

# DOCUMENTACIÓN DE ASSETS - SPACE DEFENDER

**Versión:** 1.0

**Fecha:** Febrero 2026

## 1. Introducción a Assets

Los assets son todos los recursos visuales, de audio, modelos 3D y otros medios utilizados en el videojuego Space Defender. Esta documentación proporciona un inventario completo de todos los assets, sus especificaciones técnicas y su propósito en el juego.

Space Defender es un juego estilo Galaga/Space Invaders desarrollado con Babylon.js y TypeScript, donde el jugador controla una nave espacial que debe defender la galaxia de oleadas de enemigos alienígenas y un jefe final.

## 2. Inventario de Assets

### 2.1 Resumen por Tipo

Tipo de Asset	Cantidad	Formatos	Tamaño Total Aprox.
Modelos 3D	1	GLB	50-500 KB
Sprites Enemigos	3	PNG	100-300 KB
Audio (opcional)	3	MP3	300-600 KB
VFX (Partículas)	1	Generado	N/A
Fondo Espacial	1	Generado	N/A
Total	9	Mixtos	~500 KB - 1.5 MB

## 3. Assets Visuales - Nave del Jugador

### 3.1 Modelo de la Nave Principal

**Archivo:** nave.glb

**Ubicación:** /public/models/nave.glb

**Especificaciones Técnicas:**

- Formato: GLB (glTF Binary)
- Tipo: Modelo 3D
- Escala en juego: 0.05 (5% del tamaño original)
- Motor de renderizado: Babylon.js

**Descripción Visual:**

Modelo 3D de nave espacial del jugador. Diseño futurista con capacidad de disparo. La nave se posiciona en la parte inferior de la pantalla y puede moverse horizontalmente.

#### Elementos del Modelo:

- Cuerpo principal de la nave
- Detalles de cabina
- Propulsores traseros
- Alas laterales
- Texturas y materiales integrados

#### Propósito en el Juego:

Representación visual del jugador controlable. Aparece durante todo el gameplay. Posición inicial: Centro inferior de la pantalla (Y: -8).

#### Implementación en Código:

```
const result = await SceneLoader.ImportMeshAsync(
    "",
    "/models/",
    "nave.glb",
    this.scene,
);
// Escala ajustada para el juego
mesh.scaling = new Vector3(0.05, 0.05, 0.05);
```

#### Controles Asociados:

- Flechas izquierda/derecha o A/D: Movimiento horizontal
- Espacio: Disparar proyectiles

#### Notas:

- Si el modelo no carga, se genera una nave de respaldo con geometría básica
- La nave parpadea en rojo cuando recibe daño (invulnerabilidad temporal)

## 4. Assets Visuales - Enemigos

### 4.1 Enemigo Nivel 1 - Básico

**Archivo:** enemiesLV1.png

**Ubicación:** /public/models/enemiesLV1.png

#### Especificaciones Técnicas:

- Formato: PNG
- Resolución: Variable (sprite 2D)
- Modo de Color: RGBA (con transparencia)
- Tamaño en juego: 3.0 unidades

#### Descripción Visual:

Sprite de enemigo alienígena básico. Diseño pixel art o estilizado. Color distintivo para diferenciarse de otros niveles de enemigos.

**Propósito en el Juego:**

Enemigo más común y fácil de derrotar. Aparece en las primeras oleadas del juego. Otorga 100 puntos al ser destruido.

**Comportamiento:**

- Patrón de movimiento: Zigzag horizontal, onda sinusoidal, o diagonal
- Capacidad de disparo hacia el jugador
- Se elimina con un solo impacto

**Notas:** Tipo 0 en el sistema de enemigos (row % 3 === 0).

**4.2 Enemigo Nivel 2 - Medio**

**Archivo:** enemiesLV2.png

**Ubicación:** /public/models/enemiesLV2.png

**Especificaciones Técnicas:**

- Formato: PNG
- Resolución: Variable (sprite 2D)
- Modo de Color: RGBA (con transparencia)
- Tamaño en juego: 3.2 unidades

**Descripción Visual:**

Sprite de enemigo alienígena de nivel intermedio. Diseño ligeramente más amenazante que el nivel 1. Color distintivo (diferente al LV1).

