โปรแกรมเกม “Galaxy Attack: Alien Shooter” ที่คุณส่งมาให้ ผมจะแยกเป็นหัวข้อชัดเจน พร้อมสรุปว่าแต่ละส่วนทำอะไร

---

1. ขอบเขตของโปรแกรม (Scope)

โปรแกรมนี้เป็น **เกมยิงอวกาศ** (Space Shooter) แบบง่าย ๆ เล่นบนเว็บเบราว์เซอร์ มีฟีเจอร์หลักดังนี้:

1. **ตัวผู้เล่น** (Player)
    - เป็นเครื่องบินอวกาศ 🚀 ใช้คีย์ซ้าย-ขวา เคลื่อนที่
    - ยิงกระสุนด้วย Spacebar หรือ Tab
    - มีชีวิต (lives) จำกัด 3 ครั้ง
2. **ศัตรู** (Enemies)
    - มีหลายแถวและหลายคอลัมน์
    - เคลื่อนที่แบบซิกแซกลงมาช้า ๆ
    - ยิงกระสุนตอบโต้ผู้เล่น
    - ถ้าศัตรูโดนยิงหมดจะเลื่อน Level
3. **ระบบกระสุน** (Bullets)
    - ของผู้เล่น: ยิงขึ้นไป ทำลายศัตรู
    - ของศัตรู: ยิงลงมา ทำลายผู้เล่น
4. **ระบบคะแนนและเลเวล** (Score & Level)
    - คะแนนเพิ่มตามการยิงศัตรู และเวลาที่เล่น
    - ผ่าน Level ถ้าศัตรูทั้งหมดถูกทำลาย
    - เกมจบ (Game Over) ถ้าผู้เล่นชีวิตหมด หรือศัตรูชนฐาน/ผู้เล่น
5. **อินเทอร์เฟซ** (UI)
    - แสดง Lives, Score, Level
    - มี Modal หน้าต่างเริ่มเกมและแสดง Game Over
    - เครื่องบินผู้เล่นเป็นตัว emoji แสดงบนหน้าจอ
6. **เอฟเฟกต์เสียง** (Sounds)

    - ใช้ Web Audio API สร้างเสียง:
        - ยิงปืน
        - ศัตรูถูกยิง
        - ผู้เล่นโดนยิง
        - เกมเริ่ม/จบ
7. **ฉากหลัง** (Background)
    - ดาวเคลื่อนไหวเล็ก ๆ แบบ Blink
8. **รองรับการปรับขนาดหน้าจอ**
    - Canvas ขยายเต็มหน้าจอ
    - ปรับขนาดเมื่อ Resize

---

2. โครงสร้างของโปรแกรม (Structure)

โค้ดสามารถแบ่งเป็น **โมดูล/ส่วนหลัก** ได้ดังนี้:

2.1 HTML

- โครงสร้าง DOM หลัก:

```
<canvas id="gameCanvas"></canvas>  <!-- สำหรับวาดเกม -->
<div id="playerPlane">✈</div>      <!-- emoji เครื่องบิน -->
<div id="ui"> ... </div>           <!-- Lives, Score, Level -->
<div id="modal"> ... </div>        <!-- Modal เริ่ม/จบเกม -->
```

2.2 CSS

- จัด Layout และ Style:
    - Canvas เต็มหน้าจอ
    - Modal overlay สำหรับเริ่มเกมและ Game Over
    - Emoji เครื่องบินเคลื่อนไหวได้

2.3 JavaScript

JavaScript ถูกจัดเป็น **ส่วนหลัก ๆ**:

2.3.1 Setup Canvas

```
const canvas = document.getElementById('gameCanvas');
const ctx = canvas.getContext('2d');
```

- ตั้งค่า canvas ขนาดเต็มหน้าจอ
- รองรับการปรับขนาดหน้าจอ

2.3.2 Game Variables

- ตัวแปรควบคุมเกม:
    - `gameRunning`, `score`, `lives`, `level`
    - `player` (x, y, bullets)
    - `enemies` array
    - `stars` array (ฉากหลัง)
    - `enemyBullets` array

