

LA REALIDAD AUMENTADA:

APLICACIONES ACCESIBLES PARA LA INFORMACIÓN Y LA EDUCACIÓN



Joan Esgleas Tafalla | Segundo de Bachillerato | 2019/2020

Col·legi Sant Marc de Barcelona | Tutor: Israel Merino

“A key part of that journey is making an open platform where any developer can create anything they want.”

— Mark Zuckerberg

Me gustaría agradecer el soporte y la enorme ayuda prestada por mi tutor Israel Merino, que pese a todas las adversidades presentadas ha encontrado tiempo para orientarme y enseñarme conceptos imprescindibles para este trabajo. También quiero agradecer las horas que mi padre, Jordi Esgleas, ha invertido para ayudarme a redactar este trabajo y a programar las aplicaciones RA.

Muchas gracias.

Índice

1. INTRODUCCIÓN	6
1.1 Problema	6
1.2 Hipótesis	7
1.3 Objetivos de este trabajo	7
1.4 Justificación	9
1.4.1 Justificación personal	9
1.4.2 Justificación general	10
2. PARTE TEÓRICA	11
2.1 ¿Qué es la RA?	11
2.2 Historia de la RA	13
2.3 Diferencias entre la RA y la RV	17
2.4 ¿Cómo funciona la RA?	18
2.5 Aplicaciones actuales de la RA	20
2.6 Posibles aplicaciones de la RA	23
3. PARTE PRÁCTICA	25
3.1 Introducción	25
3.2 Aplicación para identificar carteles de películas	27
3.2.1 Crear un software que pueda trabajar en un teléfono y compatible con la RA	28
3.2.2 Crear un sistema RA	29
3.2.3 Crear un sistema para distinguir imágenes y seguirles el movimiento	29
3.2.4 Crear una interfaz donde aparezca la información de la película	30
3.2.5 Crear un contenedor de datos donde almacenar toda la información de cada película	31
3.2.6 Resultado	32
3.3 Aplicación para representar puntos rectas y planos en RA	33
3.3.1 Crear un eje de coordenadas en RA	33
3.3.2 Situar el eje de coordenadas en el espacio RA	37
3.3.3 Crear una UI que te permita crear elementos y asignarles ecuaciones	38
3.3.4 Crear un sistema que cree los elementos	39
3.3.5 Hacer estos elementos visibles en RA	43
3.3.6 Crear un sistema que marque la intersección de dos planos como una recta en un espacio RA	43
3.3.6.1 Detectar los dos planos	43
3.3.6.2 Archivar los componentes de los planos	45
3.3.6.3 Crear la recta con la ecuación que represente la intersección	45

3.3.7 Resultado.....	48
4. CONCLUSIÓN.....	49
5. BIBLIOGRAFÍA.....	51

1. INTRODUCCIÓN

1.1 Problema

La realidad aumentada (de ahora en adelante RA) es una tecnología con diversas aplicaciones en el día a día ya que permite transferir imágenes digitales en un espacio tridimensional que puede coincidir con el mundo físico. Esta tecnología ha sido casi exclusivamente desarrollada para el ocio y las redes sociales.

También ha sido desarrollada para otros campos, pero muy poco accesible para el público en general debido al hardware o a la dificultad de uso.

Sin embargo, la RA presenta infinitas oportunidades para ser implementada de manera accesible en múltiples terrenos. Entre ellos destacan: La publicidad, que se vería ampliamente beneficiada en la aplicación estas tecnologías, pudiendo incrementar la interactividad con sus productos. La RA también contribuiría a una mejor accesibilidad de la información. Por otra parte, la RA ayudaría al comercio online facilitando el acceso a los puntos de venta y a sus productos. Por último, en mi opinión, la RA podría ser aplicada a la educación con amplio éxito debido a la capacidad que tiene esta tecnología de visualizar conceptos abstractos y de esta forma facilitar la adquisición de conocimientos.

Sin embargo, la RA no ha sido desarrollada en amplitud en los campos anteriores y eso, en mi opinión, es una enorme oportunidad desechada. Creo que la RA podría mejorar en muchos aspectos muchas disciplinas (la publicidad, la

industria...). Por lo tanto, voy a dedicar mi TR a mejorar la accesibilidad de la RA en la educación y en la búsqueda de información. En el caso de este trabajo mejoraré la accesibilidad a la información referente a películas y la capacidad de visualizar conceptos matemáticos como planos, rectas y puntos en un espacio real.

1.2 Hipótesis

Mi TR se centrará en la investigación y la creación de herramientas accesibles RA, que puedan ser aplicadas y usadas en campos cotidianos. Por lo tanto, mi hipótesis es la siguiente: Hoy en día, resulta posible la creación de herramientas de realidad aumentada para mejorar determinados ámbitos como la educación y el acceso a la información, que son los que han sido objeto de proyectos informáticos en el presente trabajo.

1.3 Objetivos de este trabajo

Con este trabajo tengo diversos objetivos. Mi primer objetivo es averiguar si la RA es una tecnología con aplicaciones útiles para nuestro día a día y que nos ayude en algún ámbito concreto o si por lo contrario la RA como tecnología no puede aportar nada al público en general.

Mi segundo objetivo es desarrollar una aplicación que nos ayude a acceder a la información de una manera más cómoda y sencilla, y me gustaría centrar esta aplicación en el cine.

Mi tercer objetivo es encontrar una aplicación de esta tecnología a la educación debido a que creo que este campo se vería extremadamente beneficiado por esta tecnología.

Mi cuarto objetivo es aprender a desarrollar una aplicación para teléfono debido a que es una cosa que no he hecho nunca, aprender cómo funciona un motor gráfico y un lenguaje de programación.

Mi quinto objetivo es aprender cómo funciona la RA, y ser capaz de entender y dominar el fundamento matemático que yace detrás de esta tecnología.

Y mi último objetivo consiste en hacer de la RA una tecnología mas accesible para el público en general

1.4 Justificación

1.4.1 Justificación personal

Desde que jugué a Invizimals en la PlayStation Portable (Figura 1) me ha fascinado la RA, ver aquellos monstruos tomar vida en el mundo real me pareció casi mágico hasta el punto de obsesionarme con aquellas criaturas, su historia, sus habilidades... Incluso esta obsesión me llevó a completar una colección de más de 470 cromos.

Pero lo que más me fascinó sin ninguna duda fue la tecnología salida de una película ciencia ficción que era capaz de hacer que aquellos monstruos salieran de la pantalla y se alojarán en el mundo real.



Figura 1: Imagen del juego Invizimals para la PlayStation Portable.

Fuente: Game España, n.d.

Desde que jugué a ese juego con 6 años he intentado comprender cómo funciona la RA y siempre he estado convencido de que esta tecnología de ciencia ficción tenía muchos más usos. No sólo el de un videojuego sino también para las comunicaciones, la educación etc.

1.4.2 Justificación general

La RA es una tecnología muy útil pero muy poco desarrollada y accesible fuera de ámbitos muy concretos (la comunicación y el entretenimiento) pero podría ayudarnos en otros campos.

Yo quiero que mi trabajo de investigación encuentre nuevas aplicaciones a la RA y en la medida de lo posible desarrollar herramientas para otros campos que no sean los anteriormente expuestos.

La RA puede ser una muy buena herramienta para facilitarnos el acceso a la información y también puede resultar como una buena manera de enseñar, así que creo que desarrollar y explorar aplicaciones de la RA en este campo puede ser muy positivo.

2. PARTE TEÓRICA

2.1 ¿Qué es la RA?

La RA es una visión del mundo físico modificada por un sistema informático (Figura 2), de la cual algunos elementos reales son sobreimpuestos o extraídos en los siguientes sentidos: sonido, gráficos, video y hápticos.(Schueffel, 2017).



