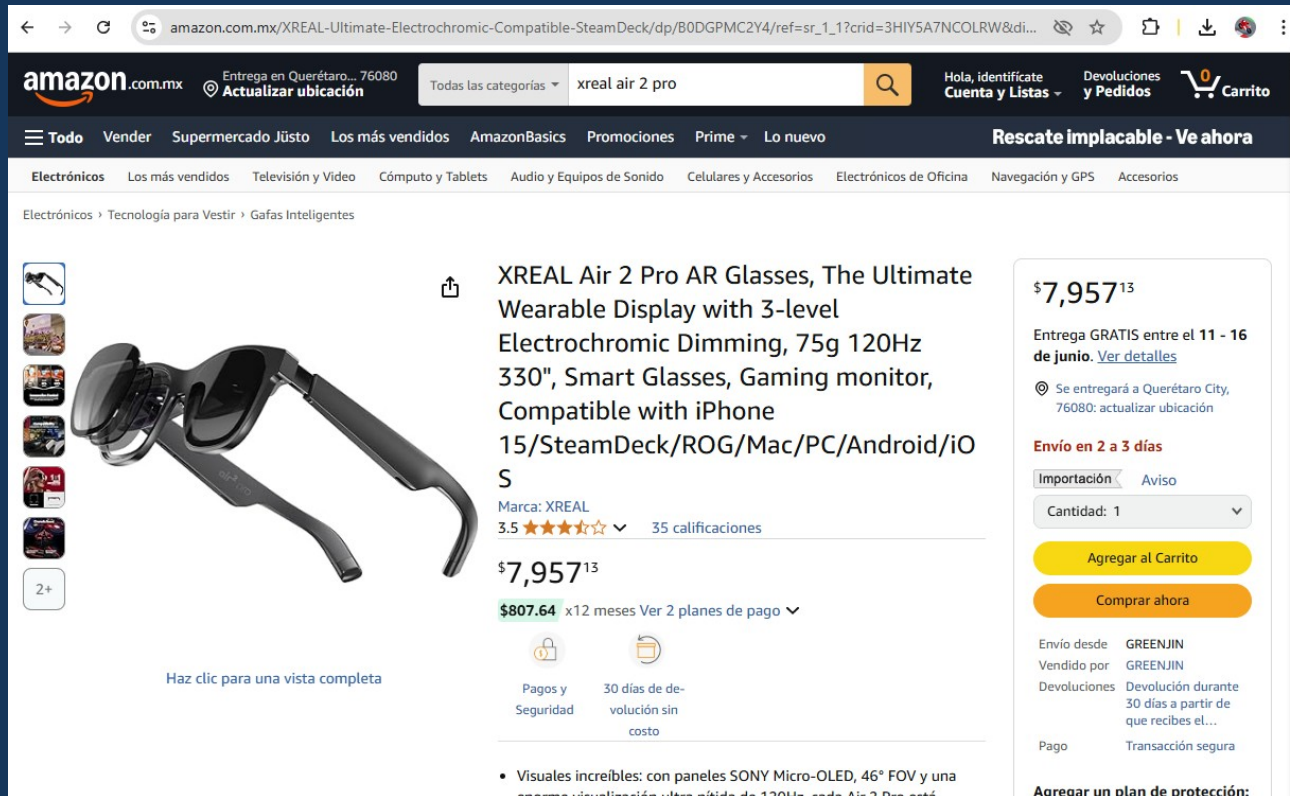


Realidad Extendida

José Luis Carreño Arteaga
jcarreno53@yahoo.com.mx

Las gafas de realidad extendida de nuestra elección

- Referencia:
- XREAL Air 2 Pro AR Glasses



The screenshot shows the Amazon Mexico product page for the XREAL Air 2 Pro AR Glasses. The page features a large image of the glasses on the left, with a vertical strip of smaller images showing different views and use cases. The main title is "XREAL Air 2 Pro AR Glasses, The Ultimate Wearable Display with 3-level Electrochromic Dimming, 75g 120Hz 330", Smart Glasses, Gaming monitor, Compatible with iPhone 15/SteamDeck/ROG/Mac/PC/Android/iOS". The price is listed as \$7,957¹³, with a financing option of \$807.64 x12 meses. The page also includes a star rating of 3.5 from 35 reviews, a "Haz clic para una vista completa" button, and a "Rescate implacable - Ve ahora" banner. The right sidebar contains delivery information, shipping options, and purchase buttons like "Agregar al Carrito" and "Comprar ahora".

