

Inmersión a la primera computadora que llegó a México adaptando HTC Vive a Meta Quest

Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Cruz Lovera, T.M. (2024). Inmersión a la primera computadora que llegó a México adaptando HTC Vive a Meta Quest. Cuadernos Técnicos Universitarios de la DGTIC, 2 (2) páginas (8 - 23).

<https://doi.org/10.22201/dgtic.ctud.2024.2.2.44>

Tayde Martín Cruz Lovera

Dirección General de Cómputo y de
Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México

taydevr@comunidad.unam.mx

ORCID: 0009-0003-9519-8805

Resumen

Se describe cómo se realizó la adaptación de la aplicación de realidad virtual Inmersión 650. Esta aplicación, inicialmente diseñada para PC y ejecutada en el dispositivo HTC Vive, fue adaptada para el dispositivo independiente Meta Quest 2, el cual funciona con un sistema operativo basado en Android. Durante este proceso se rediseñaron algunas interacciones con el objetivo de mejorar la experiencia del usuario.

Palabras clave:

Realidad Virtual, interfaces de usuario diegéticas, inmersión, IBM 650.

1. INTRODUCCIÓN

El observatorio de visualización Ixtli, pionero en realidad virtual en México desde su inauguración en 2004, ha sido testigo de la evolución de la tecnología en los últimos 20 años. En la última década, la realidad virtual ha experimentado avances significativos, lo que ha permitido el desarrollo de dispositivos más accesibles y portables.

Para mantenerse relevante, el Observatorio Ixtli se ha adaptado progresivamente a estas nuevas tecnologías, al actualizar su infraestructura y desarrollar aplicaciones y proyectos para brindar a la comunidad universitaria acceso a esta tecnología.

La aplicación Inmersión 650 se creó en 2018 para conmemorar los 60 años de la llegada de la primera computadora a México, la IBM 650. Esta aplicación se desarrolló para dispositivos HTC Vive, la cual permite aprender a perforar una tarjeta usando el código de Hollerith, introducirla en la unidad de lectura y posteriormente leer los datos de la tarjeta en la consola.

El objetivo de este reporte es mostrar el trabajo de adaptación y mejora de la aplicación Inmersión 650 que inicialmente se desarrolló para dispositivos de realidad virtual para computadora hacia dispositivos independientes, y así mejorar la portabilidad y usabilidad.

Al finalizar este reporte se precisan, en un glosario, algunos conceptos técnicos utilizados.

2. DESARROLLO TÉCNICO

Originalmente la aplicación se desarrolló con la versión 5 de Unity, un motor de juegos. En el momento de la adaptación, la versión estable más reciente era la 2022, es decir, una diferencia de seis versiones, en donde, además, la parte de XR (realidades extendidas) ha experimentado una gran variedad de cambios y mejoras.

Inicialmente se utilizó SteamVR como framework de realidad virtual debido a su buena documentación, ejemplos sólidos, y su compatibilidad con varios dispositivos de realidad virtual para PC. Sin embargo, solo es compatible con PC, lo que obligó a buscar otro framework para adaptarse a Meta Quest 2, un dispositivo independiente con un sistema operativo basado en Android.

2.1 ANÁLISIS

Aunque ambos equipos comparten algunas características, también presentan diferencias significativas, especialmente en términos de sistema operativo y potencia gráfica y de cómputo, como se puede observar en la Tabla 1.