Figura 2: Ejemplo de RA.

Fuente: James Martin/CNET, 2020.

Es decir, la RA es ver la realidad y modificarla mediante un software empleado técnicas y sensores integrados en el dispositivo para situar los objetos digitales en un espacio y el renderizado de gráficos 3D en tiempo real para hacer tele presentes estos objetos a nuestro dispositivo.(Steure, 1993)

Estos objetos digitales son llamados “**augogram**”. Y la práctica para proyectarlos en la realidad es llamada “**augography**”.(Yen, Tsai, & Wu, 2013)

La realidad mejorada puede ser constructivo es decir añadir a la realidad o destructivo ocultando y remplazando objetos de la realidad. (Rosenber, 1992)

Según Rosenber la realidad mejorada puede desasociar al usuario de la realidad haciendo que se sumerja en un mundo creado por computador (la actual realidad virtual) o hacerlo más consciente del mundo físico que le envuelve añadiendo

información al mundo real (la actual realidad aumentada) con el objetivo de que el usuario sea más consciente de sus alrededores.

La realidad mediada por computadora se divide en varias subcategorías según el nivel de remplazo de la realidad llevado a cabo por una computadora: la realidad virtual (RV), la realidad aumentada (RA) y la realidad mixta (RM).

La realidad virtual (Figura 3): Consiste en la creación de espacios virtuales donde interactúan objetos y agentes tanto reales como virtuales. (Milgram & Kishino, 1994)



Figura 3: Elejemplo realidad virtual.

Fuente: Elso tanoperdido, n.d.

La realidad aumentada: Consiste en la adición y/o supresión de elementos de una realidad definida.(Mann, 1996).

La realidad mixta (Figura 4): Combina espacios tanto reales como virtuales donde agentes reales y virtuales interactúan.(Milgram & Kishino, 1994)



Figura 4: Ejemplo de realidad mixta.

Fuente: Autodesk, n.d.

Este trabajo se centrará en la RA es decir en la realidad mediada por computadora.

2.2 Historia de la RA

- 1901: L. Frank Baum , un autor de ficción menciona por primera vez un dispositivo que proyecta información en la vida real llamado “character marker” (L. Frank Baum, n.d.)
- 1957–62: Morton Heilig, director de fotografía, crea y patenta un simulador llamado Sensorama con imágenes, sonido, vibración y olor.(Heilig, 1962)
- 1968: Ivan Sutherland inventa la pantalla montada en la cabeza y la posiciona como una ventana a un mundo virtual.(Sutherland, 1968)
- 1975: Myron Krueger crea Videoplace para permitir a los usuarios interactuar con objetos virtuales.(Steure, 1993)
- 1980: La investigación realizada por Gavan Lintern de la Universidad de Illinois es el primer trabajo publicado que muestra el valor de una pantalla de visualización para enseñar habilidades de vuelo en el mundo real (Lintern, 1980)
- 1980: Steve Mann crea la primera computadora portátil, un sistema de visión por computadora con texto y superposiciones gráficas en una escena fotografiada.(Mann, 1996)
- 1986: Dentro de IBM, Ron Feigenblatt describe la forma de RA más experimentada en la actualidad (a saber: "ventana mágica" COMO por ejemplo,

Pokémon Go basado en teléfonos inteligentes), el uso de una pequeña pantalla plana "inteligente" colocada y orientada a mano.(IBM & Feigenblatt, 2005)

- 1987: Douglas George y Robert Morris crean un prototipo funcional de un sistema de "pantalla de visualización frontal" basado en un telescopio astronómico (un concepto precursor de la realidad aumentada) que se superpone en el ocular del telescopio, sobre las imágenes reales del cielo, estrella de intensidad múltiple, e imágenes del cuerpo celeste, y otra información relevante. (Miller, 2008)
- 1990: El término "realidad aumentada" usado por primera vez por Thomas P. Caudell, un ex investigador de Boeing.(Lee & Billinghamurst, 2012)
- 1992: Louis Rosenberg desarrolló uno de los primeros sistemas de RA en funcionamiento, llamados "Virtual Fixtures", en el Laboratorio de Investigación de la Fuerza Aérea de los Estados Unidos.(Rosenberg, 1992)
- 1992: Steven Feiner, Blair MacIntyre y Doree Seligmann presentan un diseño de un prototipo de sistema de RA llamado KARMA.
- 1993: CMOS, un active-pixel sensor de metal-óxido-semiconductor fue desarrollado en la NASA, estos sensores son muy usados para la localización óptica en la tecnología RA actual.(Cui, Kharel, & Gruev, 2017)

- 1994: Julie Martin crea el primer espectáculo usando la RA llamado Dancing in Cyberspace.(Casebook, 2014)
- 1999: Frank Delgado, Mike Abernathy crean un sistema para proyectar imágenes de mapa encima imágenes aéreas de ciudades, caminos etc.(Delgado, Abernathy, White, & Lowrey, 1999)
- 1999: US Naval Research Laboratory empieza un investigación de las aplicaciones de la RA en el campo de batalla llamado (BARS).(Lintern, Roscoe, & Sivier, 1990)
- 2000: Rockwell International Science Center, muestra un sistema de RA inalámbrico para orientarse.(Behringer, Tam, McGee, Sundareswaran, & Vassiliou, 2000)
- 2009: ARToolkit (Librería que he usado en la parte práctica) es lanzada en Adobe Flash.
- 2009: Se lanza el videojuego de Invizimals.
- 2014: Snapchat lanza los populares filtros RA.
- 2015: Microsoft anuncia HoloLens.

- 2016: Niantic lanza Pokemon Go llevando a la RA interactivo a gran popularidad.
- 2017: Apple lanza al mercado ARKit (Apple, 2019)
- 2018: Unity anuncia el soporte RA para su motor grafico con ARCore(Alamparambil & Mowrer, n.d.)
- 2019: Microsoft anuncia HoloLens 2

A partir del año 2000 empieza a haber un desarrollo masivo de esta tecnología así que he hecho una síntesis de los productos, descubrimientos etc. Que han sido importantes, que han desarrollado la RA de manera significativa i/o han influenciado mi TR.

2.3 Diferencias entre la RA y la RV

A pesar que la RA y la RV tienen un origen común como realidad mejorada ya que comparten mucha teoría matemática (Rosenber, 1992) recientemente se han separado en dos ramas distintas.

Hoy en día entendemos la RV como la experiencia que experimentamos con los dispositivos RV (las gafas HTC Vive, Oculus Rift ...). En cambio, entendemos la RA como la superposición de imágenes virtuales sobre la realidad.

Así como en nuestro día a día raramente tenemos contacto con la RV la RA, en estos últimos años, se ha convertido en la tecnología más comúnmente usada debido al incremento de popularidad de productos para el público general como los populares filtros de Snapchat y videojuegos como Pokémon GO.

Otra importante diferencia es como se muestran las imágenes al usuario. La RV usa un sistema que consiste en mostrar una imagen desde dos perspectivas distintas, estas imágenes son proyectadas directamente a los ojos desde unas gafas que anulan nuestra visión periférica, consiguiendo así la sensación de tridimensionalidad ((1) *MIT Explains: How Does Virtual Reality Work?* - YouTube, n.d.), en cambio la RA usa una pantalla para mostrar una fusión de las imágenes capturadas por una cámara y las imágenes generadas digitalmente.

2.4 ¿Cómo funciona la RA?

Debido a la complejidad que presenta la creación de espacios RA, en este punto se explica de una forma sintética y sencilla como funciona la RA.