amazon.com.mx Entrega en Querétaro... 76080 Actualizar ubicación


Todas las categorías xreal air 2 pro


Hola, identifícate Cuenta y Listas Devoluciones y Pedidos Carrito

Todo Vender Supermercado Justo Los más vendidos AmazonBasics Promociones Prime Lo nuevo Rescate implacable - Ve ahora

Electrónicos Los más vendidos Televisión y Video Cómputo y Tablets Audio y Equipos de Sonido Celulares y Accesorios Electrónicos de Oficina Navegación y GPS Accesorios

Electrónicos > Tecnología para Vestir > Gafas Inteligentes





 XREAL Air 2 Pro AR Glasses, The Ultimate Wearable Display with 3-level Electrochromic Dimming, 75g 120Hz 330", Smart Glasses, Gaming monitor, Compatible with iPhone 15/SteamDeck/ROG/Mac/PC/Android/iOS

Marca: XREAL 3.5 ★★★★★ 35 calificaciones

\$7,957¹³

\$807.64 x12 meses Ver 2 planes de pago

 Pagos y Seguridad  30 días de devolución sin costo

Haz clic para una vista completa

Envío desde GREENJIN Vendido por GREENJIN Devoluciones Devolución durante 30 días a partir de que recibes el... Pago Transacción segura

Agregar un plan de protección:

Diseño interactivo para la Realidad extendida:

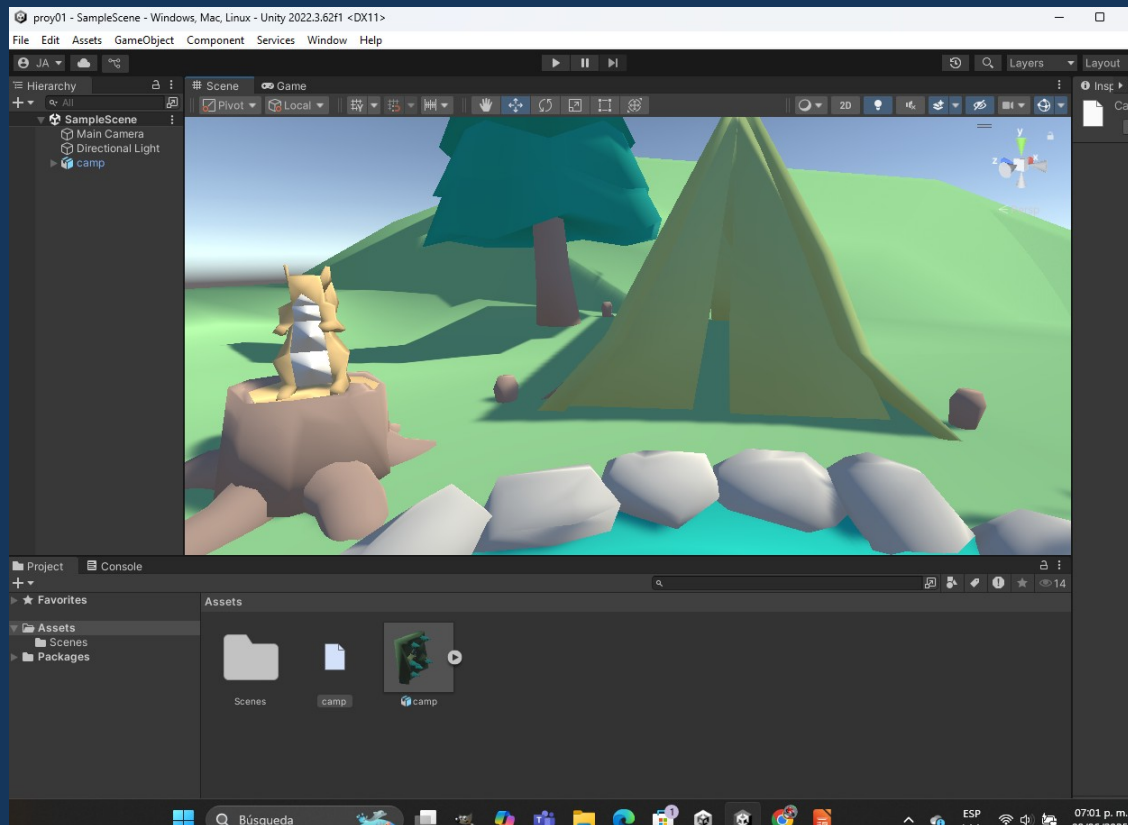
- Referencia:
- IA
- En resumen, el diseño interactivo es esencial para que las experiencias de RX y WebXR sean intuitivas, atractivas y útiles para los usuarios. Permite a los diseñadores crear entornos virtuales y realidades aumentada que sean tanto inmersivas como fáciles de usar.
-
-
-

