

AETHERIA: REIGN OF MAGIC

Documento de Diseño de Juego (GDD) Completo

1. INFORMACIÓN BÁSICA

- Título:** Aetheria: Reign of Magic
- Género:** Estrategia en Tiempo Real / Simulación de Ecosistemas Mágicos
- Plataforma:** PC (Windows, Mac, Linux)
- Motor:** Unity 2022.3 LTS
- Público Objetivo:** Jugadores de estrategia, simulación y fantasía (16+)
- Clasificación ESRB:** T (Teen)

2. CONCEPTO Y VISIÓN

2.1 Concepto Central

Aetheria es un juego de simulación estratégica que modela un ecosistema mágico dinámico donde dos facciones en conflicto (Alianza de la Magia vs Legión de la Corrupción) compiten por el control del territorio mediante la manipulación de sistemas naturales mágicos interconectados.

2.2 Propósito Único

Crear un ecosistema vivo donde cada decisión del jugador genere efectos en cadena a través de múltiples sistemas acoplados, generando comportamientos emergentes únicos en cada partida.

2.3 Hipótesis de Diseño Central

"Un ecosistema mágico tiende a la estabilidad si el flujo de maná supera la tasa de corrupción durante al menos tres ciclos consecutivos"

Métrica Global:

$$\text{Estabilidad} = (\sum \text{densidad_maná} - \sum \text{nivel_corrupcion}) / \text{total_celdas} > 0.6$$

3. MECÁNICAS PRINCIPALES

3.1 Sistema de Facciones

Alianza de la Magia

- Genera y protege maná
- Construye Santuarios
- Unidades: Novice Mages
- Victoria por estabilidad

Legión de la Corrupción

- Expande y mantiene corrupción
- Construye Pozos Corruptores
- Unidades: Corrupt Slaves
- Victoria por dominación

3.2 Sistema de Recursos

- **Maná:** Recurso de la Alianza, generado por tierra mágica y estructuras
- **Corrupción:** Recurso de la Legión, expandido por terreno corrupto y estructuras
- **Límite de Edificios:** Máximo 10 estructuras por facción

3.3 Sistema de Construcción

- **Santuarios** (Costo: 100 Maná) - Suprime corrupción en radio 7
- **Pozos Corruptores** (Costo: 100 Corrupción) - Expande corrupción en radio 5

3.4 Sistema de Unidades

- **Novice Mage** (Costo: 50 Maná) - Suprime corrupción, ataca estructuras
- **Corrupt Slave** (Costo: 50 Corrupción) - Expande corrupción, ataca estructuras

4. ARQUITECTURA TÉCNICA

4.1 Stack Tecnológico

- **Motor:** Unity 2022.3 LTS
- **Lenguaje:** C#
- **Input System:** Unity Input System 1.7.0
- **UI Framework:** Unity UI + TextMeshPro
- **Audio:** Unity Audio Mixer

4.2 Patrones de Diseño Implementados

- **Singleton:** Managers principales (GameManager, GridManager, UIManager)
- **Observer:** Sistema de eventos entre simulaciones
- **Component:** Entidades del juego (unidades, edificios, criaturas)
- **State Machine:** Comportamientos de unidades y criaturas

5. SISTEMAS DE SIMULACIÓN

5.1 SIMULACIÓN 1: Sistema Integrado Maná-Corrupción

Scripts Principales:

- `ManaFlowSimulation.cs`
- `CorruptionSimulation.cs`
- `GridManager.cs`

Parámetros de Balance:

// Mana Flow

radiInfluencia = 1

umbralAislamiento = 2

```
umbralCristalizacion = 3  
probNacimiento = 0.3  
densidadInicialMana = 0.1  
densidadInicialArboles = 0.01
```

```
// Corrupción
```

```
tasaBase = 0.25
```

```
factorMana = 0.4
```

```
radioSantuario = 5
```

```
fuerzaSupresion = 0.15
```

Estados Celulares:

```
public enum CellState
```

```
{  
    TierraNormal = 0,    // Blanco  
    TierraMagica = 1,    // Azul  
    CristalMagico = 2,   // Cyan  
    ArbolAncestral = 3   // Verde  
}
```

Reglas del Sistema Integrado:

- Los árboles ancestrales generan maná y suprimen corrupción
- La corrupción degrada el maná existente
- Santuarios crean zonas de supresión de corrupción
- Pozos corruptores aceleran la expansión de corrupción

5.2 SIMULACIÓN 2: Sistema de Eventos Mágicos

Script Principal: `MagicEventsSimulation.cs`

Eventos Implementados:

- **Oleada de Maná:** Convierte tierra normal en mágica, crea cristales
- **Floración Corrupta:** Expande corrupción rápidamente
- **Ola de Purificación:** Reduce corrupción globalmente
- **Terremoto Mágico:** Cambios caóticos en el grid

Parámetros de Frecuencia:

