

GDD – Andrew's Space Defender: 3070

I. Ficha Técnica y Concepto

- **Título del Juego:** *Andrew's Space Defender: 2750* (Generado a partir del nombre del proyecto y la ambientación en el año 2750).
- **Género:** Shooter Espacial (Tipo "Galaga").
- **Plataforma:** Navegador Web (implementado con tecnología Babylon.js).
- **Target (Público Objetivo):** Jóvenes.
- **Elevator Pitch:**

"En el año 3070, la Tierra enfrenta su mayor amenaza. Toma el control del piloto Andrew y su nave de la Armada Espacial para detener la invasión alienígena en este intenso shooter arcade. Sobrevive a oleadas enemigas, esquiva proyectiles letales y domina el espacio en una experiencia que revive la nostalgia de los clásicos con la tecnología web moderna."

II. Análisis MDA (El Núcleo del Diseño)

Aesthetics (Estética - La experiencia)

¿Qué emociones buscamos evocar en el jugador?

- **Tensión y Reto:** Provocada por la dificultad del juego y la amenaza constante de perder una de las 3 vidas disponibles.
- **Nostalgia:** Evocación de recuerdos de los juegos clásicos arcade del género.

Dynamics (Dinámicas - El comportamiento)

¿Cómo interactúan las reglas para crear esas emociones?

- **Gestión de Disparo Automático (Toggle):** El disparo se activa y desactiva con una sola pulsación de la tecla espacio. Esto genera una dinámica donde el jugador puede concentrarse más en el movimiento y la esquiva mientras la nave dispara sola.
- **Oportunidad y Recolección:** La aparición ocasional de *boosts* (mejoras) de velocidad y aumento de balas crea momentos de empoderamiento temporal frente a la dificultad base.
- **Presión Temporal (Hit & Run):** Dado que los enemigos aparecen por un tiempo limitado y luego desaparecen, el jugador se ve forzado a ser agresivo para eliminarlos antes de que huyan si quiere obtener el "High Score".

Mechanics (Mecánicas - Las reglas base)

Definición resumida de las reglas que soportan las dinámicas:

- **Movimiento:** Desplazamiento en el eje horizontal utilizando las teclas WASD y flechas del teclado.
- **Sistema de Vidas:** El jugador posee 3 vidas; el impacto de una nave o proyectil enemigo resta una vida.
- **Condición de Victoria/Derrota:** Se gana el nivel al derrotar a todos los enemigos; se pierde si se agotan las vidas.
- **Sistema de Oleadas:** Los enemigos entran por los lados de la pantalla y permanecen un tiempo determinado en el mapa.

Aquí tienes la continuación del documento (Partes III, IV y V), completadas según la información del *Blueprint* y los estándares de diseño solicitados.

III. Mecánicas Detalladas (Game System Design)

- **Core Loop (Ciclo de Juego):** El ciclo principal se define por la siguiente secuencia:
 1. **Inicio:** El jugador aparece en el nivel con el contador de vidas al máximo (3) y el scroll vertical del fondo activo.
 2. **Combate:** Aparecen enemigos progresivamente entrando por los laterales de la pantalla.
 3. **Temporizador de Enemigo:** Cada enemigo permanece en el mapa un tiempo limitado; si no es destruido, desaparece (huye).
 4. **Resolución:**
 - **Victoria:** Se derrota a la totalidad de los enemigos asignados al nivel.
 - **Derrota:** El jugador pierde sus 3 vidas por impacto de naves o proyectiles.
- **Controles (Inputs):**
 - **Movimiento:** Teclas WASD y Flechas del Teclado (Arriba, Abajo, Izquierda, Derecha).

- **Disparo:** Tecla Espacio. Funciona como un interruptor (*toggle*): se presiona una vez para activar el disparo automático y se presiona nuevamente para desactivarlo (no es necesario mantenerla pulsada).
- **Matemáticas del Sistema (Valores Estándar Propuestos):** Dado que se solicita definir estándares coherentes:
 - **Velocidad del Jugador:** 10 unidades/segundo.
 - **Velocidad de Proyectil:** 20 unidades/segundo (debe ser más rápido que la nave).
 - **Tiempo de Vida del Enemigo:** 5 a 8 segundos en pantalla antes de desaparecer.
 - **Salud del Jugador:** 3 Vidas (muerte instantánea al contacto).
 - **Probabilidad de Power-Up:** 5% de probabilidad al destruir un enemigo de soltar un *boost* (velocidad o munición).
- **Sistemas de Puntuación y Progresión:**
 - **Puntos:** +1 punto por cada enemigo derrotado.
 - **High Score:** Se otorga una bonificación de "Puntuación Perfecta" solo si el jugador logra eliminar al 100% de los enemigos antes de que huyan o termine el nivel.