El primer paso es la creación de un modelo matemático del espacio en el cual se encuentra el dispositivo.

Para crear ese modelo matemático se usa un sistema de seguimiento de movimiento de la cámara, es decir registrar los movimientos de la cámara en referencia al espacio. Hay dos métodos para seguir el movimiento de la cámara. El primer método consiste en una técnica llamada “*Depth tracking*”. Consiste en registrar puntos de contraste en la imagen capturada por cámara (Figura 5 y 6) y calcular la distancia entre cada uno de ellos, de esta forma el sistema consigue calcular la posición de la cámara respecto a esos puntos creando así un modelo matemático del espacio (Figura 7). También se pueden añadir algunos sensores como por ejemplo, un sensor de profundidad que ayuda al sistema RA a ser más preciso con su “*Depth tracking*”.

Otro método común es el uso combinado acelerómetros, GPS, giroscopios, compases en estado sólido y RFID para triangular la posición y el ángulo de la cámara, y así crear un modelo matemático del espacio en el que se encuentra el dispositivo RA.

Actualmente los dispositivos RA usan una combinación de estos dos métodos para crear el modelo matemático del espacio (Apple, 2019).

Una vez el modelo matemático del espacio está creado, una imagen digital es renderizada, replicando la perspectiva de la cámara.

Después la imagen captada por la cámara y las imágenes renderizadas por el dispositivo son superpuestas. Creando así la ilusión de que un objeto digital está en el mundo físico (Figura 8).

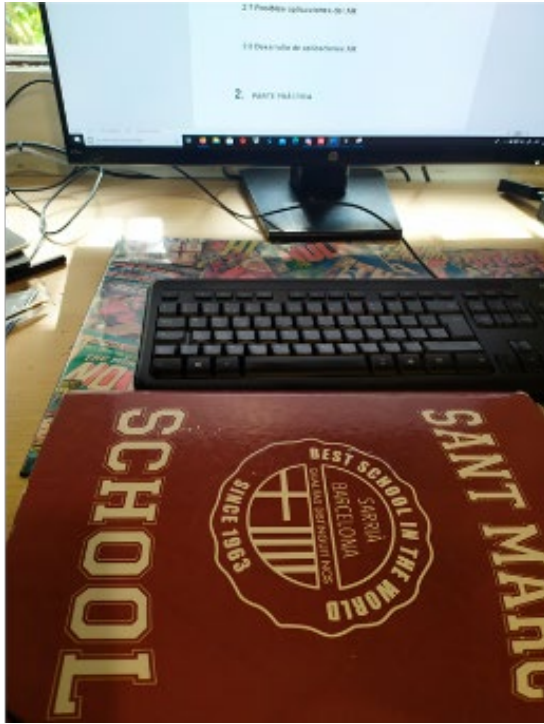


Figura 5: La cámara registra una imagen.

Fuente propia.

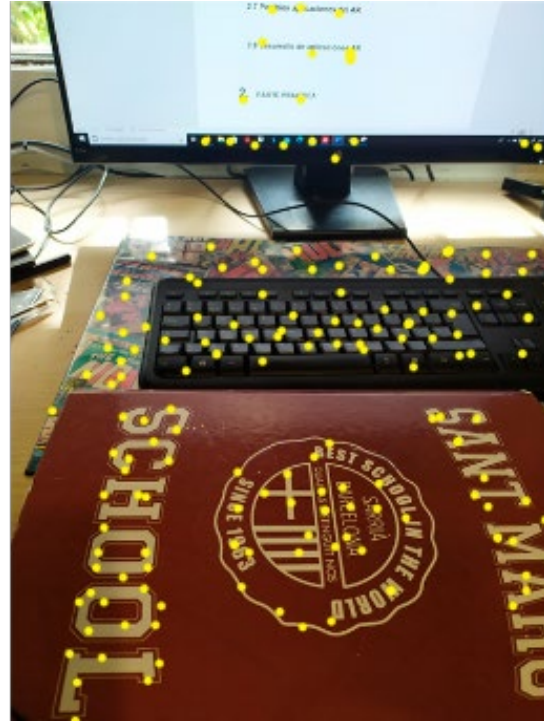


Figura 6: Se detectan los puntos de contraste.

Fuente propia.

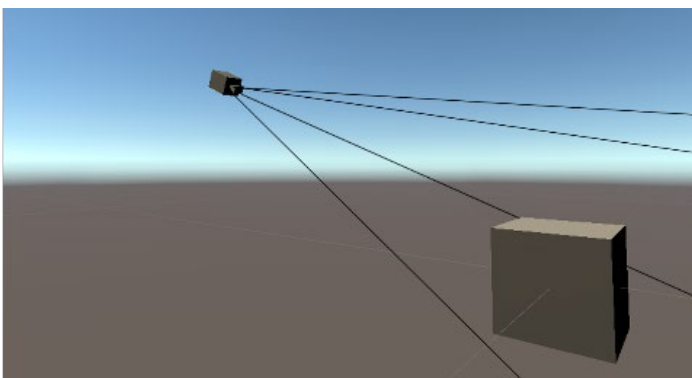


Figura 7: Se crea un modelo matemático del espacio y se crea una imagen digital.

Fuente propia.

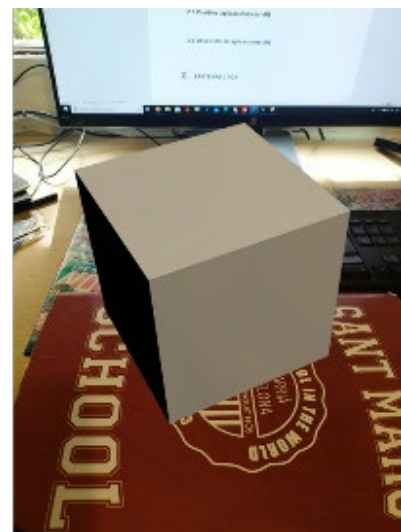


Figura 8: Se fusionan ambas imágenes.

Fuente propia.

2.5 Aplicaciones actuales de la RA

La RA es una tecnología que en los últimos años ha sido desarrollada en muchos ámbitos gracias a la creación de software y herramientas para desarrollar aplicaciones RA como ARToolkit y ARCore. Debido a la relativa facilidad que supone para las empresas desarrollar herramientas RA hoy en día la RA ha sido aplicado en muchos campos.

Según Tractica (un centro de investigación de tecnologías emergentes) este gráfico muestra las descargas de aplicaciones activas basadas en la RA en distintos campos (juegos, industria, información...). (Gráfico 1)