Tabla 1

Comparación entre los dispositivos de realidad virtual HTC Vive y Meta Quest 2

Características	HTC Vive	Meta Quest 2
Año de lanzamiento	2016	2019
Tipo de dispositivo	PC VR, es decir, requiere una PC de gama alta y una conexión compuesta de varios cables con la PC	Todo en uno, es un sistema independiente, no necesita conexiones adicionales
Tipo de seguimiento	Lighthouse tracking, utiliza dos sensores externos, montados en las paredes o trípodes, conectados entre ellos	Inside-Out, utiliza sensores y cuatro cámaras integradas en el propio dispositivo
Requiere PC	Sí, debe cumplir los requerimientos mínimos para funcionar	No, es independiente. Pero puede conectarse y funcionar como visor de PC VR de forma opcional
Baterías	Sólo en los mandos y son recargables	En el visor es recargable y una pila AA en cada mando
Resolución por ojo	1080x1200	1440x1600
Tasa de refresco de la pantalla	90 Hz	72 y 90 Hz
Sistema operativo	Windows (PC)	Basado en Android
Potencia gráfica y de cómputo	Mayor, dependiendo de la GPU y CPU de la PC	Limitada, su chip es un Qualcomm Snapdragon XR2
Posibilidades de seguimiento de manos	No	Sí, en las condiciones de iluminación adecuadas
Enfoque o vista del exterior (AR)	No	Sí, en blanco y negro y con baja resolución
Portabilidad	Muy poca, necesita una PC de alta gama, dos sensores externos adicionales al visor con tripié y mandos	Mucha, sólo se necesita el visor y los mandos
Instalación y calibración	Complicada, se tiene que conectar la PC, varios cables e instalar los sensores externos, además de realizar una calibración del nivel del piso y el espacio seguro	Sólo se tiene que configurar el nivel del piso y el espacio seguro
Resumen de las ventajas	Alta precisión en el seguimiento de movimiento y mayor potencia gráfica	Altamente portátil y de instalación fácil
Principales desventajas	No es portátil y requiere una instalación y calibración más complicadas	Limitado por la duración de la batería, aproximadamente dos horas, y menor potencia gráfica

Se llevó a cabo un análisis y comparación de las opciones de frameworks para Unity disponibles en el momento de la adaptación, lo cual se puede apreciar en la Tabla 2.

Tabla 2

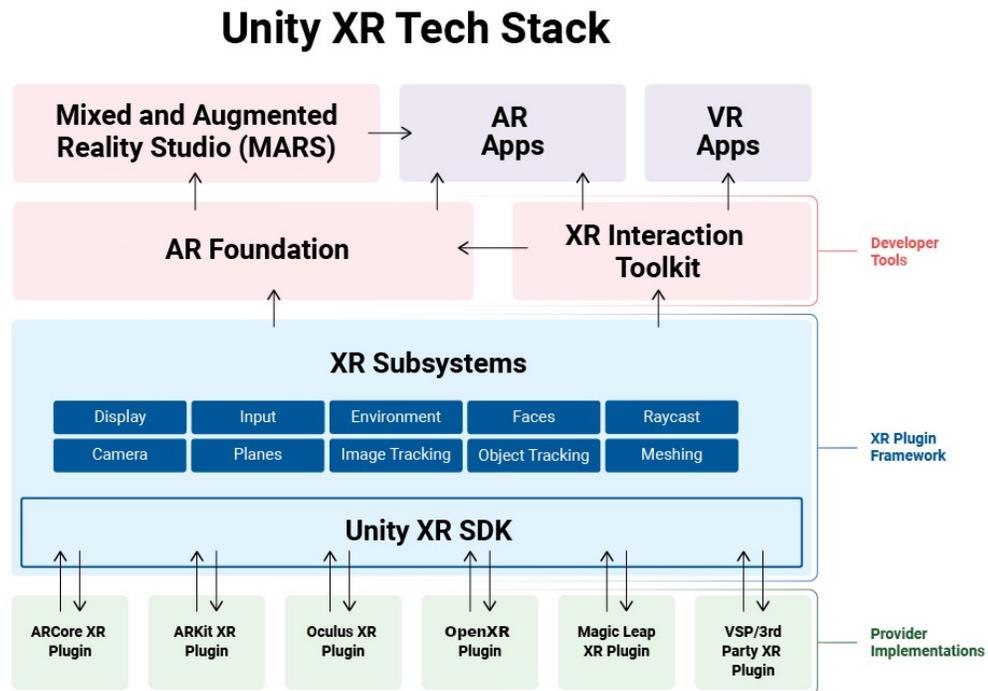
Comparación entre los principales frameworks de realidad virtual para Unity