Diseño interactivo para la Realidad extendida:

- Referencia:
- Sean Keogh 3 de septiembre de 2024
- El arte del diseño de interactivo en XR reside en liberarse de las limitaciones del mundo físico y abrazar lo imposible. Al aprovechar las capacidades únicas de XR, los diseñadores pueden crear experiencias inmersivas, intuitivas y atractivas que van más allá de lo que ofrecen las interfaces tradicionales. A medida que la tecnología XR evoluciona, el potencial de innovación en el diseño de interacción es ilimitado. La clave está en aprovechar este potencial con inteligencia, creando interfaces funcionales y atractivas: simples cuando es necesario, complejas cuando es necesario. Este enfoque permite a los diseñadores redefinir las interacciones digitales, abriendo nuevos mundos de posibilidades.

Analizando el Modelo

- Del proyecto:
- Ir al folder media/gltf/camp y de ahí copiar el archivo camp.blend
- Pasar este archivo por un convertidor en línea: Blend to Obj
- Presentarlo en Unity rotar el modelo -90 en X
-



El código QR

● Del proyecto:



Wright State University

 WEBXR

Elementos a importar

- 3DOF permite rotación pero no traslación. Con traslación sería un proyecto 6DOF.

- import {WebXRButton} from './js/util/webxr-button.js'; // importa para el botón de realidad aumentada
- import {Scene} from './js/render/scenes/scene.js'; // Importa para escena
- import {Renderer, createWebGLContext} from './js/render/core/renderer.js'; // Crea contexto WebGL
- import {Gltf2Node} from './js/render/nodes/gltf2.js'; // render del modelo
- import {SkyboxNode} from './js/render/nodes/skybox.js'; // Presentación del Domo
- import {BoundsRenderer} from './js/render/nodes/bounds-renderer.js'; // Limites de renderizado
- import {InlineViewerHelper} from './js/util/inline-viewer-helper.js';
- import {QueryArgs} from './js/util/query-args.js';
-
-
-

polyfill para soporte de dispositivos que soportan realidad virtual o realidad aumentada

```
// Si se solicita, utilizar polyfill para proveer el soporte de dispositivos móviles  
// y dispositivos que soportan unicamente WebVR.
```

```
import WebXRPolyfill from './js/third-party/webxr-polyfill/build/webxr-polyfill.module.js';  
if (QueryArgs.getBool('usePolyfill', true)) {  
  let polyfill = new WebXRPolyfill(); // polyfill y un ejemplo de código para construir aplicaciones de realidad  
  aumentada (AR)                       // y realidad virtual (VR) usando WebXR.  
}
```