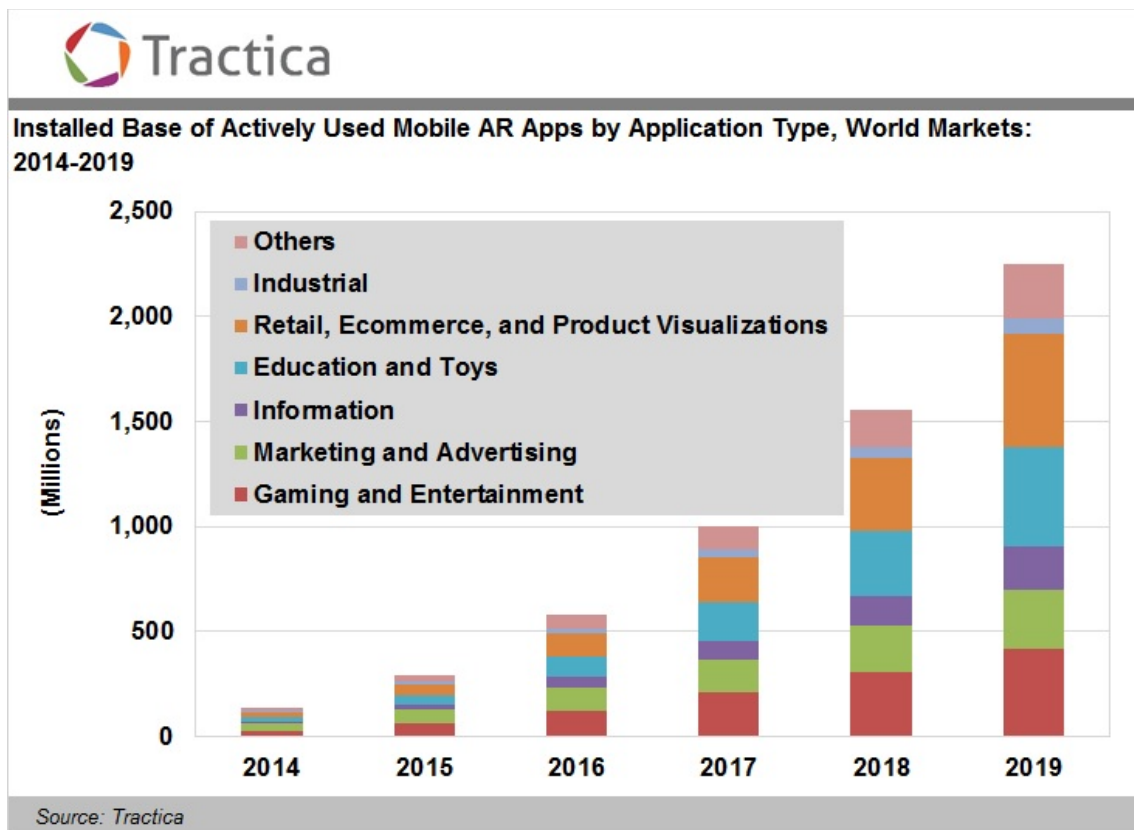


Gráfico 1: Descargas de aplicaciones activas basadas en la RA.

Fuente: Tractica Omdia, n.d.

Tal como muestra el gráfico la RA ha tenido mucha incidencia en los campos del comercio, del entretenimiento y de la educación

En el campo del comercio cabe destacar las aplicaciones de visualización de productos en RA por ejemplo IKEA place. (Figura 9)



Figura 9: Uso de la aplicación IKEA Place.

Fuente: Xataka, 2017.

En el campo del entretenimiento la incidencia de la RA ha sido muy importante gracias a los videojuegos en concreto Pokémon GO. (Figura 10)



Figura 10: Uso de la aplicación Pokemon GO

Fuente: Niantic, n.d.

La RA también está empezando a incidir en el mundo de la educación y de la formación gracias a la implementación de estas tecnologías en museos por ejemplo el “Museo nacional de Singapur” en su exposición llamada Story of the Forest. La exposición se centra en 69 imágenes de la Colección William Farquhar de dibujos de historia natural. Estos han sido convertidos en animaciones tridimensionales con las que los visitantes pueden interactuar. (“How Museums are using Augmented Reality - MuseumNext,” n.d.)

Otra aplicación importante mencionar es la versión de GeoGebra, una aplicación de cálculo y representación gráfica de conceptos matemático, para teléfono móvil que incluye una funcionalidad RA, en la cual me he basado para hacer la aplicación en la parte práctica. (Figura 11) (“GeoGebra AR – GeoGebra,” n.d.)

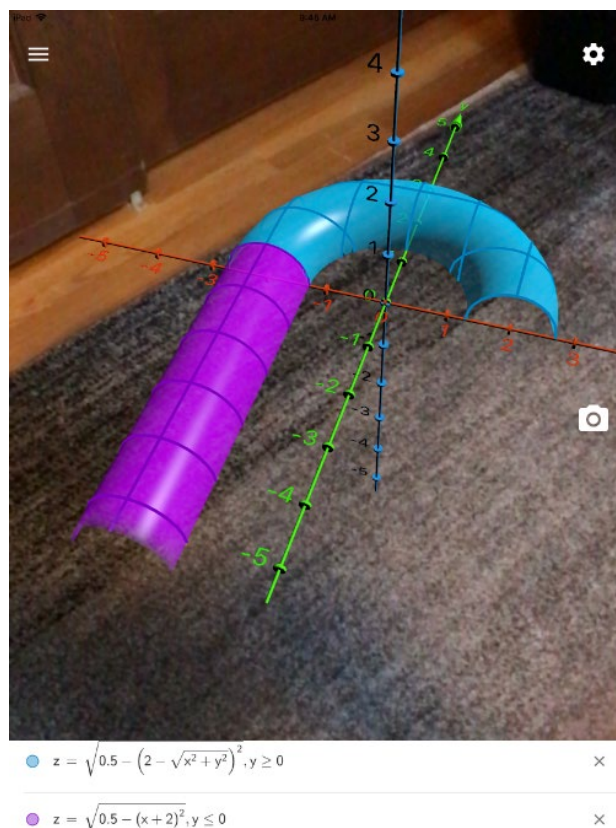


Figura 11: Uso de la aplicación GeoGebra AR

Fuente: Tim Brzezinski, n.d.

Por último en menor medida la RA ha tenido una incidencia en la industria como muestran empresas como RICOH que emplean la tecnología RA para facilitar la logística i la distribución de productos. (Figura 12) (“(38) DHL tests augmented reality in warehouse order-picking - YouTube,” n.d.)

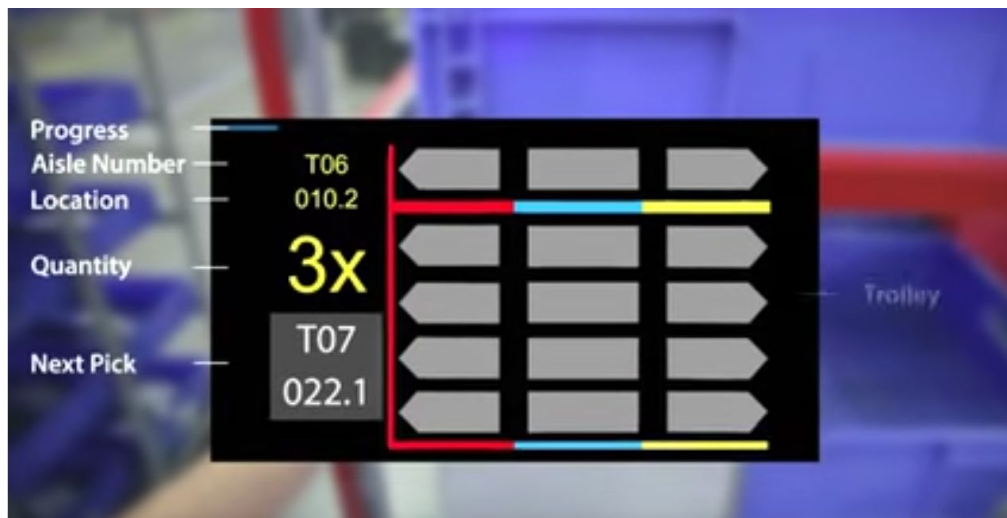


Figura 12: Ejemplo del uso de la tecnología RA por RICOH

Fuente: Eurobuild TV 2015.

2.6 Posibles aplicaciones de la RA

A pesar de que la RA ya está siendo aplicado en muchos ámbitos, al ser una tecnología recientemente desarrollada, está en constante actualización. Esto lleva a que se desarrollen constantemente actualizaciones y se le encuentren nuevos usos a la RA.