Características	SteamVR	Oculus Integration SDK	XR Interaction Toolkit	Virtual Reality Toolkit (VRTK)	Windows Mixed Reality Toolkit (MRTK)
Desarrollador	Valve Corporation	Meta Platforms Technologies	Unity Technologies	VRTK Contributors	Mixed Reality Toolkit
Compatibilidad con dispositivos	Amplia pero solo para PC VR	Específica a dispositivos Oculus / Meta	Amplia, PC VR y dispositivos todo en uno	Amplia, PC VR y dispositivos todo en uno	Amplia, PC VR y dispositivos todo en uno
Capacidad de VR / AR	VR	VR y AR	VR y AR	VR y AR	VR y AR
Open Source	No	No	Sí	Sí	Sí
Costo	Gratis	Gratis	Gratis	Gratis	Gratis
Soporte para OpenXR	Sí	No	Sí	No	Sí
Ejemplos	Variados	Limitados	Variados	Variados	Variados
Facilidad de uso	Moderada	Moderada	Alta	Alta	Alta
Actualización y mantenimiento	Regular	Regular	Regular	Variable	Regular
Funcionalidades avanzadas		Seguimiento de manos, ojos y vista a través de las cámaras	Seguimiento de manos, ojos y vista a través de las cámaras	Seguimiento de manos y ojos	Seguimiento de manos, ojos y vista a través de las cámaras
Ventajas	Amplia compatibilidad y soporte para PC VR	Integración completa con dispositivos Oculus / Meta	Compatible con múltiples dispositivos y plataformas	Amplia gama de opciones de interacción	Apoyo y contribuciones continuas
Desventajas	Solo funciona con dispositivos de PC VR y está limitado a algunas áreas	Limitado a dispositivos Oculus / Meta y dificultad de uso regular	Puede requerir más configuración inicial	Depende de las contribuciones del grupo de desarrollo de código abierto	Microsoft lo desarrolló y le daba mantenimiento, pero desde 2023 dejó de darle soporte

Se seleccionó el framework XR Interaction Toolkit, desarrollado por Unity, debido a su facilidad de uso, su frecuente actualización y su soporte multiplataforma, que garantiza la compatibilidad con una variedad de dispositivos, independientemente de la marca. Al ser desarrollado por el propio motor de juegos, está integrado dentro de su arquitectura, como se puede apreciar en la Figura 1.

Figura 1

Diagrama general de la arquitectura de XR de Unity



Nota: Adaptado de Unity XR Tech Stack, por Unity, 2024, Unity Documentation (<https://docs.unity3d.com/Manual/XRPluginArchitecture.html>).

2.2 ADAPTACIÓN DE CONTENIDO Y FUNCIONALIDADES

La transición entre versiones significativamente diferentes del motor de juego y el notable número de cambios necesarios relacionados al funcionamiento de ambos frameworks, obligaron a reconstruir la aplicación desde cero en Unity. Esto implicó integrar el framework y luego configurar los parámetros de exportación de la aplicación adecuados para su ejecución en Meta Quest 2.

