

# Card Tracker

## COMPUTACIÓN UBICUA E INTELIGENCIA AMBIENTAL

### **Descripción.**

Card Tracker es una aplicación para android que nos permitirá jugar a diferentes juegos de mesa, utilizando solamente un mazo de cartas.

Para ello, la aplicación tiene una serie de cartas registradas y unos diseños 3d que aparecerán cuando la cámara enfoque dichas cartas.

Además la versión final de la aplicación tendrá la capacidad de crear tus propias cartas con tan solo un lápiz y un papel. Posteriormente podrás decidir qué diseños 3d asocias a dichas cartas, haciendo la aplicación perfecta para una versión gratuita inicial ya que no necesitas ningún elemento complementario.

Para conseguir la inmersión necesaria, se dará opción a utilizar unas cardboard y usar la realidad aumentada - realidad virtual. Esta idea fue concebida después de mostrar la aplicación a distintos clientes potenciales.

Además de la opción que utiliza la cardboard, habrá otra versión sin ella, con las mismas funciones.

### **¿Cuál es el problema a resolver?**

Card Tracker se presenta, no solo como una aplicación para jugar con tus amigos o familiares, si no como una solución a unos problemas típicos para los consumidores habituales de juegos de mesa; el espacio que ocupan, el problema de transportarlos, y el precio son algunos de estos.

Todos ellos son solucionados con Card Tracker.

### **Competencia.**

Card tracker no tiene ningún tipo de competencia directa. No hay aplicaciones que usen esta idea, por lo que le otorga un gran potencial.

Además, al tener una versión para uso de cardboard y otra sin ellas, podemos abarcar un público mucho mayor.

**Beneficios.**

La principal fuente de beneficios de la aplicación será por anuncios. Sin embargo, habrá juegos u opciones, como aumentar el número de modelos 3d activos en pantalla, que deben desbloquearse mediante un pago inicial.

**Software y Hardware utilizado.**

El software utilizado para el desarrollo de la aplicación ha sido unity 2018.1.0 y vuforia. He creado una base de datos de vuforia con más de 10 tipos de cartas que nos permite abarcar una gran cantidad de juegos distintos, desde juegos de tablero a juegos de cartas tradicionales.

Para la inclusión posterior del modo de fabricar tus propios juegos, se utilizará una base de datos más reducida pudiendo ser usada sin aumentar el coste de manera excesiva.

Para el testeo de la aplicación he utilizado un Motorola moto G5 Plus, y unas cardboard genéricas.

**Limitaciones por software y hardware.**

Para el uso de la aplicación será necesario un móvil con android. El sistema android no podrá ser inferior a la versión Marshmallow (API level 23).

Actualmente, se permite cargar 5 cartas a la vez.

Las cardboard compatibles son las GoogleCardboard aunque también se ha añadido la opción de poder usar una genérica.

**Principales objetivos a mejorar previo a la publicación de la aplicación.**

Los diseños 3d pueden ocupar demasiado. Debemos encontrar una manera de reducir su peso acorde a los hardware que van a correr la aplicación, los móviles. El juego Orc'it soluciona este problema usando unos modelos poco poligonos.

La opción de crear el propio juego es una opción clave a desarrollar previo al lanzamiento. Su principal dificultad es la captura de una carta hecha rápidamente por el usuario. Para ello habrá que guardar en la base de datos distintas cartas con similar forma o encontrar una manera de detectar la figura dibujada.

Habrà que hacer un estudio sobre la rentabilidad de comprar una versión mejorada de vuforia de cara a aumentar la cantidad de diseños 3d disponibles en una pantalla. Hay que tener en cuenta, no solo el dinero si no los problemas de rendimiento que pueda ocasionar, ya que durante la creación del producto se dió una mayor importancia a la fluidez que a la detección en sí de las cartas.

