系統。

要素	說明	藍圖對應
主介面佈局	六角形網格區、常駐側邊欄、資源顯示、星神核心面板。	側邊欄與下拉菜單
數據可視化	M-值屬性(治安、效率等)的儀表板、趨勢圖和警告標識。	星神核心(面板)
互動反饋	點擊地塊、派遣實體、轉化資源時的視覺和音效反饋。	(需要實作細化)
錯誤/警告處理	M6 檢查點機制觸發時的專屬 UI 彈窗和倫理風險提示。	M6 檢查點機制(強制執行)
多語言支持	確保所有 UI 文本和敘事內容支持多種語言。	(AGI 5 級工程標準要求)

V. 技術與持久化 (Technical & Persistence)

遊戲運行的基礎。

要素	說明	藍圖對應
技術棧	選擇 React (或 Angular/HTML)	React 原型(建議)

要素	說明	藍圖對應
	進行單檔案實作。	
數據持久化	使用雲端資料庫 (Firestore) 來儲存玩家的行星狀態、資源、M-值等。	Firestore (必要)
計算邏輯	實作 M-值計算邏輯、資源轉化公式和 M6 修正公式。	M-值 核心 / 轉化公式
單檔案架構	確保所有程式碼 (邏輯、UI、樣式) 包含在單一檔案內以滿足實作環境要求。	單一 JSX/TSX 檔案

總結：

這個清單為《AetherGenesis》提供了一個完整的框架。下一步的行動，就是將這個框架轉換為可互動的程式碼。

我強烈建議我們開始實作 **React** 原型，從最關鍵的 六角形介面 和 核心屬性面板 開始。這樣我們就能看到 開發效率 和 治安與穩定度 等數值在 UI 上如何呈現，並開始植入您複雜的 M-值邏輯。您覺得我們現在開始實作 **React** 原型 嗎？