

Game Design Document

Borg Cataclysm

Defensa del Cuadrante Alfa - Proyecto Academico POO

1. High Concept

1.1 Frase Principal

Un shooter arcade 2D donde el jugador pilota naves de la Federacion para contener una invasion masiva del Colectivo Borg, utilizando un sistema de armas modular y progresion tactica.

1.2 Descripcion Ampliada

Borg Cataclysm combina la nostalgia de los clasicos 'bullet-hell' con una arquitectura moderna orientada a objetos. El jugador enfrenta oleadas procedurales de naves Borg (Scouts, Spheres y Cubes). El nucleo del juego reside en la gestion del cooldown de armas y la recoleccion de power-ups para sobrevivir a una dificultad que escala exponencialmente. Visualmente, el juego adopta la estetica LCARS (Library Computer Access and Retrieval System) de Star Trek para su interfaz de usuario.

2. Pilares de Diseno y Tecnologia

2.1 Pilares

- Accion Inmediata: Controles precisos y respuesta visual instantanea.
- Dificultad Dinamica: Los enemigos escalan en vida, dano y velocidad por nivel.
- Fidelidad Tematica: Atmosfera basada en Star Trek

2.2 Stack Tecnologico

- Lenguaje: C++17.
- Framework Grafico: SFML 3.0.0.
- Arquitectura: POO Pura (Herencia, Polimorfismo, Composicion).
- Patrones: Singleton (UITheme), Strategy (Callbacks de disparo), Factory (Proyectiles).

3. Historia y Universo

El Colectivo Borg ha abierto un conducto transcurvatura cerca de la estacion Deep Space Nine. Como ultima linea de defensa, el jugador debe pilotar prototipos de la Federacion para ganar tiempo.

3.1 Facciones y Amenazas

- Borg Scout: Rapido, poca vida, disparos lineales simples.
- Borg Sphere: Resistencia media, disparos curvos de plasma.
- Borg Cube (Boss): Gran capacidad de absorcion de dano y patrones de ataque multiples.

4. Gameplay Core

4.1 Sistema de Armas (Modular)

El sistema se basa en la clase base Weapon, permitiendo disparar diferentes entidades Projectile:

- LaserLauncher: Disparos rapidos de energia lineal.
- MissileLauncher: Projectiles autopulsados con aceleracion constante.

4.2 Movimiento y Defensa

El jugador opera en el 20% inferior de la pantalla. Posee un sistema de Escudo (Shield) que se regenera o recupera mediante power-ups antes de afectar la Integridad Estructural (HP).

4.3 Power-Ups

- Rapid Fire: Reduce el cooldown global de las armas.
- Double Shot: Duplica el output de proyectiles por disparo.
- Invulnerabilidad: Campo de fuerza temporal (efecto visual de parpadeo).

5. Arquitectura Tecnica (POO)

Este proyecto demuestra el uso avanzado de objetos segun la consigna de la catedra:

5.1 Jerarquia de Entidades

Uso de clase abstracta Entity que hereda de sf::Drawable y sf::Transformable. De ella derivan SpaceShip y Projectile, implementando polimorfismo en los metodos update() y draw().

5.2 Gestion de Recursos

Implementacion de un ResourceManager para carga centralizada de texturas y sonidos, optimizando el uso de memoria RAM evitando duplicados.

5.3 Interfaz de Usuario (UI)

Uso del patron Singleton en UITheme para garantizar que todos los elementos de la interfaz compartan la misma identidad visual LCARS de forma eficiente.

6. Persistencia y High Scores

Al finalizar la partida (Game Over), el sistema interactua con el sistema de archivos (fstream):

- Lectura: Carga el Top 10 historico desde un archivo .txt.
- Validacion: Compara el puntaje obtenido con la tabla existente.
- Escritura: Si el jugador clasifica, se solicita su nombre y se actualiza el registro de forma persistente.