**Propósito en el Juego:**

Enemigo de dificultad media. Aparece mezclado con enemigos básicos. Otorga 100 puntos al ser destruido.

**Comportamiento:**

- Patrones de movimiento similares al LV1
- Misma resistencia (un impacto)
- Ligeramente más grande visualmente

**Notas:** Tipo 1 en el sistema de enemigos (row % 3 === 1).

**4.3 Enemigo Nivel 3 - Fuerte**

**Archivo:** enemiesLV3.png

**Ubicación:** /public/models/enemiesLV3.png

**Especificaciones Técnicas:**

- Formato: PNG
- Resolución: Variable (sprite 2D)
- Modo de Color: RGBA (con transparencia)
- Tamaño en juego: 3.4 unidades (enemigo) / 8.0 unidades (boss)

### **Descripción Visual:**

Sprite de enemigo alienígena de nivel avanzado. Diseño más intimidante y detallado. Colores más oscuros o vibrantes para indicar peligro.

### **Propósito en el Juego:**

- Como enemigo: Enemigo más fuerte en oleadas normales
- Como Boss: Jefe final que aparece al alcanzar 2000 puntos

### **Comportamiento como Enemigo:**

- Patrones de movimiento variados
- Un impacto para destruir
- Otorga 100 puntos

### **Comportamiento como Boss:**

- Tamaño: 8.0 unidades (el doble que enemigos normales)
- Vida: 10 puntos de salud
- Disparo: Un proyectil cada 2 segundos
- Movimiento: Horizontal lento
- Recompensa: 1000 puntos + Victoria del juego

### **Notas:**

- Tipo 2 en el sistema de enemigos (row % 3 === 2)
- El boss usa este mismo sprite pero con escala mayor y tinte rojo

## **5. Assets de Audio**

### **5.1 Sistema de Audio**

El juego implementa un sistema de audio dual:

- Sonidos personalizados: Archivos MP3 opcionales
- Sonidos generados: Web Audio API como respaldo

**Ubicación de archivos:** /public/sounds/

### **5.2 Sonido de Disparo**

**Archivo:** disparo.mp3 (opcional)

**Ubicación:** /public/sounds/disparo.mp3

### **Especificaciones Técnicas:**

- Formato: MP3
- Bitrate recomendado: 128 kbps
- Duración: 0.1-0.3 segundos
- Volumen en juego: 30%

### **Descripción de Audio:**

Efecto de sonido de disparo láser/energético. Sonido corto y satisfactorio que proporciona retroalimentación al jugador.

**Sonido Generado (Fallback):**

```
// Web Audio API - Onda cuadrada
oscillator.type = 'square';
oscillator.frequency: 800Hz → 200Hz (0.1s)
```

**Propósito en el Juego:**

Retroalimentación auditiva cada vez que el jugador dispara un proyectil.

### 5.3 Sonido de Explosión

**Archivo:** explosion.mp3 (opcional)

**Ubicación:** /public/sounds/explosion.mp3

**Especificaciones Técnicas:**

- Formato: MP3
- Bitrate recomendado: 128 kbps
- Duración: 0.2-0.5 segundos
- Volumen en juego: 40%

**Descripción de Audio:**

Efecto de explosión cuando un enemigo es destruido o el jugador recibe daño. Sonido impactante con decay progresivo.

**Sonido Generado (Fallback):**

```
// Web Audio API - Ruido filtrado
Tipo: Buffer de ruido blanco
Filtro: Lowpass 1000Hz → 100Hz
Duración: 0.3 segundos
```

**Propósito en el Juego:**

- Destrucción de enemigos
- Impacto de proyectiles enemigos en el jugador
- Colisiones entre enemigos y jugador
- Derrota del boss (explosión grande)

### 5.4 Música de Fondo

**Archivo:** musica\_fondo.mp3 (opcional)

**Ubicación:** /public/sounds/musica\_fondo.mp3

**Especificaciones Técnicas:**

- Formato: MP3

- Bitrate recomendado: 128-192 kbps
- Duración: Loop continuo
- Volumen en juego: 20%
- Reproducción: Loop infinito

#### Descripción de Audio:

Música ambiental estilo arcade/retro que acompaña el gameplay. Debe ser energética pero no distractora.