El campo de la arquitectura y diseño urbano podría beneficiarse mucho de esta tecnología tal como muestra la aplicación creada por la universidad de Canterbury llamada CityViewAR, la cual permitió ver a los ingenieros y arquitectos los edificios que habían sido destruidos por el terremoto Christchurch en Canterbury (GB) en 2012. (Figura 13) (Lee & Billinghamurst, 2012)



Figura 13: Uso de la aplicación CityViewAR

Fuente: Hitlabnz 2010

Esta aplicación muestra el potencial de esta tecnología en este campo y empresas como Daqri y Trimble Inc. han empezado a desarrollar tecnología basada en la RA para el sector de la arquitectura.

En el campo de la industria la RA ya ha sido probada, la RA tiene mucho más potencial en el ámbito de la formación y la seguridad de los trabajadores de la industria tal como apuntan Marco Mandolini de la “Università Politecnica delle Marche” en el libro “Advances on mechanics, design engineering and Manufacturing II”(Cavas-Martínez et al., n.d.)

3. PARTE PRÁCTICA

La parte práctica de este TR consistirá en la creación de dos aplicaciones para teléfono que usen la RA.

Una usará la RA para mostrar información sobre una película, así que la RA puede ser usado para mostrar y facilitar el acceso a información.

En segundo lugar, desarrollaré una herramienta que podrá ser aplicada en el mundo de la educación, en el campo de la educación la RA podría tener más incidencia de la que actualmente ya tiene.

En este punto he mencionado algunos campos en los cuales se está explorando la aplicación de la RA, pero al ser una tecnología potente y en actual desarrollo la RA aún puede ser aplicado en muchos otros.

3.1 Introducción

Mi objetivo principal de la parte práctica es crear usos accesibles de la RA y que puedan ser útiles en algún campo en concreto. Teniendo en cuenta la accesibilidad decidí usar el teléfono como hardware en vez de gafas u ordenadores ya que un teléfono es más accesible y un buen soporte de RA.

Una vez tuve claro que hardware iba a usar empecé a investigar cómo hacer la aplicación, primero investigué algunas opciones como las IDE (Integrated Development Environment o entorno de desarrollo integrado) Android Studio o Xcode pero al final me decidí por el motor gráfico Unity, el mismo que usó Niantic para desarrollar Pokémon GO, debido a que es posible usar una versión gratuita para proyectos no comerciales y puedo crear builds *“proceso por el cual el código*

fuente es transformado a un programa que puede ser ejecutado por sí mismo en un dispositivo” (“What is Build? - Definition from Techopedia,” n.d.) tanto para Android como para Apple. Además, el lenguaje de programación que usa es, **C#** un lenguaje de programación con el cual estoy muy familiarizado.

“C# Es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes”.(ECMA-334 5th Edition / C# Language Specification, 2017).

Una vez me decidí con el motor, encontré que existían diversos SDK para trabajar con RA *“Un SDK o “software development kit” es un conjunto de herramientas para desarrollar un tipo de software en concreto*” (“CCNA Data Center DCICT 640-916 Official Cert Guide - Navaid Shamsee, David Klebanov, Hesham Fayed, Ahmed Afrose, Ozden Karakok - Google Llibres,” n.d.) empecé usando Vuforia, pero luego decidí cambiar a XRCore debido a que Vuforia tenía muchas restricciones a la hora de programar nuevas funciones.

Una vez tuve las herramientas necesarias, para desarrollar la aplicación, planteé una lluvia de ideas para decidir la función de las aplicaciones. La primera idea que tuve fue desarrollar una aplicación cuya funcionalidad consistiera en que, al enfocar a un cartel se muestre en forma de realidad aumentada información de la película, el tráiler, donde verla, etc.

La segunda idea que decidí seguir fue, crear una herramienta para representar puntos, rectas y planos en un espacio de RA. Similar a la aplicación de GeoGebra, pero más simplificado y accesible.

3.2 Aplicación para identificar carteles de películas

Una vez definidos los dos temas sobre los que aplicaría la RA, empecé a planificar que pasos tenía que seguir para llevar a cabo la aplicación. Para hacer ello use Trello, un programa para organizar y programar tareas.(Figura 14)

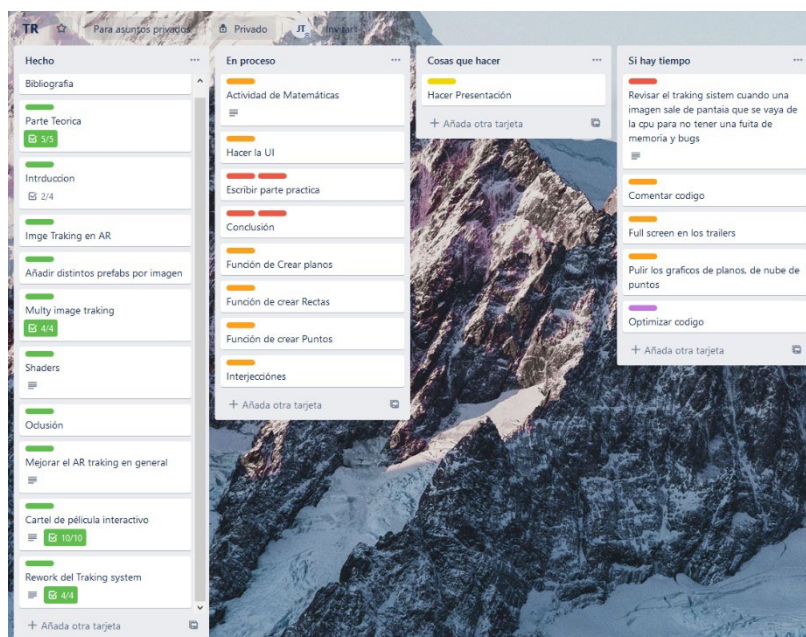


Figura 14: Captura de mi tabla de Trello.

Fuente propia.

Para hacer este proyecto necesitaba seguir los siguientes pasos.

- Crear un software que pueda trabajar en un teléfono y compatible con la RA
- Crear un sistema RA
- Crear un sistema para distinguir imágenes y seguirles el movimiento
- Crear una interfaz donde aparezca la información de la película

- Crear un contenedor de datos donde almacenar toda la información de cada película

3.2.1 Crear un software que pueda trabajar en un teléfono y compatible con la RA

Primero decidí empezar a intentar crear una aplicación básica en la cual sólo se podía ver un cubo y empecé a experimentar con las relaciones de aspecto (la relación de aspecto es la proporción entre el ancho y la altura de la imagen (*Anamorphic Now*, n.d.)) de un móvil, las definiciones etc. Después tuve que crear un proceso de segmentación usando Android Studio y Unity para poder generar versiones para teléfono de una manera eficiente. Este proceso de segmentación funciona de la siguiente forma: Unity hace la primera “build” con la ayuda de una API (Application Programming Interfaces). Una API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos de Android Studio, esta “build” crea una APK (Application Package Kit) que es copiada al teléfono, una vez la APK está en el teléfono es descomprimida automáticamente y una vez descomprimida es ejecutada. Así puedo hacer modificaciones y probarlas en el teléfono de una manera cómoda y rápida.

3.2.2 Crear un sistema RA

Una vez terminado el proceso de segmentación el sistema RA tenía que ser implementado. Para implementar la RA se ha de acceder a información captada por sensores del teléfono como la cámara o el giroscopio, por suerte el SDK de XRCore ya tiene una herramienta que accede a la cámara y al giroscopio automáticamente y lo vincula automáticamente a una cámara virtual en el motor gráfico para que la perspectiva sea la misma.