### **Trabajo realizado.**

Lo primero que hice fue pensar la idea para la aplicación. A partir de los problemas que he tenido con mis amigos muchas veces. Una vez tenía la idea hice un estudio con los clientes de una tienda de juegos de mesa en mi pueblo. Estos me dieron otra idea, hacer la misma aplicación usando VR-AR. Al ver que a la mayoría le parecía una aplicación interesante no dudé en realizarla.

Elegí un nombre y me lancé a buscar aplicaciones similares y el software que usaría para desarrollarla. Como expliqué en el apartado de competencia, no había ninguna aplicación similar.

Para desarrollar la aplicación pensé directamente en vuforia y unity ya que los había utilizado con anterioridad.

Primero planteé hacer la aplicación de un solo juego, relacionada directamente con un juego de mesa en particular que quizás sea la idea más acertada debido a la carga de diseños 3d, pero decidí hacerla con un menú inicial para elegir el juego ya que a la hora de presentarla me parece que muestra un mayor potencial.

Para empezar instale unity y vuforia. Mi objetivo principal fue crear la escena de un juego bastante conocido pero de manera simplificada, el Gobb'it, de ahí el nombre de la escena y del juego, Orc'it.

Para comenzar a crear el juego necesitaba primero crear las cartas y añadirlas a la base de datos de vuforia. Hice las cartas con un generador de códigos bidi. Cada código bidi es una letra del abecedario, desde la 'a' hasta la 'k'.

Después de esto tuve los primeros problemas a la hora de usar la base de datos en unity. Estos problemas provenían de la licencia del producto. Una vez solucionados, pude asociar las imágenes de la base de datos a objetos en unity.

Mientras buscaba la manera de arreglarlo me di cuenta de que vuforia permite usar 2 opciones, priorizar la captura de la imagen para que no se equivoque entre carta y carta, cosa que tras testarlo vi que rara vez pasaba, o priorizar la velocidad. Esta segunda opción es la que utilicé por las mecánicas del juego.

Para hacerlo con una concepto similar a Goob'it busqué modelados 3d con animaciones. Estos modelos debían tener pocos polígonos para que no necesitase un hardware demasiado potente, ya que al fin y al cabo funcionará en un dispositivo móvil.

Asocié los modelos 3d a las cartas, y hice una animación en bucle para que todas se moviesen.

Probé el juego en mi móvil y vi que había un problema, el enfoque de la cámara. Así que me puse a modificar el script de cámara del vuforia para solucionar dicho