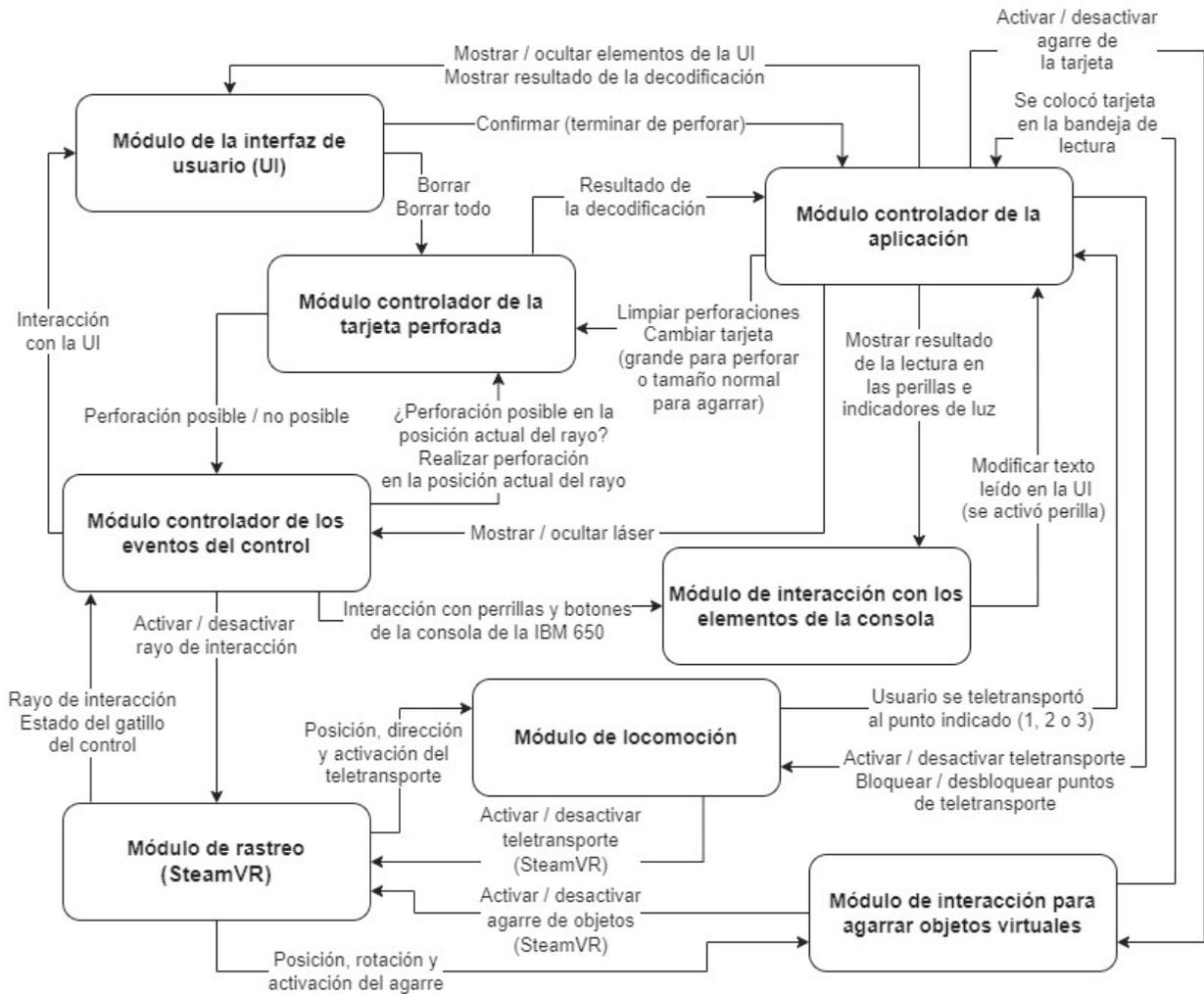
El siguiente paso consistió en integrar y organizar los elementos 3D que conforman el entorno 3D, incluyendo la arquitectura del lugar, el cielo y la IBM 650.

Posteriormente se realizaron los cambios necesarios en los *scripts* originales. En Unity, un script es un componente que se adjunta a un objeto para definir su comportamiento y funcionalidad dentro de la aplicación. Como se observa en la Figura 2, la aplicación Inmersión 650 se desarrolló originalmente

identificando módulos específicos relacionados a SteamVR, los cuales son: módulo de rastreo, módulo de locomoción y módulo de interacción.

Figura 2

Diagrama de módulos del desarrollo original



Nota: Adaptado de Diagrama de comunicación entre módulos (p. 49), por Cruz Lovera, T., 2022, Desarrollo de la aplicación de realidad virtual "Inmersión 650" para la exposición "Del bulbo a la nube" [Tesis de licenciatura, Universidad Nacional Autónoma de México].

Los demás módulos realizan sus funciones de forma independiente del framework de realidad virtual. El cambio en el código se limitó a sustituir, y en algunos casos adaptar, referencias a objetos o llamadas a métodos.

2.3 MEJORA DE LAS INTERACCIONES

Con el objetivo de mejorar las interacciones en términos de naturalidad y facilidad de uso, se llevó a cabo un rediseño de las interacciones. En la Tabla 3 se resumen las distintas interacciones usadas en el desarrollo original.

Tabla 3

Tabla de interacciones del desarrollo original

Interacción	Función	Disponibilidad	Mano
Teletransportación (trackpad + orientación del control)	Movimiento en el mundo virtual	En todo momento	Ambas
Interacción por medio de rayos (orientación del control + gatillo)	Realizar perforaciones en la tarjeta, activar botones de la UI y botones de la consola de la IBM 650	Únicamente cuando es necesaria	Derecha
Interacción directa (colisión del control con el objeto + gatillo)	Agarrar la tarjeta para colocarla en la bandeja de entrada de la unidad lectora y perforadora	Únicamente cuando es necesaria	Ambas

Nota: Adaptado de Diagrama de comunicación entre módulos (p. 45), por Cruz Lovera, T., 2022, Desarrollo de la aplicación de realidad virtual “Inmersión 650” para la exposición “Del bulbo a la nube” [Tesis de licenciatura, Universidad Nacional Autónoma de México].