3.2.3 Crear un sistema para distinguir imágenes y seguirles el movimiento

Una vez creada las bases de la aplicación tenía que crear un sistema que identificara el cartel de cada película y le siguiera la posición. Decidí empezar por la parte que permitiera seguir la posición de una imagen determinada. Para ello empecé a investigar y encontré que el SDK ya tiene implementada una función para hacerlo, tras modificar un poco esta herramienta conseguí que funcionara.

Esta herramienta funciona de la siguiente forma: se registran todos los colores de los píxeles de la pantalla en cada fotograma. La serie de colores es registrada y comparada con la de una base de datos la cual tiene registrada todas las portadas de películas que quiero identificar, si coinciden, se crea un espacio digital y tiene como punto de origen el centro de la imagen.

Una vez este sistema ha sido implementado tenía que buscar un modo de distinguir la imagen y crear un espacio digital distinto por cada cartel. Para ello tuve que reescribir el código de la herramienta del SDK que sigue a imágenes determinadas para asignarles a cada imagen que identificaba un nombre. Una

vez el software tiene el nombre puede distinguir que espacio digital crear, y donde crearlo.

3.2.4 Crear una interfaz donde aparezca la información de la película

Una vez que las bases de la aplicación estaban completadas diseñé una interfaz (Figura 15 y 16), usando las herramientas del motor de Unity para luego enlazarlas a un contenedor de datos donde alojar todos los datos de cada una de las películas.

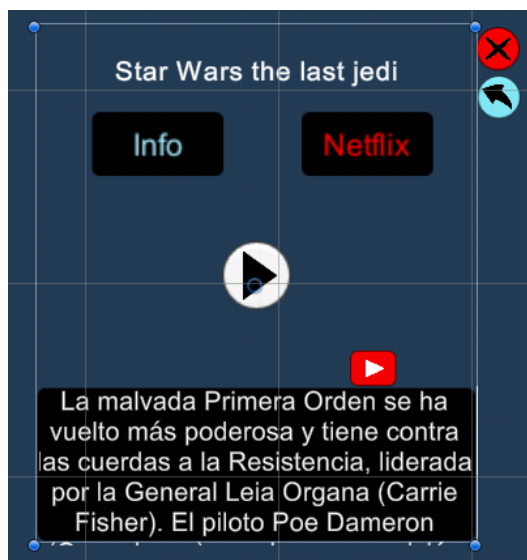


Figura 15: UI del menú inicial de al capturar una película.

Fuente propia.

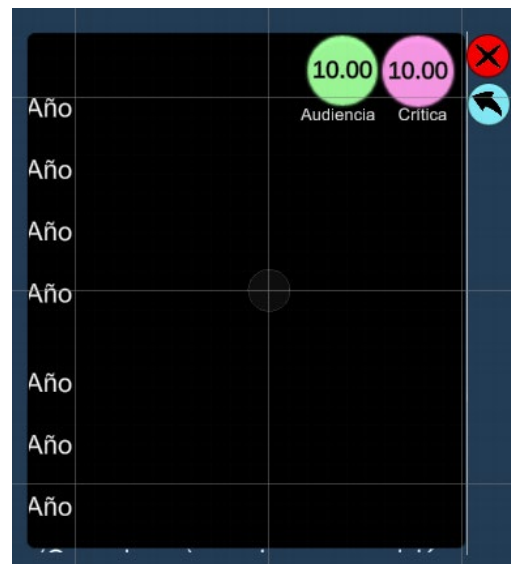


Figura 16: UI de la información ampliada.

Fuente propia.

3.2.5 Crear un contenedor de datos donde almacenar toda la información de cada película

Una vez tuve una interfaz funcional empecé a preparar un contenedor de datos, primero me decanté para hacerlo en forma de base de datos de una forma similar a la forma que funciona el reconocimiento del cartel, pero encontré una manera más eficaz de hacerlo usando una herramienta de Unity llamada "Scriptable Object". Un "Scriptable Object" es un contenedor de información a la cual se puede acceder de manera libre, con este contenedor organizar y acceder a la información es más cómodo.

Aquí está el código fuente del "Scriptable Object", donde se puede ver toda la información que contiene:

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

[CreateAssetMenu(fileName = "Nueva Pelicula", menuName = "Pelicula")]
public class Pelicula : ScriptableObject
{
    //Información sobre la Pelicula
    public string nombreDePelicula;
    public string urlNetflix;
    public string urltrailer;

    public float year;
    public float duracion;
    public string pais;
    public string director;
    public string guion;
    public string musica;
    public string fotografia;
    public string reparto;
    public string sinopsis;