problema que consistía en que automáticamente fuese enfocando la imagen, para que no tengamos que ir pulsando la pantalla en la versión de vr-ar.

```
void Awake()
{
    // Check for an initialization error on start.

    VuforiaRuntime.Instance.RegisterVuforiaInitErrorCallback(OnVuforia
    InitializationError);
}

void Start()
{
    var vuforia = VuforiaARController.Instance;
    vuforia.RegisterVuforiaStartedCallback(OnVuforiaStarted);
    vuforia.RegisterOnPauseCallback(OnPaused);
    SetupGUIStyles();
}

private void OnVuforiaStarted()
{
    CameraDevice.Instance.SetFocusMode(
        CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
}

private void OnPaused(bool paused)
{
    if (!paused) // resumed
    {
        // Set again autofocus mode when app is resumed
        CameraDevice.Instance.SetFocusMode(
            CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
    }
}

void OnGUI()
{
    // On error, create a full screen window.
    if (mErrorOccurred)
        GUI.Window(0, new Rect(0, 0, Screen.width,
Screen.height), DrawWindowContent, "");
}

/// <summary>
```

```

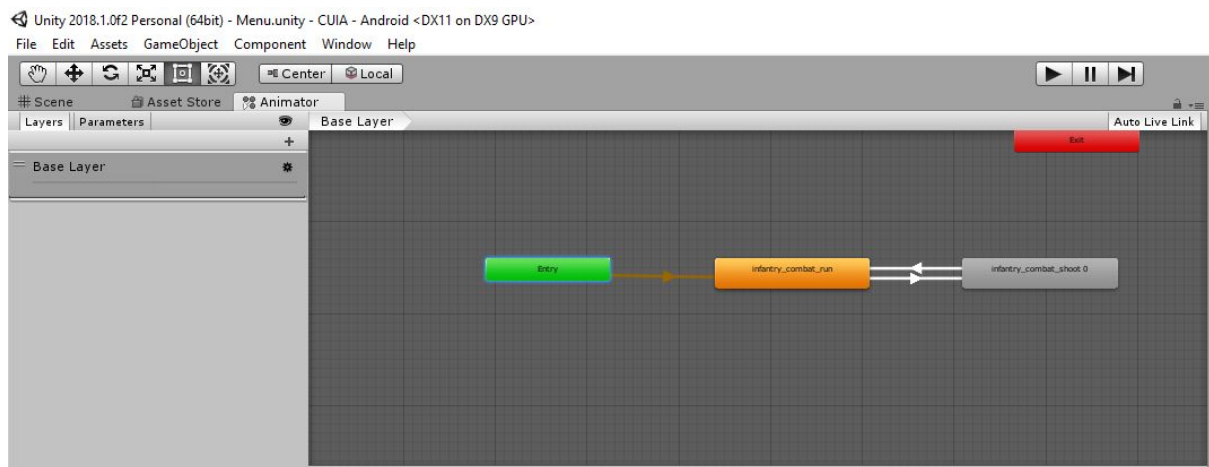
    ///      When this game object is destroyed, it unregisters
itself as event handler
    /// </summary>
    void OnDestroy()
    {

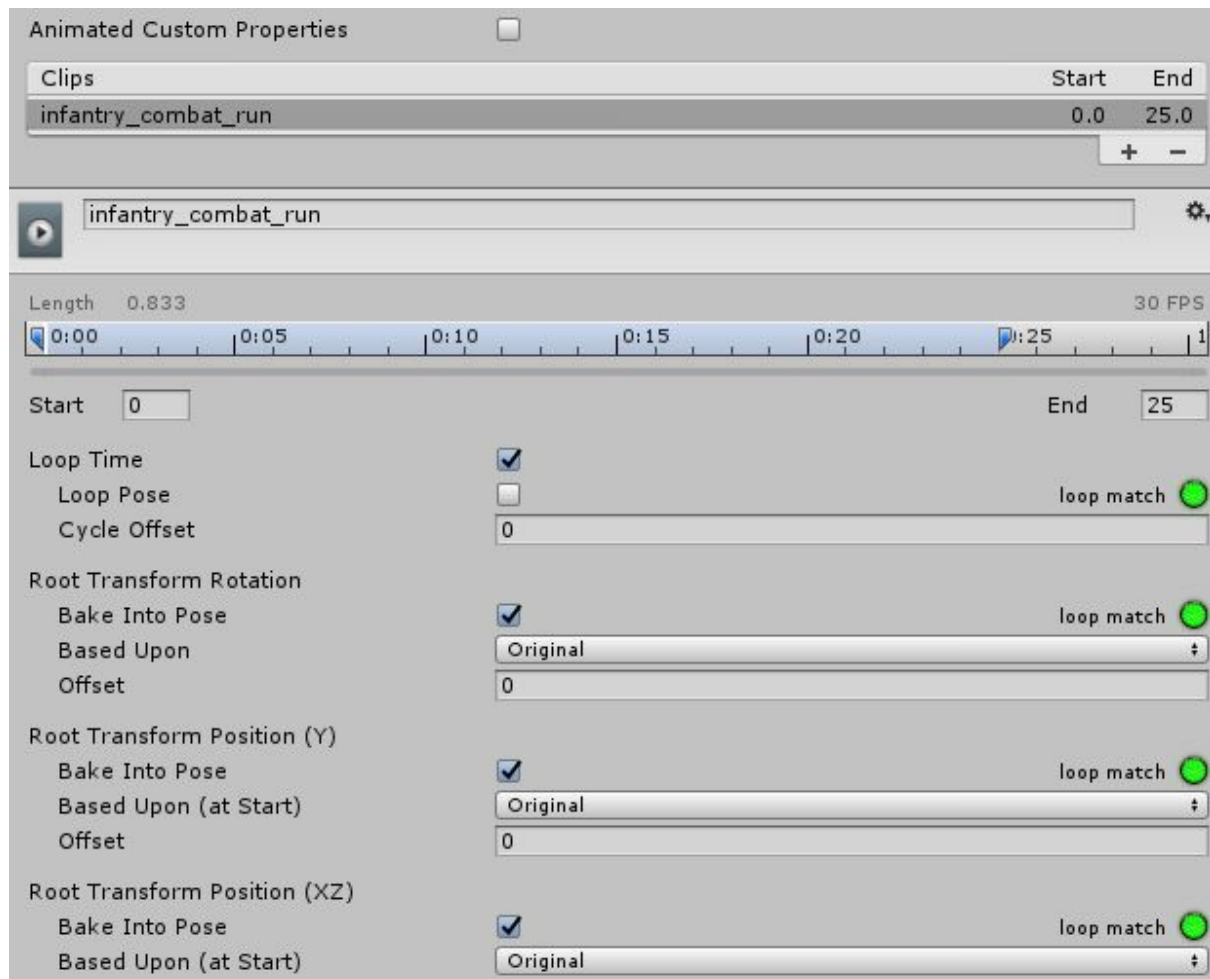
VuforiaRuntime.Instance.UnregisterVuforiaInitErrorCallback(OnVufor
iaInitializationError);
    }