IV. Narrativa y Mundo (Worldbuilding)

Contexto narrativo que justifica la acción del juego.

- **Sinopsis y Guion:** Corre el año **3070**. La Tierra se encuentra bajo asedio por una flota de naves alienígenas hostiles que buscan la invasión total. La humanidad ha desplegado su última línea de defensa en la órbita terrestre para repeler el ataque.
- **Personajes:**
 - **Andrew:** Protagonista y piloto de élite de la Armada Espacial. Es el encargado de pilotar la nave defensora y proteger el planeta de la amenaza inminente.
 - **Los Invasores:** Naves alienígenas (sin nombre específico aún) que actúan como la fuerza antagonista.

V. Diseño de Niveles (Level Design)

Estructura del entorno y progresión de la dificultad.

- **Layouts de Mapas:** Actualmente no existen mapas visuales definidos. La estructura espacial se basa en un fondo con scroll vertical infinito.
 - **Patrón de Aparición:** Los enemigos no tienen posiciones estáticas iniciales; entran dinámicamente por los lados de la pantalla hacia el área de juego.
- **Objetivos por Nivel:** El objetivo principal es la Supervivencia. El jugador debe evitar perder sus 3 vidas a causa de impactos directos con naves o proyectiles enemigos. El objetivo secundario para maximizar la puntuación es destruir a todos los enemigos antes de que abandonen la pantalla.
- **Flujo de Dificultad:** Se define una curva de dificultad progresiva:
 1. **Fase 1 (Introducción):** Enemigos lentos que aparecen de uno en uno. Sirve para que el jugador se adapte al movimiento y al sistema de disparo automático.
 2. **Fase 2 (Escalada):** Aumento de la frecuencia de aparición (dos enemigos simultáneos) y reducción del tiempo que permanecen en pantalla, obligando al jugador a moverse más rápido.
 3. **Fase 3 (Clímax):** Aparición de oleadas mixtas y presencia de enemigos con mayor velocidad de proyectil, culminando con la aparición de un "Jefe" o nave más resistente al final de la secuencia.

Aquí tienes las **Partes VI y VII** del documento, generadas estrictamente con la información disponible en el *Design Blueprint* y las especificaciones técnicas de *Galaga Mechanics*, dejando abiertos los espacios solicitados.

VI. Arte y Audio (Look & Feel)

Se define la base técnica de los recursos audiovisuales, dejando la dirección artística abierta para futuras iteraciones según las instrucciones del proyecto.

- **Tecnología de Assets:**
 - **Visuales:** Uso de **Sprites** para la nave del jugador, enemigos y balas. Implementación de fondos con **scroll vertical** para simular el avance espacial.

- **Efectos Especiales (VFX):** Uso de **ParticleSystem** (Babylon.js) para generar las explosiones al destruir enemigos.
- **Audio:**
 - Implementación técnica mediante el objeto Sound de Babylon para efectos sonoros.
- **Estilo Visual:** [Por definir - Espacio reservado para futura especificación].
- **Paleta de Colores:** [Por definir - Espacio reservado para futura especificación].

VII. Arquitectura de Software (Ingeniería)

Esta sección establece las herramientas base de ingeniería. Los detalles estructurales específicos (diagramas y lógica profunda) se mantienen abiertos según lo indicado en el plan de diseño.

- **Tecnología (Motor):**
 - **Babylon.js** (Framework para navegador web).
- **Componentes Técnicos Clave:**
 - SpriteManager: Para la gestión optimizada de múltiples entidades 2D (balas y enemigos).
 - ParticleSystem: Para la gestión de efectos visuales.
 - Sound: Para la gestión de audio.
- **Estructura Lógica (Diagramas y Patrones):** [Por definir - Espacio reservado. La información sobre diagramas de clases, estados, patrones de diseño y persistencia de datos está pendiente de definición].