Mover

Sentar

Configuraciones

Alimentos Stock

Barra de Hambre

MiniJuego 2d y Nubecita

Marcador Mascota

CANVAS

Acciones

ESCENA MINIJUEGO 2D

INTERACTUAR DE FORMA TACTIL CON MASCOTA

USUARIO

MENU PRNICIPAL

REPRESENTACION DE MASCOTA AR

10 SEGUNDOS DE ESCANEO REPRESENTACION

**(ESCENA)Comienzo del juego:** Al abrirse el juego aparece el lobito caminando hacia el frente, con el nombre del juego apareciendo con alguna animación y detrás del lobito un fondo low poly con dinámica. La idea es que solo se presione la pantalla para comenzar con el escaneo AR y representación de la mascota AR, donde el lobo tire como un comentario press play o algo asi mientras esta sentado.

**(ESCENA)Escaneo de AR:**

-En esta escena es donde primero, se va escanear el lugar donde va ser representado el lobito, mostrar unos segundos en pantalla con un comentario para que el usuario escanee el sector.

- 1 Luego, aparecerá el terreno y debemos posicionarlo en el plano que mas nos parezca,(Quiza permitir agrandar o achicar terreno), al presionar la pantalla el terreno se representara donde se apreto, y saldrán arboles y plantas random desde él. Este será el lugar donde el lobito se moverá.

- 2 Luego, aparecerá un marcador sobre los planos que detecto la cámara, y al presionar la pantalla que se represente el lobito en el lugar donde se ubicaba ese marcador, si este marcador colisiona con un mesh collider no se debe poder poner el lobito. (Raycast), una vez que sea detectado en que plano quiero el lobito, ese plano se transforme en una “plaza” pasto y piedras random y arboles random. No pude hacerlo por no sacar el tamaño del plano.

- En esta escena se va poder relacionar con la mascota de forma tactil, ya sea hacerla mover, saltar, acariciar, que mire para los costados, que se siente se pare y se mueva por el mismo terreno, presionando en el y detectando con el raycast e nque parte del terreno y que el lobo camine hacia ese lugar. (Raycast)

* Canvas para las opciones sonido, graficos.
* Canvas para los minijuegos que nos llevaría a otras escenas.
* Canvas donde vamos a una tienda y comprar alimento para la mascota con las moneditas ganadas.
* Canvas con comida en stock, donde puede ser lanzada al lobo cayendo de arriba o lanzarla desde la cámara y colisione con el terreno
* Canvas donde va estar una barra de alimento de la mascota, donde no se debe gastar en caso que suceda, se debe comenzar nuevamente el juego y nuestra mascota muere xD, es obligatorio hacer una condición de perdida o ganado para la facultad.
* Un minijuego dentro de esta, es una nubecita que respawnea arriba de nuestra mascota y se mueve hacia los lados, donde lanza monedas o bombas, donde debemos agarrar, si colisionamos con dos bombas, se termina el juego y sino son 60 seg.
* Marcador para ver donde esta nuestra mascota para no perderla xD

**(ESCENA)Minijuego 2d:** Un juego 2d donde me parece buena idea mezclar AR con 3D con 2D, en un mismo proyecto para hacerlo mas cambiante. Donde el usuario debe saltar con la mascota por plataformas e ir ganando monedas para poder comprarle comida para alimentar a la mascota.

TEMA GUARDADO LOCAL: Buscar e investigar sobre un guardado eficiente, donde no sea base de datos, sino guardado local en el celular.

//SI SE LLEGA CON EL TIEMPO

(ESCENA) Mundo virtual: Generar un mundo virtual donde nosotros aparecemos y vemos en nuestro celu ese mundo, pero a medida que nos movemos en el real nos movemos en ese siguiendo a nuestra mascota o paseándola y explorando ese mundo. Mundo LOW POLY.

API AR:

ArFoundation/ARCORE: <https://docs.unity3d.com/Packages/com.unity.xr.arcore@4.2/manual/>

The ARCore XR Plug-in package enables ARCore support via Unity's multi-platform XR API. This package implements the following XR subsystems:

* [Session](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/session-subsystem.html)
* [Camera](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/camera-subsystem.html)
* [Depth](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/depth-subsystem.html)
* [Input](https://docs.unity3d.com/2020.3/Documentation/ScriptReference/XR.XRInputSubsystem.html)
* [Planes](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/plane-subsystem.html)
* [Raycast](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/raycasting-subsystem.html)
* [Anchors](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/anchor-subsystem.html)
* [Face tracking](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/face-tracking.html)
* [Image tracking](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/image-tracking.html)
* [Environment probes](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/environment-probe-subsystem.html)
* [Occlusion](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/occlusion-subsystem.html)

This version of ARCore XR Plug-in uses ARCore 1.24 and supports the following functionality:

* Device localization
* Horizontal plane detection
* Vertical plane detection
* Point clouds
* Pass-through camera view
* Light estimation
* Anchors
* Oriented feature points
* Hit testing
* Session management
* ARCore APK on-demand installation
* 2D Image tracking
* Face tracking
* Environment probes
* Occlusion

This package does not support the following subsystems:

* [Object tracking](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/object-tracking.html)
* [Participant](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/participant-subsystem.html)
* [Mesh](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/mesh-subsystem.html)
* [Body tracking](https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/api/UnityEngine.XR.ARSubsystems.XRHumanBodySubsystem.html)

Lightship: <https://lightship.dev/>

<https://lightship.dev/docs/ardk/index.html>

requisitos: https://lightship.dev/docs/ardk/ardk\_fundamentals/system\_reqs.html#doxid-system-reqs