Como se aprecia en la Tabla 3, en el desarrollo original se emplearon tres tipos de interacciones, una para locomoción y dos para interacción con objetos. Para la interacción con objetos se utilizó:

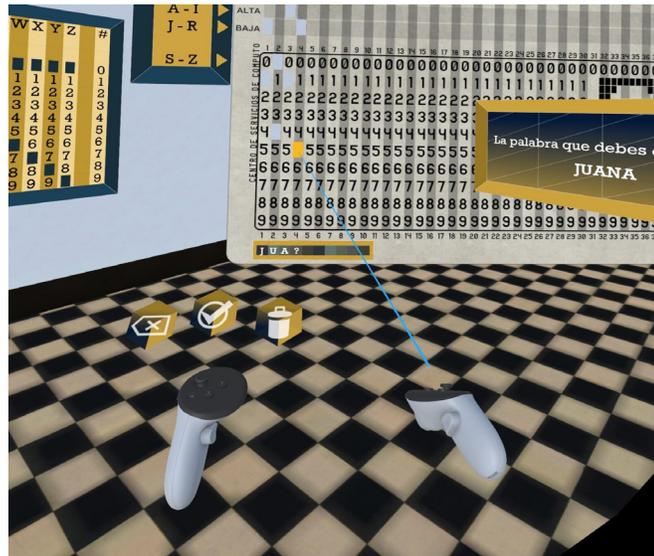
Interacción por medio de rayos – Consistía en apuntar al objeto con el cual se desea interactuar mediante rayos, que por lo general son visibles. Luego con un botón del control, se realizaba una acción sobre ese objeto similar al funcionamiento del ratón en una computadora.

Interacción directa – Implicaba acercar las manos virtuales o los controles al objeto con el cual se desea interactuar, de manera que haya una colisión física. Esto permitía agarrar o empujar el objeto utilizando el “motor de físicas” del motor de juegos.

La interacción por medio de rayos se utilizó para la perforación de la tarjeta, como se puede observar en la Figura 3. Esto se debió a la necesidad de brindar al usuario precisión para indicar el lugar exacto donde realizar la perforación.

Figura 3

Captura de pantalla de la interacción original de perforación de la tarjeta

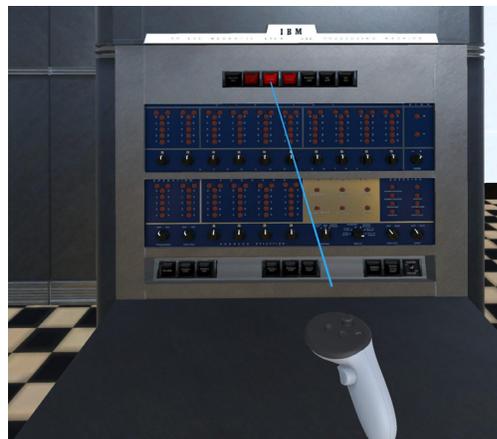


Nota: En la captura de pantalla se pueden observar los botones para borrar la última perforación, borrar todas las perforaciones y confirmar, así como el rayo de interacción para perforar la tarjeta.

También se utilizó la interacción por medio de rayos para interactuar con los botones de la consola de la IBM 650, como se puede ver en la Figura 4. Esto se debió al tamaño de los numerosos botones de la consola, los cuales eran muy pequeños en comparación con el tamaño de las manos virtuales, representadas por los controles del HTC Vive.

Figura 4

Captura de pantalla de la interacción original con los botones de la consola de la IBM 650



Y, por último, también se utilizó la interacción por medio de rayos para interactuar con los botones de la interfaz de usuario que se puede ver en el control izquierdo en la Figura 3. Estos botones, activados únicamente en la parte de perforación de la tarjeta, permitían borrar perforaciones y concluir el proceso.

Con el objetivo de mejorar las interacciones se llevaron a cabo los ajustes mostrados en la Tabla 4.