2.3.3 Player Controls

- ฟังเหตุการณ์ `keydown` / `keyup`:
    - ArrowLeft / ArrowRight = เคลื่อนที่
    - Space / Tab = ยิงกระสุน

2.3.4 Enemy Setup

- สร้างศัตรูหลายแถวหลายคอลัมน์
- เก็บค่า:
    - ตำแหน่ง x, y
    - ทิศทาง (ซ้าย/ขวา)
    - สถานะ `alive`

2.3.5 Shooting Mechanics

- Player bullets: เคลื่อนที่ขึ้น, ตรวจ Collision กับศัตรู
- Enemy bullets: เคลื่อนที่ลง, ตรวจ Collision กับผู้เล่น

2.3.6 Update & Draw

- `update()`
    - คำนวณตำแหน่งผู้เล่น, ศัตรู, กระสุน, ตรวจ Collision

    - อัพเดต Stars และคะแนน
- `draw()`
    - วาด Stars, bullets, enemies

2.3.7 Player Plane (Emoji)

- เคลื่อนตาม `player.x` และ `player.y`
- หมุนตามทิศทางที่กด

2.3.8 UI Update

- `updateLives()`: แสดงจำนวนชีวิต
- `document.getElementById('score').innerText` : แสดงคะแนน
- `document.getElementById('level').innerText` : แสดง Level

2.3.9 Sound Functions

- `playTone(frequency, duration, type)` : เล่นเสียงแบบ Waveform
- ฟังก์ชันเฉพาะ:
    - `soundStartGame()`, `soundShootBullet()`, `soundHitEnemy()`, `soundHitPlayer()`, `soundGameOver()`

2.3.10 Game Over & Level Complete

- `triggerGameOver(message)` : แสดง Modal, หยุดเกม
- ตรวจสอบ Level Complete แล้วเพิ่ม Level

2.3.11 Game Loop

```
function gameLoop(){
    update();
    draw();
    updatePlayerPlane();
    if(gameRunning) requestAnimationFrame(gameLoop);
}
```

2.3.12 Start Game

- เริ่มเกมเมื่อกดปุ่ม `#startBtn`
- รีเซ็ตตัวแปร, สร้างศัตรู, เริ่มเสียงและ loop

---

3. แผนภาพโครงสร้าง (Simplified Diagram)

```
HTML
 ├─ <canvas id="gameCanvas">      --> วาดฉากเกม
 ├─ <div id="playerPlane">         --> emoji เครื่องบิน
 ├─ <div id="ui">                  --> Lives, Score, Level
 └─ <div id="modal">               --> Start / Game Over

CSS
 └─ Layout + style (canvas, modal, plane, UI)

JavaScript
 ├─ Variables
 │   ├─ player
 │   ├─ enemies
 │   ├─ bullets
 │   ├─ enemyBullets
 │   └─ stars
 ├─ Controls (keyboard)
 ├─ Sounds (Web Audio API)
 ├─ Functions
 │   ├─ initEnemies()
 │   ├─ shootBullet()
 │   ├─ enemyShoot()
 │   ├─ update()
 │   ├─ draw()
 │   ├─ updatePlayerPlane()
 │   ├─ updateLives()
 │   └─ triggerGameOver()
 └─ Game Loop & Start Game
     ├─ requestAnimationFrame(gameLoop)
     └─ setInterval(enemyShoot)
```

สรุปคือ **โครงสร้างเกมนี้ค่อนข้างเป็นโมดูลชัดเจน**:

- HTML = UI / Layout
- CSS = การจัดวาง + style
- JS = Logic ทั้งหมด (Player, Enemy, Bullets, Collision, UI, Sound, Game Loop)