`probabilidadOleadaMana = 0.2f`

`probabilidadFloracionCorrupta = 0.2f`

`probabilidadOlaPurificacion = 0.15f`

`probabilidadTerremotoMagico = 0.1f`

Características:

- Sistema de notificaciones UI con efectos visuales
- Efectos de sonido específicos por evento
- Screen shake para terremotos
- Partículas y efectos visuales 2D

5.3 SIMULACIÓN 3: Ecosistema de Fauna Mágica

Scripts Principales:

- `MagicalFaunaSimulation.cs`
- `CreatureBehavior.cs`
- `EcosystemManager.cs`

Especies Implementadas:

- **Lumispark:** Atraído por maná, repele corrupción
- **Shadowling:** Atraído por corrupción, repele maná
- **Neutral Spirit:** Se mantiene en zonas equilibradas

Atributos por Especie:

Especie	Salud	Energía	Velocidad	Visión
Lumispark	50	100	1.5	4
Shadowling	80	120	1.0	3
Neutral Spirit	100	150	1.2	5

Comportamientos:

- Movimiento por gradientes ambientales
- Consumo de energía y muerte natural
- Efectos en el entorno (generar/suprimir maná/corrupción)
- Sistema de audio para comportamientos

5.4 SIMULACIÓN 4: Sistema de Construcción y Gestión

Scripts Principales:

- `BuildingSystem.cs`
- `Sanctuary.cs`
- `Corruptor.cs`

- `GameManager.cs`

Características:

- Dos facciones con recursos separados
- Límites de construcción por facción
- Efectos territoriales de edificios
- Generación pasiva de recursos
- Restricciones de terreno para construcción

5.5 SIMULACIÓN 5: Sistema de Unidades Autónomas

Scripts Principales:

- `UnitManager.cs`
- `UnitBehavior.cs`

Unidades Implementadas:

- **Novice Mage:** Suprime corrupción, comportamiento defensivo
- **Corrupt Slave:** Expande corrupción, comportamiento ofensivo

Atributos de Unidades:

salud = 100

daño = 10

velocidadMovimiento = 2f

rangoAccion = 3f

rangoVision = 6f

Comportamientos IA:

- Movimiento hacia objetivos estratégicos

- Toma de decisiones periódica
- Ataque automático a estructuras enemigas
- Acciones de área según facción

6. INTERCONEXIÓN DE SISTEMAS

6.1 Matriz de Dependencias

Sistema Origen	Dato	Sistema Destino	Efecto
ManaFlow	gridManá	Corruption	Acelera/ralentiza expansión
Corruption	gridCorrupcion	ManaFlow	Degrada estados de maná
Ambos	Estado Grid	Fauna	Spawn y movimiento criaturas
Events	Cambios masivos	Todos	Modificaciones temporales del grid
Buildings	Efectos locales	Ambos	Santuarios/Pozos afectan áreas
Units	Acciones	Ambos	Unidades modifican territorio
Fauna	Efectos locales	Ambos	Criaturas afectan entorno

6.2 Flujo de Ejecución por Frame

1. Input Manager procesa controles

2. Game Manager verifica estado global
3. Simulaciones actualizan (si no en pausa)
4. UI Manager refresca interfaz
5. Audio Manager gestiona sonidos

7. INTERFAZ DE USUARIO

7.1 Arquitectura UI

Script Principal: `UIManager.cs`

Paneles Implementados:

- **Panel de Simulación:** Controles de todas las simulaciones
- **Panel de Pausa:** Menú de pausa con navegación
- **HUD:** Recursos, facción, contadores

7.2 Sistema de Controles

Controles por Simulación:

- Velocidad de simulación (sliders)
- Densidades iniciales (sliders)
- Auto-simulación (toggles)
- Parámetros de comportamiento (sliders)

7.3 Navegación

- **Teclado:** ESC (pausa), TAB (controles), R (reiniciar)
- **Gamepad:** Start (pausa), Select (controles)
- Navegación con gamepad en todos los menús

8. SISTEMA DE AUDIO

8.1 Arquitectura de Audio

Script Principal: [AudioManager.cs](#)

Canales de Audio:

- **Música:** Ambiente orquestal
- **SFX:** Efectos de juego
- **Ambiente:** Sonidos ambientales

8.2 Implementación

```
public class AudioManager : MonoBehaviour
{
    public AudioSource musicSource;
    public AudioSource sfxSource;
    public AudioSource ambientSource;

    // Control de volumen por categoría
    public void SetMasterVolume(float volume)
    public void SetMusicVolume(float volume)
    public void SetSFXVolume(float volume)
}
```

9. SISTEMA DE INPUT

9.1 Arquitectura

Script Principal: [InputManager.cs](#)

Sistema Implementado:

- Unity Input System 1.7.0
- Soporte para teclado y gamepad
- Eventos unificados para toda la aplicación