```

Script modificado de la cámara.

Después, cree una nueva escena haciendo exactamente lo mismo pero para otro juego, zoomfightbie, que simplemente son modelos para un juego de tablero. En estos modelos no había ni siquiera un controlador de la animación, así que tuve que crearlo con las animaciones sueltas que traían lo assets.





Una vez acabadas las escenas que movían los 2 juegos, me puse a hacer el menú de entrada para poder seleccionar el juego que queremos usar. Tuve muchos problemas a la hora de crear un menu principalmente por el escalado. Al escalar a distinta resolución la pantalla los elementos del canvas se descolocaban. Para solucionar esto activé la opción de escalado con el tamaño de pantalla y puse una referencia de resolución de 243x390. Añadí una imagen que formaría mi menú y cree 2 botones para los 2 juegos utilizados. Ambos con un Rect Transform centrado y transparentes.

Para el funcionamiento de los botones y poderme mover entre escena cree un SceneManager, este controlaría que boton se ha pulsado y a que escena debe ir. Para no crear 2 funciones genéricas hice un script que nos permite asociar a cada botón la escena a donde ir.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
```

```

public class LevelManager : MonoBehaviour {

    public void CargaNivel(string pNombreNivel)
    {
        SceneManager.LoadScene(pNombreNivel);
    }
}

```

*Script LevelManager*

Al igual que necesitaba una manera de acceder a un juego también necesitaba una manera de volver al menú una vez seleccionemos un juego. Para ello estuve investigando que KeyCode estaba relacionada en unity con “back” en android. Este KeyCode era el escape, una vez encontré esto cree de nuevo un script y un canvas para asociárselo.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Back : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        if (Input.GetKeyDown(KeyCode.Escape)) {
            SceneManager.LoadScene("Menu"); }
    }
}

```

*Script escena anterior*

Tras hacer esto, la base del juego estaba completamente hecha. La compilé, cree la apk y lo probé todo en mi móvil. Ahora solo nos faltaría su versión VR-AR.

Esta versión ha sido bastante engorrosa de hacer debido a que la documentación del vuforia es muy mala y he tenido que ir probando poco a poco.