#### Música Generada (Fallback):

```
// Melodía simple arcade
Notas: C, E, G, E, C, G, E, C (262Hz, 330Hz, 392Hz...)
Tipo de onda: Triangle
Intervalo: 300ms entre notas
```

#### Propósito en el Juego:

- Ambiente sonoro durante el gameplay
- Se detiene en Game Over
- Se reinicia al comenzar nueva partida

## 6. Assets de Efectos Visuales (VFX)

### 6.1 Sistema de Partículas - Explosiones

**Tipo:** Generado proceduralmente

**Motor:** Babylon.js ParticleSystem

#### Especificaciones Técnicas:

- Partículas por explosión: 50-100
- Textura: Gradiente radial generado por canvas
- Colores: Amarillo (#FFD700) → Naranja (#FF4500) → Rojo oscuro (#8B0000)
- Duración: 0.2-0.5 segundos por partícula
- Blend mode: Additive (efecto de brillo)

#### Tipos de Explosiones:

Tipo	Tamaño	Uso
Normal	1.0x	Dstrucción de enemigos
Pequeña	0.5x	Chispas de impacto
Grande	2.0x	Derrota del boss

#### Propósito en el Juego:

Retroalimentación visual satisfactoria cuando se destruyen enemigos o el boss.

**Implementación:**

```
explosionSystem.createExplosion(position, size);
explosionSystem.createSparks(position);
explosionSystem.createLargeExplosion(position);
```

**6.2 Fondo Espacial**

**Tipo:** Generado proceduralmente

**Motor:** Babylon.js MeshBuilder

**Especificaciones Técnicas:**

- Estrellas: 200 esferas pequeñas
- Tamaño de estrellas: 0.03-0.18 unidades
- Posición Z: 3-8 unidades (detrás del gameplay)
- Color de fondo: RGB(0.02, 0.02, 0.08) - Azul muy oscuro

**Colores de Estrellas:**

- Blanco (brightness, brightness, brightness)
- Azul claro (brightness0.8, brightness0.9, brightness)
- Amarillo (brightness, brightness, brightness\*0.7)

**Propósito en el Juego:**

Crear ambiente espacial inmersivo sin distraer del gameplay.

**7. Assets de UI (Interfaz de Usuario)**

**7.1 Elementos de Interfaz**

**Elemento Tipo Posición Color**

Elemento	Tipo	Posición	Color
LIVES	Texto HTML	Superior izquierda	Blanco
SCORE	Texto HTML	Superior derecha	Blanco
Boss Health Bar	Div HTML	Inferior al LIVES	Rojo/Verde
GAME OVER	Overlay HTML	Centro	Rojo
YOU WIN	Overlay HTML	Centro	Verde

**7.2 Barra de Vida del Boss**

**Especificaciones:**

- Ancho: 200px
- Alto: 20px
- Color barra: Gradiente rojo (#FF0000 → #FF4444)
- Borde: 2px sólido blanco
- Animación: Transición suave al reducir

**Estados:**

- 100% - 60%: Normal
- 60% - 30%: Fase agresiva
- 30% - 0%: Fase enfurecida

**8. Especificaciones de Resolución y Escala**

**8.1 Dimensiones del Juego**

Elemento	Valor	Descripción
Bounds X	-8 a +8	Límites horizontales
Bounds Y	-10 a +8	Límites verticales
Cámara	Ortográfica	Vista 2D desde arriba
Ortho Size	10 unidades	Tamaño de vista

**8.2 Escalas de Entidades**

Entidad	Escala/Tamaño
Nave jugador	0.05 (5%)
Enemigo LV1	3.0 unidades
Enemigo LV2	3.2 unidades
Enemigo LV3	3.4 unidades
Boss	8.0 unidades
Proyectil	0.1 x 0.4 unidades

**9. Formatos de Archivo y Justificación**

**9.1 Justificación de Formatos**

Tipo	Formato	Razón
------	---------	-------



<b>Modelo 3D</b>	GLB	Formato binario glTF, compacto y rápido de cargar
<b>Sprites</b>	PNG	Soporta transparencia, ideal para sprites 2D
<b>Audio</b>	MP3	Amplio soporte en navegadores, buen balance calidad/tamaño
<b>Partículas</b>	Canvas	Generación dinámica, sin archivos externos necesarios

