# Documento Técnico: Experiencia de Realidad Virtual Multijugador

## 1. Introducción

Este documento técnico detalla las decisiones arquitectónicas, técnicas y optimizaciones implementadas durante el desarrollo de la experiencia de realidad virtual multijugador. El proyecto se ha desarrollado utilizando Unity 6 con Universal Render Pipeline (URP), XR Interaction Toolkit y Unity Netcode para proporcionar una experiencia inmersiva y colaborativa en entornos de realidad virtual.

## 2. Arquitectura del Sistema

### 2.1. Estructura General

La arquitectura del sistema se ha diseñado siguiendo un enfoque modular que separa claramente las diferentes funcionalidades:

* **Capa de Presentación**: Interfaz de usuario en VR y elementos visuales interactivos.
* **Capa de Lógica de Juego**: Comportamientos de los objetos, física e interacciones.
* **Capa de Networking**: Sincronización de estados y comunicación entre instancias.
* **Capa de Optimización**: Sistemas para garantizar el rendimiento en dispositivos XR.

## 3. Decisiones Técnicas

### 3.1. Optimizaciones URP

**Configuraciones clave de URP implementadas:** - Uso del perfil específico Mobile\_RPAsset para dispositivos XR

### 3.2. Solución de Networking

**Unity Netcode vs Mirror Networking**

Tras evaluar ambas opciones, se optó por **Unity Netcode** por: - Integración nativa con el ecosistema Unity - Implementación más directa para este tipo de proyecto - Documentación actualizada y en constante evolución - Soporte para NetworkVariables que facilitó la sincronización de estados de objetos

La implementación permite conexiones host-cliente en entornos de red local (LAN) mediante direcciones IP, sin necesidad de servidores externos, simplificando las pruebas y la implementación final.

### 3.3. Sistema XR

Se implementó el XR Interaction Toolkit con las siguientes características: - Configuración de XR Origin con Dynamic Move Provider para desplazamiento fluido - Integración del XR Device Simulator para pruebas en editor

### 3.4. Física e Interacciones

Se diseñaron tres tipos principales de objetos interactivos:

1. **Objeto 1 (Objeto con física completa)**:
   * Rigidbody con gravedad activada
   * Implementación de Physics.Simulate para mayor precisión en simulaciones físicas
   * Interacción directa con los controladores XR
2. **Objeto 2 (Plataforma cinemática)**:
   * Movimientos controlados mediante triggers
   * Movimientos animados mediante código
3. **Objeto 3 (Puerta giratoria con restricciones)**:
   * Uso de Hinge Joint para restricción de movimiento

### 3.5. Sincronización Multijugador

La sincronización entre jugadores se implementó mediante: - NetworkTransform para posiciones, rotaciones y gestos de avatares - NetworkVariable para exclusividad de objetos interactivos

Para la manipulación de objetos, se implementó: - Desactivación automática de colliders al ser sostenidos - Sincronización de propiedades físicas con NetworkRigidbody - Sistema de autoridad para resolver conflictos de interacción

## 4. Optimizaciones

### 4.1. Gráficos y Renderizado

**GPU Instancing** - Habilitado en todos los materiales para reducir draw calls

**Iluminación** - Baked lighting para elementos estáticos - Uso estratégico de Light Probes para iluminación dinámica eficiente

### 4.2. Gestión de Memoria

**Addressables** - Planificación de implementación para carga asíncrona de assets - (Nota: implementación pendiente por restricciones de tiempo)

**Optimización de texturas** - Compresión adaptada a plataformas móviles

**Optimización de modelos 3D** - Uso mínimo de poligonos para equilibrar apariencia y rendimiento

## 5. Decisiones de Implementación Específicas

### 5.1. Avatares y Representación del Jugador

* Utilización del modelo 3D “Loomis Head Foundation” para representar jugadores remotos
* Ocultación de la cabeza para el jugador local (visibilidad solo para otros jugadores)

### 5.2. Interfaz de Usuario

* UI adaptada para entornos VR con interacción directa
* Teclado virtual extraído del VR Sample para entrada de texto (configuración de IP)
* Sistema de validación para direcciones IP con formato 192.168.x.x
* Feedback visual para acciones críticas (conexión, errores, etc.)

### 5.3. Objeto Interactivo con Exclusividad

* Implementación de NetworkVariable para rastrear el estado de “sostenido”
* Desactivación automática de collider cuando un jugador sostiene el objeto
* Sistema de autoridad para resolver conflictos de manipulación
* Sincronización eficiente de propiedades físicas

## 6. Limitaciones y Mejoras Futuras

### 6.1. Limitaciones Actuales

* La desconexión de la red no está completamente implementada
* Algunos objetos no están sincronizados para todos los jugadores
* Carencia de mecánica creativa específica (limitada por tiempo de desarrollo)
* Pruebas unitarias no implementadas (limitadas por tiempo de desarrollo)

### 6.2. Oportunidades de Mejora

* Implementación completa de Addressables para carga dinámica de assets
* Desarrollo de mecánica creativa específica para entornos colaborativos VR
* Optimización adicional para diferentes dispositivos XR
* Implementación de pruebas unitarias y automatizadas
* Ampliación de interacciones físicas sincronizadas

## 7. Conclusiones

El desarrollo de esta experiencia de realidad virtual multijugador ha demostrado la viabilidad de crear entornos colaborativos inmersivos utilizando Unity 6, URP y Unity Netcode. Las decisiones técnicas implementadas han permitido alcanzar un equilibrio entre rendimiento, fidelidad visual e interactividad.

Las optimizaciones aplicadas han sido fundamentales para mantener el rendimiento en dispositivos XR, especialmente en lo referente a la renderización y sincronización en red. La arquitectura modular facilita la extensión futura del proyecto con nuevas funcionalidades.

A pesar de las limitaciones de tiempo que impidieron la implementación completa de algunas funcionalidades, la base técnica establecida es sólida y proporciona una plataforma adecuada para continuar el desarrollo.