Primero fuí a Player Settings y activé:

Vuforia Augmented Reality Supported

## Virtual Reality Supported

En Virtual Reality SDKs añadí vuforia y las cardboard

Luego en la configuración de vuforia busqué Eyewear Type y le puse las siguientes opciones:

Phone + Viewer

Vuforia

Cardboard v1 (Google)

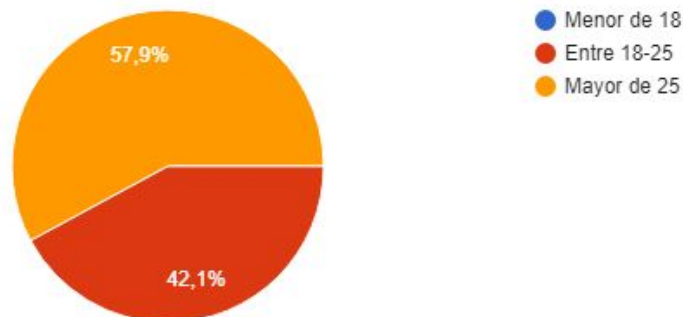
Tras esto, descargué un asset de unity de Vuforia con VR\_AR y ya empezó todo a funcionar de manera correcta. Compilé para una versión mínima de android necesaria.

Finalmente comprobé que todo iba bien en mi móvil y usando la cardboard.

## Estudios/cuestionarios realizados.

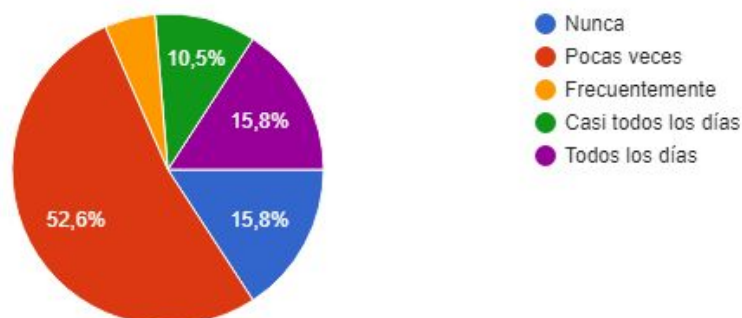
### Edad

19 respuestas



### ¿Juegas a juegos de móviles?

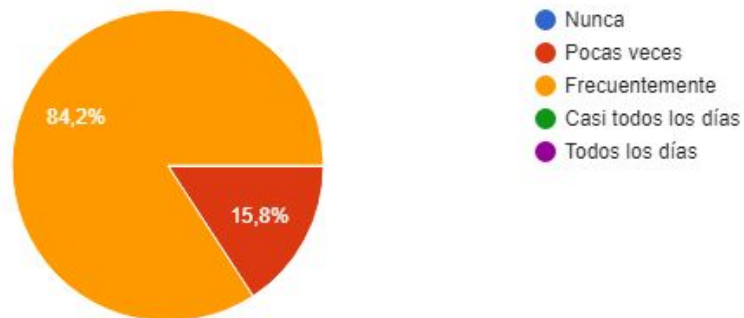
19 respuestas





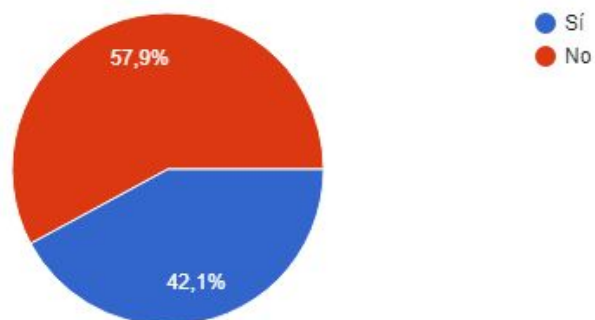
### ¿Juegas a juegos de mesa?