## 9.2 Compatibilidad

- Navegadores: Chrome, Firefox, Edge, Safari (modernos)
- WebGL: Requerido para Babylon.js
- Web Audio API: Para sonidos generados

## 10. Estructura de Archivos

/Andrew's Space Defender/

```

├── public/
│   ├── models/
│   │   ├── nave.glb      # Modelo 3D del jugador
│   │   ├── enemiesLV1.png # Sprite enemigo básico
│   │   ├── enemiesLV2.png # Sprite enemigo medio
│   │   └── enemiesLV3.png # Sprite enemigo fuerte/boss
│   ├── sounds/
│   │   ├── disparo.mp3   # (Opcional) Sonido de disparo
│   │   ├── explosion.mp3 # (Opcional) Sonido de explosión
│   │   └── musica_fondo.mp3 # (Opcional) Música de fondo
│   └── textures/         # (Reservado para futuras texturas)
├── src/
│   ├── entities/
│   │   ├── Player.ts     # Carga nave.glb
│   │   ├── Enemy.ts      # Carga enemiesLV1-3.png
│   │   ├── Boss.ts       # Usa enemiesLV3.png escalado
│   │   └── Projectile.ts  # Geometría generada

```

```

|   └─ managers/
|   |   └─ AudioManager.ts   # Gestión de audio
|   |   └─ ScoreManager.ts   # Gestión de puntaje
|   └─ systems/
|   |   └─ ParticleSystem.ts  # Explosiones VFX
|   |   └─ ...
|   └─ scenes/
|       └─ GameScene.ts       # Fondo espacial generado
└─ index.html                 # UI y estilos

```

## 11. Gestión de Assets

### 11.1 Control de Versiones

Asset	Versión	Fecha	Cambios
nave.glb	1.0	Feb 2026	Modelo inicial
enemiesLV1.png	1.0	Feb 2026	Sprite creado
enemiesLV2.png	1.0	Feb 2026	Sprite creado
enemiesLV3.png	1.0	Feb 2026	Sprite creado
AudioManager	1.0	Feb 2026	Sistema de audio con fallback
ParticleSystem	1.0	Feb 2026	Sistema de explosiones

### 11.2 Consideraciones Futuras

- Nuevos enemigos: Agregar más tipos de enemigos con sprites únicos
- Power-ups: Sprites para mejoras (escudo, disparo múltiple, etc.)
- Animaciones: Sprites animados para enemigos y explosiones
- Niveles: Diferentes fondos espaciales por nivel/oleada
- Música: Agregar pistas de audio reales para mejor inmersión
- Boss adicionales: Nuevos jefes con diseños únicos

## 12. Requisitos de Carga y Rendimiento

### 12.1 Tiempos de Carga Estimados

Asset	Tamaño	Tiempo (3G)	Tiempo (4G/WiFi)
-------	--------	-------------	------------------

<b>nave.glb</b>	~200KB	2s	<0.5s
<b>Sprites (3)</b>	~150KB	1.5s	<0.3s
<b>Audio (3)</b>	~450KB	4s	<1s
<b>Total</b>	<b>~800KB</b>	<b>~7.5s</b>	<b>&lt;2s</b>

## 12.2 Optimizaciones Implementadas

- Fallback a geometría básica si el modelo 3D falla
- Sonidos generados por Web Audio API si no hay archivos MP3
- Partículas generadas proceduralmente (sin texturas externas)
- Precarga de assets antes del inicio del juego

## 13. Notas Sobre Derechos y Atribuciones

- Modelo nave.glb: Asset proporcionado por el desarrollador
- Sprites de enemigos: Assets proporcionados por el desarrollador
- Sonidos generados: Creados programáticamente, sin licencia necesaria
- Babylon.js: Licencia Apache 2.0

Se recomienda documentar la fuente específica de cada asset si fueron obtenidos de bibliotecas de recursos externos.

## Fin de la Documentación de Assets

Space Defender - Un juego estilo arcade desarrollado con Babylon.js