Inicia soporte de realidad aumentada

```
function initXR() { // inicia soporte de realidad aumentada
    xrButton = new WebXRButton({ // Solicitando sesión de realidad virtual con el botón
        onRequestSession: onRequestSession,
        onEndSession: onEndSession
    });

    document.querySelector('header').appendChild(xrButton.domElement);
    // Si el navegador proporciona una experiencia XR
    if (navigator.xr) {
        // entonces el navegador proporciona la experiencia VR
        navigator.xr.isSessionSupported('immersive-vr').then((supported) => {
            xrButton.enabled = supported;
        });
        // Ir a la función onSessionStarted
        navigator.xr.requestSession('inline').then(onSessionStarted);
    }
}
```

local-floor y bounded-floor

```
function onRequestSession() {  
  return navigator.xr.requestSession('immersive-vr', {  
    // La experiencia del usuario requiere cuando menos de un local-floor  
    // del espacio de referencia (el cual estara disponible en un dispositivo 3DoF)  
    // pero opcionalmente se puede utilizar un espacio de referencia bounded-floor  
    // cuando se encuentra disponible.  
    requiredFeatures: ['local-floor'], // El usuario no se mueve en el espacio y local-floor de pie  
  
    optionalFeatures: ['bounded-floor'] // Una experiencia limitada bounded-floor ( ) es una en la que el  
                                          // usuario se moverá alrededor de su entorno  
                                          // físico para interactuar //completamente,  
                                          // pero no necesitará viajar  
                                          // más allá de un límite fijo definido por el hardware XR.  
  }).then((session) => {  
    xrButton.setSession(session);  
    session.isImmersive = true; // Sesión inmersiva  
    onSessionStarted(session);  
  });  
}
```

Cambiando el tamaño del canvas en función del dispositivo

```
function onResize() { // Cambia el tamaño del canvas en relación con el dispositivo
  gl.canvas.width = gl.canvas.clientWidth * window.devicePixelRatio;
  gl.canvas.height = gl.canvas.clientHeight * window.devicePixelRatio;
}
```

Se crea el espacio de referencia

```
function onRefSpaceCreated(refSpace) {
  if (session.isImmersive) {
    xrImmersiveRefSpace = refSpace;
  } else {
    // Si usamos un espacio de referencia del visor, necesitamos desplazar
    // el origen ligeramente hacia abajo para colocar la cámara aproximadamente
    // a la altura correcta. (Aquí la movemos 1,6 metros, que es *muy*
    // aproximadamente la altura de los ojos de un adulto promedio).
    inlineViewerHelper = new InlineViewerHelper(gl.canvas, refSpace);
    inlineViewerHelper.setHeight(1.6);

    // Puedes lograr lo mismo sin la clase auxiliar simplemente compensando
    // el espacio de referencia con un valor y negativo:
    // refSpace = refSpace.getOffsetReferenceSpace(new XRRigidTransform({y: -1.6}));
  }
  session.requestAnimationFrame(onXRFrame);
}
```

Si la sesión es inmersiva

```
if (session.isImmersive) {  
  // Intente obtener un espacio de referencia de "piso delimitado",  
  // que alineará el piso físico del usuario con Y=0 y proporcionará  
  // los límites que indiquen dónde puede caminar el usuario de manera segura.  
  session.requestReferenceSpace('bounded-floor').then((refSpace) => {  
    onRefSpaceCreated(refSpace);  
  
    if (!boundsRenderer) {  
      boundsRenderer = new BoundsRenderer(refSpace);  
      scene.addNode(boundsRenderer);  
    } else {  
      boundsRenderer.boundedRefSpace = refSpace;  
    }  
  
    refSpace.onreset = (evt) => {  
      boundsRenderer.boundedRefSpace = evt.referenceSpace;  
    }  
  }).catch(() => {  
    // Ahí es donde se coloca la textura en relación al modelo (el piso local)  
    // Si no se admite un espacio de referencia delimitado, se recurre a  
    // un espacio de referencia del piso local.  
    // Esto sigue proporcionando un espacio relativo al piso y  
    // siempre será compatible con las sesiones inmersivas.  
    // Sin embargo, no proporciona límites y, por lo general,  
    // se espera que el usuario permanezca en un punto fijo.  
    // Si el dispositivo no puede determinar el nivel del piso (por ejemplo,  
    // con un dispositivo 3DoF), devolverá un espacio de planta local emulado,  
    // donde la vista se traslada hacia arriba mediante una altura estática para  
    // que la escena se renderice aproximadamente en el punto correcto.  
  
    console.log('Falling back to local-floor reference space');  
    session.requestReferenceSpace('local-floor').then(onRefSpaceCreated);  
  });  
}
```

Referencias Bibliográficas