    public float criticaAudiencia;
    public float crticaProfesional;
}
```

Una vez ya tenía el "Scriptable Object" empecé a completarlo con información, usé la ficha técnica de <https://www.filmaffinity.com> y la crítica fue extraída de <https://www.rottentomatoes.com/> ambas son dos sitios web de referencia en

cuanto a crítica y a información sobre cine. (“FilmAffinity,” n.d.), (“Rotten Tomatoes: Movies | TV Shows | Movie Trailers | Reviews - Rotten Tomatoes,” n.d.)

3.2.6 Resultado

Esta aplicación es una prueba de concepto (Figura 17), debido a que solo es funcional con 3 carteles de las películas “Star Wars los últimos Jedi”, “Sinister” y “Matrix”, ya que tenía que rellenar manualmente la información de cada película, además toda la información de la película esta alojada en el disco duro del dispositivo que ejecuta la aplicación haciendo limitada la cantidad de películas que pueden ser almacenadas y haciendo que la aplicación ocupe más de lo que debería. Para arreglar esto la información tendría que ser almacenada en servidores externos, pero por falta de recursos no ha sido posible desarrollar esa función.

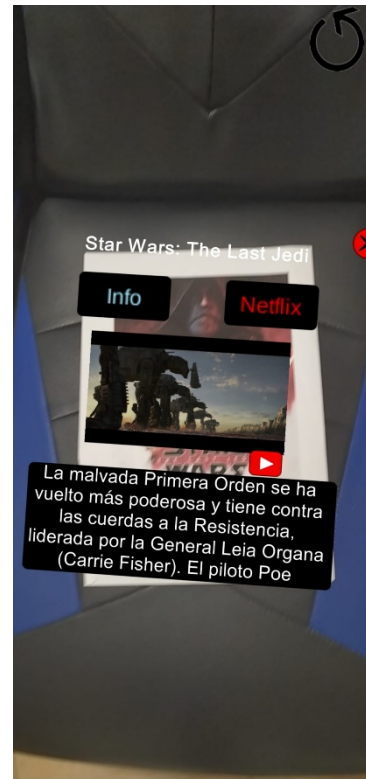


Figura 17: Resultado de la aplicación en RA.

Fuente Propia.

En conclusión, la aplicación es funcional y demuestra que es posible desarrollar una aplicación que facilite la transmisión de información de películas usando la RA.

3.3 Aplicación para representar puntos rectas y planos en RA

Para esta aplicación también use Trello en la planificación de todos los pasos para desarrollarla. Use las bases de la anterior aplicación para hacer este software compatible con teléfonos móviles y en RA.

Estos son los pasos que tenía que seguir para hacer la aplicación.

- Crear un eje de coordenadas en RA
- Situar el eje de coordenadas en el espacio RA
- Crear una UI que te permita crear elementos y asignarles ecuaciones
- Crear un sistema que cree los elementos
- Hacer estos elementos visibles en RA.
- Crear un sistema que marque la intersección de dos planos como una recta

3.3.1 Crear un eje de coordenadas en RA

Para crear un eje de coordenadas primero decidí modelar en 3D, con el programa Blender, una flecha que indicara la dirección y el sentido del eje.

(Figura 18)

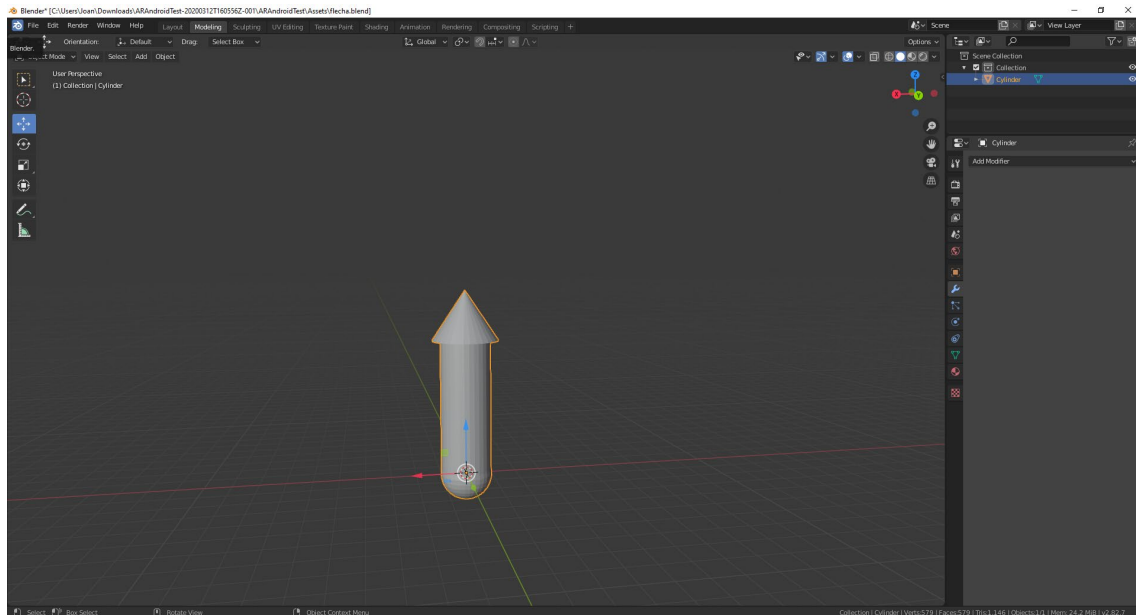


Figura 18: Captura de Blender una vez la flecha ha sido modelada.

Fuente propia.

Luego importé a Unity esta flecha y haciendo 3 copias más i coloreándolas con un color distinto creé el representante del origen de coordenadas. (Figura 19)

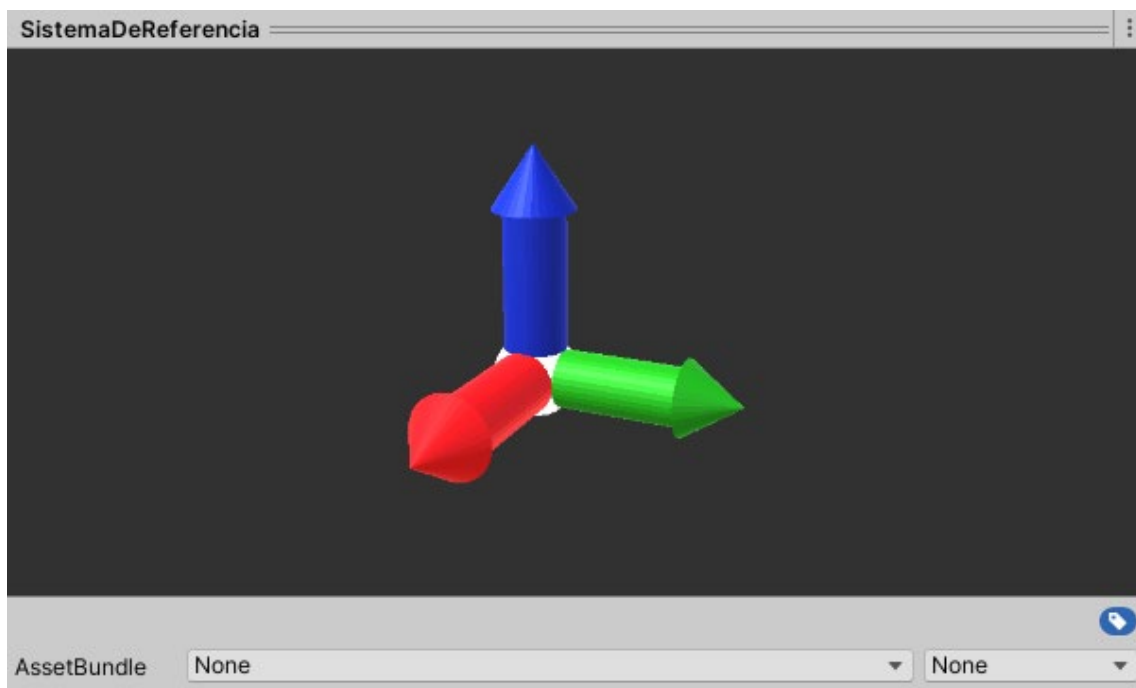


Figura 19: Representador el eje de coordenadas.

Fuente propia.

Una vez creado el origen de coordenadas creé los ejes de coordenadas para eso use una función de Unity llamada GL.line que colorea un grupo de pixeles de cierto color haciendo así una línea que uso como eje de coordenadas.

(Figura 20)

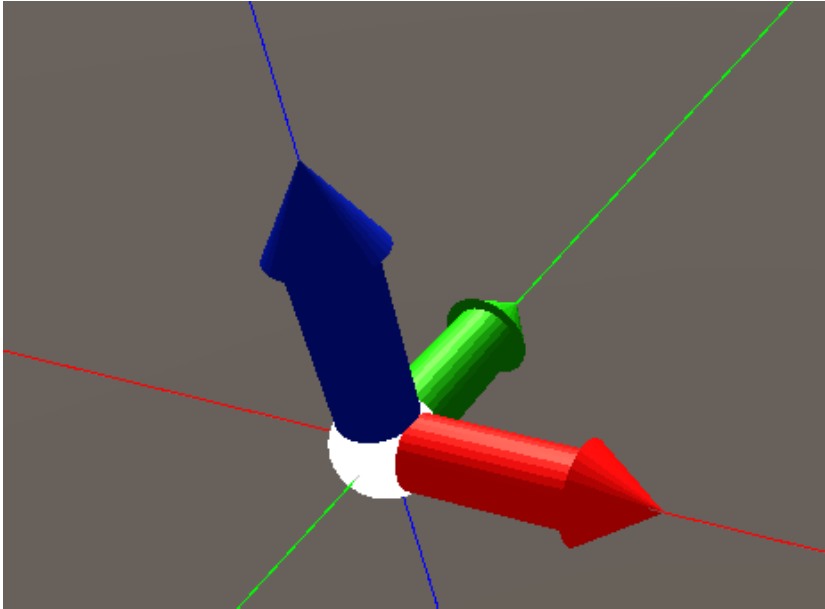


Figura 20: Eje de coordenadas con cada eje representado por una línea.

Fuente propia.

Una vez tuve el eje de coordenadas tenía que segmentarlo, para eso decidí escalar los valores cada 0,1 unidades de Unity en la dirección de cada eje, para eso cree una script de programación por procedimiento que creaba un numero cada 0,1 unidades desde el origen de coordenadas.