9.2 Mapping de Controles

Acción	Teclado	Gamepad	Evento
Pausa	ESC	Start	OnPause
Controles	TAB	Select	OnToggleControls
Reiniciar	R	North Button	OnRestart
Mover Cámara	WASD/Flechas	Left Stick	OnCameraMove
Zoom	Scroll Wheel	Triggers	OnCameraZoom

10. BALANCE Y PROGRESIÓN

10.1 Métricas de Balance

KPIs por Simulación:

Simulación	KPI Principal	Rango Óptimo	Fórmula
ManaFlow	Estabilidad	60-80%	(celdas_estables / total) * 100

Corruption	Tasa Expansión	<0.2 c/ciclo	Δ celdas_corruptas / Δ tiempo
Fauna	Diversidad	>2 especies	count(poblacion > 5)
Buildings	Eficiencia	>1.0	(beneficios - costos) / ciclos
Units	Efectividad	>0.5	(acciones_efectivas / total)

10.2 Estados del Sistema

- **ARMONÍA:** Maná estable, corrupción contenida, diversidad alta
- **CAOS:** Corrupción dominante, poblaciones colapsando
- **ESTANCAMIENTO:** Equilibrio frágil, baja productividad
- **CRECIMIENTO:** Expansión controlada, recursos en aumento

11. CONTENIDO Y ASSETS

11.1 Prefabs Implementados

- **Edificios:** Sanctuary, Corruptor
- **Unidades:** NoviceMage, CorruptSlave
- **Criaturas:** Lumispark, Shadowling, NeutralSpirit
- **Efectos:** Partículas para eventos mágicos

11.2 Sistema Visual

- **Grid:** Celdas con colores por estado
- **Overlay:** Corrupción como tintado rojo

- **Partículas:** Efectos para eventos y acciones

12. OPTIMIZACIÓN Y RENDIMIENTO

12.1 Estrategias Implementadas

- **Update Management:** Timers independientes por simulación
- **Corrutinas:** Para actualizaciones lentas (generación de recursos)
- **Pooling:** Criaturas y unidades
- **LOD Visual:** Efectos escalables por distancia

12.2 Targets de Rendimiento

- **FPS:** 60fps en hardware medio
- **Grid Size:** 50x30 celdas (óptimo balanceado)
- **Entidades Máximas:** 20 criaturas + 10 unidades
- **Memory:** <500MB RAM

13. PLAN DE PRUEBAS

13.1 Casos de Prueba

1. **Transiciones de Estado:** Armonía → Caos → Recuperación
2. **Eventos Extremos:** Múltiples eventos simultáneos
3. **Stress Test:** Máximas entidades + eventos
4. **Balance Test:** Partidas prolongadas (60+ minutos)

13.2 Métricas de Validación

- **Estabilidad:** No crashes en sesiones de 2+ horas
- **Balance:** Ninguna facción domina consistentemente
- **Rendimiento:** FPS estable durante eventos
- **UX:** Controles intuitivos en primeros 5 minutos

14. ROADMAP DE DESARROLLO

Fase 1: Núcleo (COMPLETADO)

- Sistema de simulaciones acopladas
- UI y controles básicos
- Sistema de audio

Fase 2: Contenido (ACTUAL)

- Mas tipos de criaturas y unidades
- Eventos mágicos adicionales

Fase 3: Pulido

- Optimización de rendimiento
- Balance fino
- Depuración final

15. DOCUMENTACIÓN TÉCNICA ADICIONAL

15.1 Estructura de Archivos

Assets/

|— Scripts/

— Managers/	# GameManager, UIManager, AudioManager
— Simulations/	# Las 5 simulaciones principales
— Entities/	# Unidades, Edificios, Criaturas
— Systems/	# Input, Camera, UI
— Prefabs/	# Entidades preconfiguradas
— Audio/	# Música y efectos de sonido
— Materials/	# Shaders y materiales visuales

15.2 Convenciones de Código

- **Variables públicas:** Usar [Header] para organización
- **Eventos:** Usar System.Action para comunicación entre sistemas
- **Singletons:** Patrón implementado con Instance property
- **Comentarios:** Documentación en inglés para consistencia

16. CONCLUSIÓN

Aetheria: Reign of Magic representa una implementación completa de un sistema de simulación complejo donde múltiples subsistemas interconectados generan comportamientos emergentes únicos. La arquitectura modular permite extensibilidad futura mientras mantiene un rendimiento estable y una experiencia de usuario cohesiva.

El proyecto demuestra exitosamente la viabilidad de modelar ecosistemas dinámicos mediante autómatas celulares, sistemas multiagente y modelos de difusión, todo integrado en una experiencia de juego estratégica accesible pero profunda.

Versión del Documento: 1.0

Última Actualización: Diciembre 2024

Estado: IMPLEMENTACIÓN COMPLETADA - FASE DE PULIDO