

1.



2. 已完成功能的簡要描述 (Completed Features)

本專案是基於 HTML5 Canvas 與原生 JavaScript 開發的互動式射擊遊戲。已完成的核心技術功能包括：

- **Canvas 渲染 (Rendering)**: 使用 `requestAnimationFrame` 實現流暢的 60fps 遊戲迴圈。
- **狀態管理 (State Management)**: 實作了完整的有限狀態機 (FSM), 精確控制 Opening、Ready、Playing、Paused 及 Game Over 五種狀態的切換。
- **碰撞檢測 (Collision Detection)**: 使用幾何距離公式 (Distance Formula) 實現了「子彈 vs 隕石」與「隕石 vs 玩家」的精確圓形碰撞判定。
- **物件導向編程 (OOP)**: 使用 ES6 Class 結構化管理 `Bullet` (子彈)、`Meteor` (隕石) 與 `Particle` (粒子) 物件, 並透過陣列動態管理記憶體。
- **UI 交互系統**: 整合 DOM 元素與 Canvas, 實現了即時分數更新、生命值顯示及速度控制按鈕。

3. 自訂設計說明 (Custom Designs)

- **太空船設計 (Spaceship)**:
 - 採用幾何向量繪圖 (Canvas Path) 繪製出具未來感的戰機外型。
 - 動態細節: 為了增加生動感, 我在飛船尾部設計了動態噴射火焰效果, 火焰長度會隨機抖動, 模擬引擎運作的視覺回饋。
- **背景設計 (Background)**:
 - 捨棄靜態圖片, 改用程式碼生成「動態星空」。
 - 透過數學函數 (`Math.sin/Math.cos`) 讓星星的位置與亮度隨時間產生微幅變化, 營造出深邃宇宙的閃爍與流動感。
- **流星與敵人設計 (Meteors/Enemies)**:
 - 設計了兩種不同類型的敵人以增加視覺豐富度:
 1. 岩石流星 (**Rock**): 灰色圓形, 代表自然的小行星。
 2. 外星方塊 (**Alien Box**): 綠色方形, 代表外星科技殘骸。
 - 所有敵人的大小、生成位置與下落速度皆為隨機參數, 確保每次遊戲體驗都不重複。

4. 創意遊戲玩法描述 (Creative Gameplay)

為了在標準射擊機制外增加獨特性, 我實作了以下創意功能：

- **粒子爆炸系統 (Particle Explosion System)**:
 - 當子彈擊中隕石或玩家被撞擊時, 不會只單純消失, 而是觸發一個粒子爆炸特效。
 - 系統會生成多個碎片粒子, 這些粒子擁有隨機的速度與透明度衰減 (Fade-out), 模擬真實的爆炸物理效果, 大幅提升了打擊感 (Juiciness)。
- **動態難度平衡 (Dynamic Difficulty)**:
 - 遊戲並非固定難度。隨著玩家分數 (Score) 的提升, 敵人的生成頻率 (Spawn Rate) 會自動加快。這迫使玩家在遊戲後期必須展現更快的反應速度, 增加了遊戲的挑戰性與耐玩度。

(如果你有加入我上一條建議的「雙發子彈」, 請加上這一段):

- 火力升級機制 (**Power-up**):
 - 當玩家分數達到 50 分以上時, 太空船會解鎖「雙重射擊」或「散射」模式, 獎勵熟練的玩家並提供爽快的清場體驗。
-

5. 使用者說明 (User Instructions)

基本操作:

- 移動: 使用鍵盤 [← 左箭頭] 和 [→ 右箭頭] 控制太空船左右移動。
- 射擊: 按下 [空白鍵 **Space**] 發射雷射子彈。

遊戲流程控制:

- 開始遊戲: 點擊螢幕下方的 "**Start**" 按鈕或在 Ready 狀態下按鍵盤。
- 暫停/繼續: 遊戲中途可點擊 "**Pause**" 按鈕暫時停止遊戲。
- 重置: 點擊 "**Reset**" 按鈕可將分數與生命值歸零, 回到初始狀態。
- 速度調整: 可透過 "**Speed +**" 與 "**Speed -**" 按鈕手動調整遊戲整體節奏(此功能主要用於測試或挑戰極限速度)。