19 respuestas



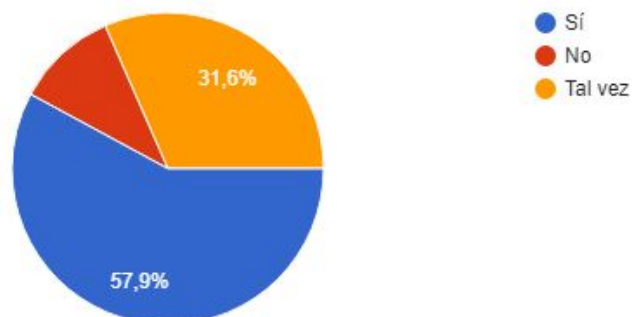
### ¿Has jugado a algún juego de realidad aumentada?

19 respuestas



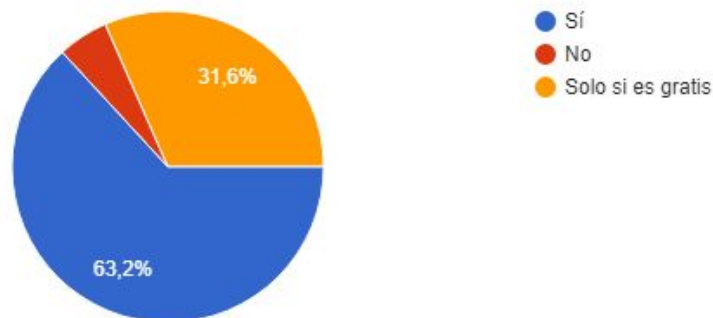
### ¿Jugarías a juegos de mesa con realidad aumentada?

19 respuestas



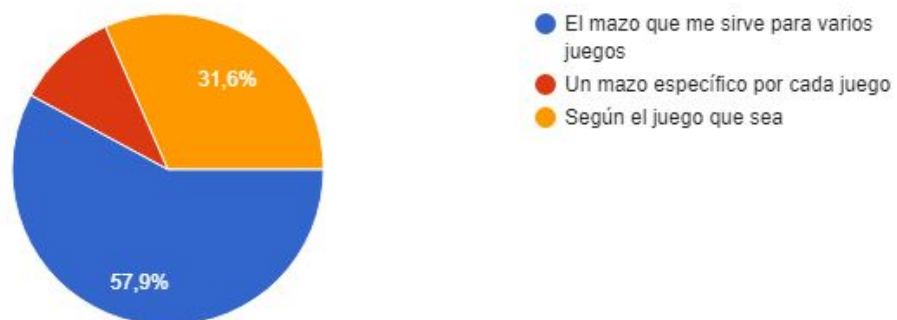
Si hubiese un complemento que te permitiera visualizar los modelos que aparecen en los juegos de cartas en realidad aumentada, ¿lo usarías?

19 respuestas



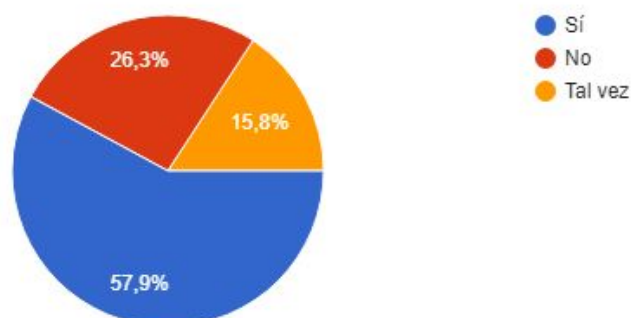
Si con un solo mazo de cartas pudieses jugar a varios juegos de mesa completamente distintos sin un gasto ...o mazo o el específico de cada juego?

19 respuestas



Si para una mejor inmersión pudieses usar cardboards (imagen) , ¿las usarías?

19 respuestas



## **Documentación utilizada**

*<https://library.vuforia.com/>*

*<https://docs.unity3d.com/Manual/vuforia-sdk-overview.html>*

*<https://code.tutsplus.com/es/tutorials/introducing-augmented-reality-with-vuforia--cms-27160>*

*<http://gibsonmartelli.com/SpacePlace/2016/01/06/integrate-ar-vr-vuforia-unity/>*

**Realizado por: Javier Oliva Cruz  
50642411E**