Tabla 4

Tabla de interacciones revisada

Interacción	Función	Disponibilidad	Mano
<p>Teletransportación (palanca hacia adelante + orientación del control)</p> <p>Rotación por intervalos (palanca hacia la derecha o izquierda)</p>	Movimiento en el mundo virtual	En todo momento	Ambas
<p>Interacción directa (colisión del control con el objeto + gatillo lateral)</p>	Agarrar objetos (con la posibilidad de activarlos con el gatillo trasero) y empujar	En todo momento	Ambas

El modo de desplazamiento se mantuvo como en el diseño original, utilizando el teletransporte. Aunque esta técnica puede parecer menos natural y afectar la sensación de presencia o interrumpirla (Riecke y Zielasko, 2021), es una técnica comprobada para reducir la sensación de mareos en realidad virtual (Al Zayer et al., 2020).

Se eliminó la interacción por medio de rayos para dejar únicamente la interacción directa. La interacción de perforación de la tarjeta y con la interfaz de usuario se modificaron al utilizar una interfaz de usuario diagética o física, la cual integra elementos visuales de manera coherente como parte del entorno virtual.

Los botones de la interfaz de usuario, que originalmente se integraron como botones planos flotantes, se sustituyeron por un panel de control con botones físicos, como se puede apreciar en la Figura 5. Estos botones se activan cuando el usuario los empuja con su mano. Además, anteriormente, el nombre solicitado al usuario para perforar se mostraba en un panel flotando, en su lugar, se muestra en la pantalla que se agregó como parte del panel de control.

Figura 5

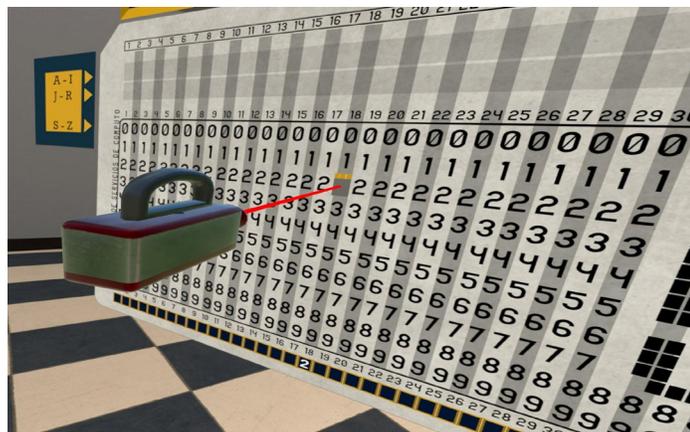
Captura de pantalla de la interacción con botones de la interfaz de usuario diegética o física



Para la perforación de la tarjeta, se integró un elemento interactivo que el usuario puede agarrar, el cual se puede observar en la Figura 6. El rayo aparece automáticamente cuando el usuario agarra el objeto y se oculta cuando el usuario lo suelta. Para perforar la tarjeta, el usuario tiene que apuntar con el rayo que emite este objeto y confirmar al oprimir el gatillo trasero.

Figura 6

Captura de pantalla de la nueva interacción de perforación de la tarjeta

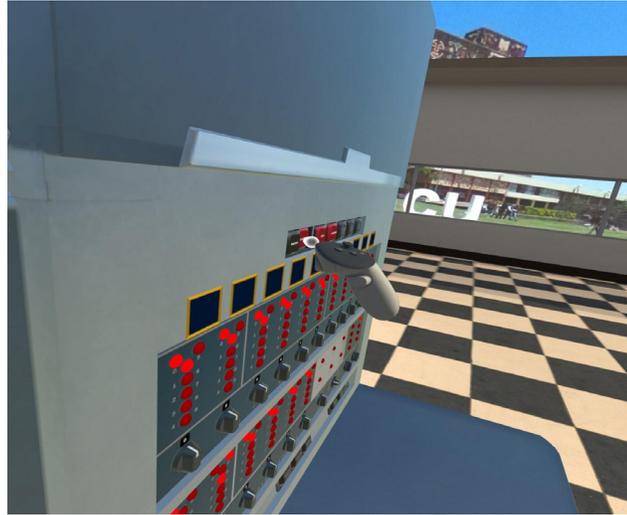


Esto no sólo permite darle una justificación a la existencia del rayo de perforación, sino que también permite controlar su aparición de forma que resulte más natural.

Por último, para la interacción con los botones de la consola de la IBM 650, se agregó una esfera pequeña a cierta distancia de la mano virtual, representada por el control del dispositivo, como se puede ver en la Figura 7, para extender la zona de interacción.

Figura 7

Captura de pantalla de la interacción de la esfera pequeña con botones de la consola



De esta manera, el usuario tiene la posibilidad de interactuar de forma precisa con objetos pequeños, como los botones de la consola de la IBM 650.

2.4 OPTIMIZACIÓN

La optimización es uno de los puntos más importantes al momento de hacer una adaptación de este tipo, debido a la diferencia de potencia gráfica y de cómputo entre los dispositivos.

Se realizó la configuración de la iluminación y se aplicaron las optimizaciones integradas en el motor de juegos. Adicionalmente, se identificaron los puntos principales que afectaban el rendimiento de la aplicación por medio del perfilador de Unity y la herramienta de métricas de Meta. Se sustituyeron los materiales por unos diseñados para dispositivos móviles, con menor calidad de iluminación, pero requiriendo menos procesamiento.

Además, el *shader* utilizado para realizar el efecto de perforaciones en la tarjeta se adaptó para su correcto funcionamiento. En la Figura 8 y Figura 9 se pueden ver los códigos en lenguaje de shaders de Unity, que es una variante de Cg/HLSL, que crean el efecto de perforaciones al usarse en conjunto, ya optimizados para realidad virtual.

Figura 8

Shader utilizado para la tarjeta

```
Shader "Custom/UnlitStencilMasked"  
{  
    Properties  
    {  
        _MainTex ("Texture", 2D) = "white" {}  
    }  
}
```

```
}  
SubShader  
{  
    Tags  
    {  
        "Queue"="Transparent"  
        "IgnoreProjector"="True"  
        "RenderType"="Transparent"  
        "PreviewType"="Plane"  
        "CanUseSpriteAtlas"="True"  
    }  
  
    Lighting Off  
    Blend One OneMinusSrcAlpha  
  
    Stencil  
    {  
        Ref 1  
        Comp notequal  
    }  
  
    Pass  
    {  
        CGPROGRAM  
        #pragma vertex vert  
        #pragma fragment frag  
  
        #include "UnityCG.cginc"  
  
        struct appdata  
        {  
            float4 vertex : POSITION;  
            float2 uv : TEXCOORD0;  
            UNITY_VERTEX_INPUT_INSTANCE_ID  
        };  
  
        struct v2f  
        {  
            float2 uv : TEXCOORD0;  
            float4 vertex : SV_POSITION;  
            UNITY_VERTEX_INPUT_INSTANCE_ID  
            UNITY_VERTEX_OUTPUT_STEREO  
        };  
  
        sampler2D _MainTex;  
        float4 _MainTex_ST;
```



```
Blend One OneMinusSrcAlpha

Stencil {
    Ref[_StencilMask]
    Comp always
    Pass replace
}

Pass {
    CGPROGRAM
    #pragma vertex vert
    #pragma fragment frag

    #include "UnityCG.cginc"

    struct appdata {
        float4 vertex : POSITION;
        UNITY_VERTEX_INPUT_INSTANCE_ID
    };

    struct v2f {
        float4 pos : SV_POSITION;
        UNITY_VERTEX_INPUT_INSTANCE_ID
        UNITY_VERTEX_OUTPUT_STEREO
    };

    v2f vert(appdata v) {
        v2f o;
        UNITY_SETUP_INSTANCE_ID(v);
        UNITY_INITIALIZE_OUTPUT(v2f, o);
        UNITY_INITIALIZE_VERTEX_OUTPUT_STEREO(o);
        o.pos = UnityObjectToClipPos(v.vertex);
        return o;
    }

    half4 frag(v2f i) : COLOR {
        UNITY_SETUP_STEREO_EYE_INDEX_POST_VERTEX(i);
        return half4(1, 1, 0, 1);
    }

    ENDCG
}
}
```

3. RESULTADOS

La adaptación de la aplicación facilitó su uso y portabilidad en las demostraciones de realidad virtual tanto en el Observatorio Ixtli como en eventos externos. Además, se propusieron nuevas formas de interacción, más parecidas a como se hacen en la realidad, con el objetivo de mejorar la experiencia de usuario, algo que aún está pendiente de evaluar; se necesita realizar pruebas de usuario y recopilación de comentarios para garantizar que satisfaga las necesidades y expectativas de los usuarios.

Algunas lecciones importantes aprendidas durante este proceso incluyen:

- La elección del framework facilitó una adaptación rápida y sin complicaciones. Además, se espera que la aplicación pueda instalarse y utilizarse en dispositivos independientes de otros fabricantes con pocos ajustes adicionales.
- La optimización es crucial debido a la gran diferencia en la capacidad gráfica y de cómputo entre ambos dispositivos.
- El uso de interfaces diegéticas en realidad virtual aún necesita más investigación, pero podría ofrecer interacciones más naturales y satisfactorias, aunque no hay indicios definitivos sobre si aumenta la sensación de inmersión, los usuarios parecen preferirlas sobre las interfaces no diegéticas (Saling et al., 2021).
- Extender la zona de interacción de las manos virtuales con un punto de interacción a unos centímetros del control puede ser una solución para implementar interacciones de agarre o manipulación de objetos muy pequeños.

4. CONCLUSIONES

La adaptación de aplicaciones de HTC Vive a Meta Quest 2 representa un proceso desafiante principalmente debido a la disminución de potencia gráfica y de cómputo disponible al pasar de un dispositivo a otro, lo que lleva a la necesidad de optimizar el rendimiento gráfico y la funcionalidad para garantizar una experiencia fluida y satisfactoria, así como realizar compromisos en términos de calidad visual.

Entre los beneficios más destacados se encuentra la mejora en la portabilidad, esto facilita la movilidad y la accesibilidad, lo que puede aumentar la audiencia y el alcance de la aplicación. Además, al eliminar la necesidad de cables y sensores externos, proporciona una experiencia más cómoda y libre de obstáculos.

También abre la puerta a posibles innovaciones, al actualizar o desarrollar nuevas aplicaciones para el Observatorio Ixtli, lo cual presenta desafíos para pensar de formas distintas en comparación con las aplicaciones no inmersivas; el diseño de aplicaciones de realidades extendidas es un área relativamente nueva y en constante evolución. Esto puede llevar a la creación de experiencias únicas y creativas que aprovechen al máximo las capacidades de esta tecnología.

REFERENCIAS

- Al Zayer, M., MacNeilage, P., & Folmer, E. (2020). *Virtual locomotion: A survey*. IEEE Transactions on Visualization and Computer Graphics, 26(6), 2315–2334. <https://doi.org/10.1109/TVCG.2018.2887379>
- Cruz Lovera, T. (2022). *Desarrollo de la aplicación de realidad virtual "Inmersión 650" para la exposición "Del bulbo a la nube"* [Tesis de licenciatura, Universidad Nacional Autónoma de México].
- IBM. (s.f.). *Reference Manual IBM 29 Card Punch*. Bitsavers.org. Recuperado el 15 de marzo de 2024, de https://bitsavers.org/pdf/ibm/punchedCard/Keypunch/029/A24-3332-3_29_Reference_Man.pdf
- Riecke, B. E., & Zielasko, D. (2021). *Continuous vs. Discontinuous (Teleport) Locomotion in VR: How Implications can Provide both Benefits and Disadvantages*. 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW).
- Saling, F., Bernhardt, D., Lysek, A., & Smekal, M. (2021). *Diegetic vs. Non-diegetic GUIs: What do virtual reality players prefer?* The Eurographics Association. <https://doi.org/10.2312/EGVE.20211333>
- Unity. (2024). *Unity XR Tech Stack*. Unity Documentation. <https://docs.unity3d.com/Manual/XRPluginArchitecture.html>

GLOSARIO

Motor de juegos. Conjunto de herramientas y funcionalidades diseñadas para facilitar el desarrollo de videojuegos y otras aplicaciones interactivas.

Realidades extendidas o XR. Término utilizado para referirse a un conjunto de tecnologías que incluyen realidad virtual (VR), realidad aumentada (AR) y realidad mixta (MR).

Teletransporte. Técnica de locomoción que permite a los usuarios moverse dentro de un entorno virtual, consiste en seleccionar un punto de destino en el mundo virtual y luego ser instantáneamente transportado a ese lugar.

Material. En gráficos por computadora se refiere a la información utilizada para definir cómo se ve un objeto en una escena virtual, esto incluye color, transparencia, textura, brillo, etc.

Shader. Programa que se ejecuta en la GPU, utilizado en gráficos por computadora para definir la apariencia visual de los objetos, determinando cómo se calcula el color y otros atributos de cada píxel en una pantalla, así como la posición de los vértices de un objeto 3D.