```

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

public class ValorsDeCordenades : MonoBehaviour
{
    [SerializeField]
    float maxNumber = 50;
    float nombreSegment;
    float perSegment;

    [SerializeField]
    GameObject numero;

    public float valor;

    ValorAlIniciar valorquetoma;

    private void Awake()
    {
        valorquetoma = numero.GetComponent<ValorAlIniciar>();
    }
    private void Update()
    {
        if (nombreSegment <= maxNumber)
        {
            Creat();
        }
    }
    void Creat()
    {
        nombreSegment += 1;
        perSegment = (nombreSegment);

        for (int i = 1; i < nombreSegment; i++)
        {
            //X
            Vector3 disxpos = new Vector3(transform.position.x + 0.1f * i, transform.position.y, transform.position.z);
            Vector3 CreatPositionxpos = transform.position + (disxpos);
            Vector3 disxneg = new Vector3(transform.position.x + 0.1f * i * -1, transform.position.y, transform.position.z);
            Vector3 CreatPositionxneg = transform.position + (disxneg);

            //Y
            Vector3 disypos = new Vector3(transform.position.x, transform.position.y + 0.1f * i, transform.position.z);
            Vector3 CreatPositionypos = transform.position + (disypos);
            Vector3 disyneg = new Vector3(transform.position.x, transform.position.y + 0.1f * i * -1, transform.position.z);
            Vector3 CreatPositionyneg = transform.position + (disyneg);

            //Z
            Vector3 diszpos = new Vector3(transform.position.x, transform.position.y, transform.position.z + 0.1f * i);
            Vector3 CreatPositionzpos = transform.position + (diszpos);
            Vector3 diszneg = new Vector3(transform.position.x, transform.position.y, transform.position.z + 0.1f * i * -1);
            Vector3 CreatPositionzneg = transform.position + (diszneg);

            valor = i;
            valorquetoma.valoress = valor;
            Instantiate(numero, CreatPositionxpos, Quaternion.identity);
            Instantiate(numero, CreatPositionxneg, Quaternion.identity);

            Instantiate(numero, CreatPositionypos, Quaternion.identity);
            Instantiate(numero, CreatPositionyneg, Quaternion.identity);

            Instantiate(numero, CreatPositionzpos, Quaternion.identity);
            Instantiate(numero, CreatPositionzneg, Quaternion.identity);
        }
    }
}

```

Una vez segmentado el sistema de referencia indiqué el nombre de los ejes con las letras X,Y,Z con sus respectivos colores. (Figura 21)

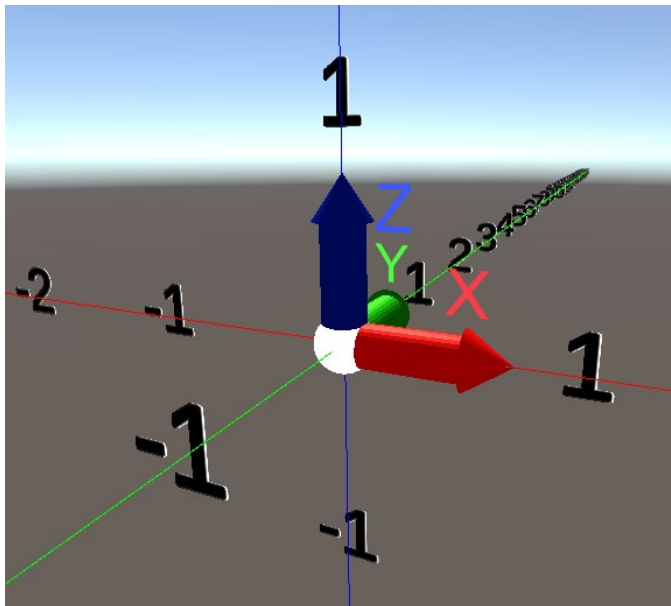


Figura 21: Eje de coordenadas segmentado.

Fuente propia.

3.3.2 Situar el eje de coordenadas en el espacio RA

Una vez creado el eje de coordenadas tenía que crear una forma para instanciarlo en un espacio RA, para ello decidí reutilizar el sistema RA que había desarrollado para la aplicación anterior, aunque tuve que modificarlo un poco. Investigando el SDK encontré una función que consistía en replicar, en un espacio digital 3D y usando planos invisibles, el entorno.

Una vez registrado el entorno instancié un indicador encima de los planos registrados para indicar donde iba a ser situado el eje de coordenadas.

3.3.3 Crear una UI que te permita crear elementos y asignarles ecuaciones

Una vez situado el eje de coordenadas tenía que crear una UI (la interfaz de usuario o UI es el conjunto de elementos de la pantalla que permiten al usuario interactuar con un software (More, 2019) capaz de crear elementos) usando el sistema de UI de Unity y usando mi teléfono para poder visualizar la aplicación. (Figura 22, 23, 24 y 25) Decidí usar las ecuaciones general y vectorial del plano y la recta para indicar matemáticamente los elementos en el espacio.

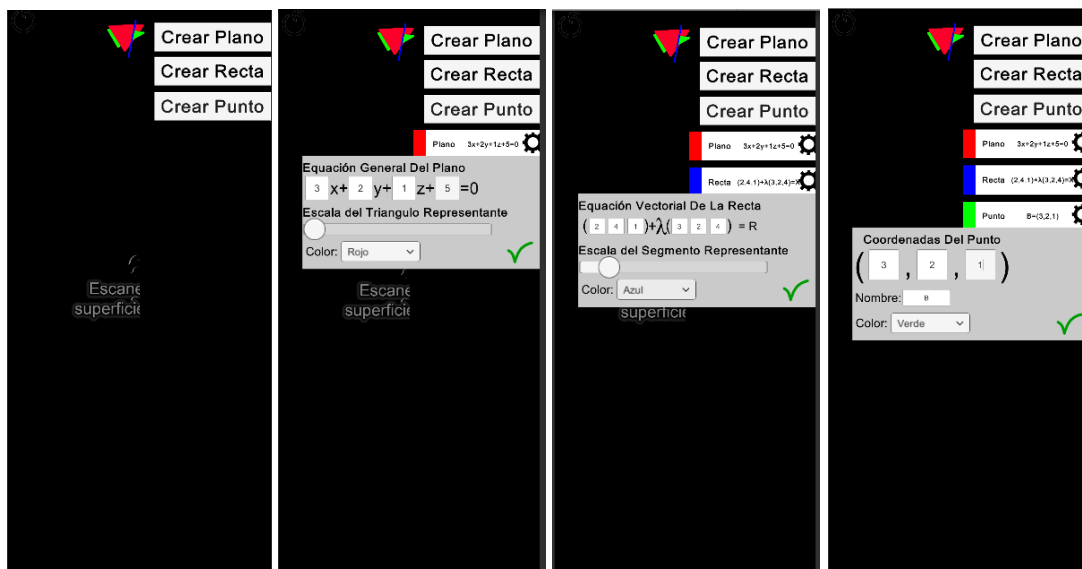


Figura 22: UI del inspector de los elementos.

Figura 23: UI para crear Planos

Figura 24: UI para crear rectas.

Figura 25: UI para crear puntos.

Fuente propia.

Fuente propia.

Fuente propia.

Fuente propia.

Una vez tuve hecha la UI capaz de crear planos, rectas y puntos tuve que crear una UI para indicar los pasos que se ha de seguir para iniciar la aplicación. Una vez estructurada la UI tuve que programar la función de cada botón, pero para eso tenía que crear los elementos.

3.3.4 Crear un sistema que cree los elementos

Para crear los elementos decidí que cada vez que se pulsa un botón para crear algo se creara un componente con alguno de los siguientes nombres “Punto”, “Plano” o “Recta”, dependiendo de que nombre tenga cada componente es activado un fragmento de código distinto (este código se encuentra en el anexo). Si el fragmento de código activado es el que corresponde con el nombre “Punto” una esfera en las coordenadas de posición indicadas con el color seleccionado y un fragmento de texto con el nombre determinado son creados. (Figura 26)

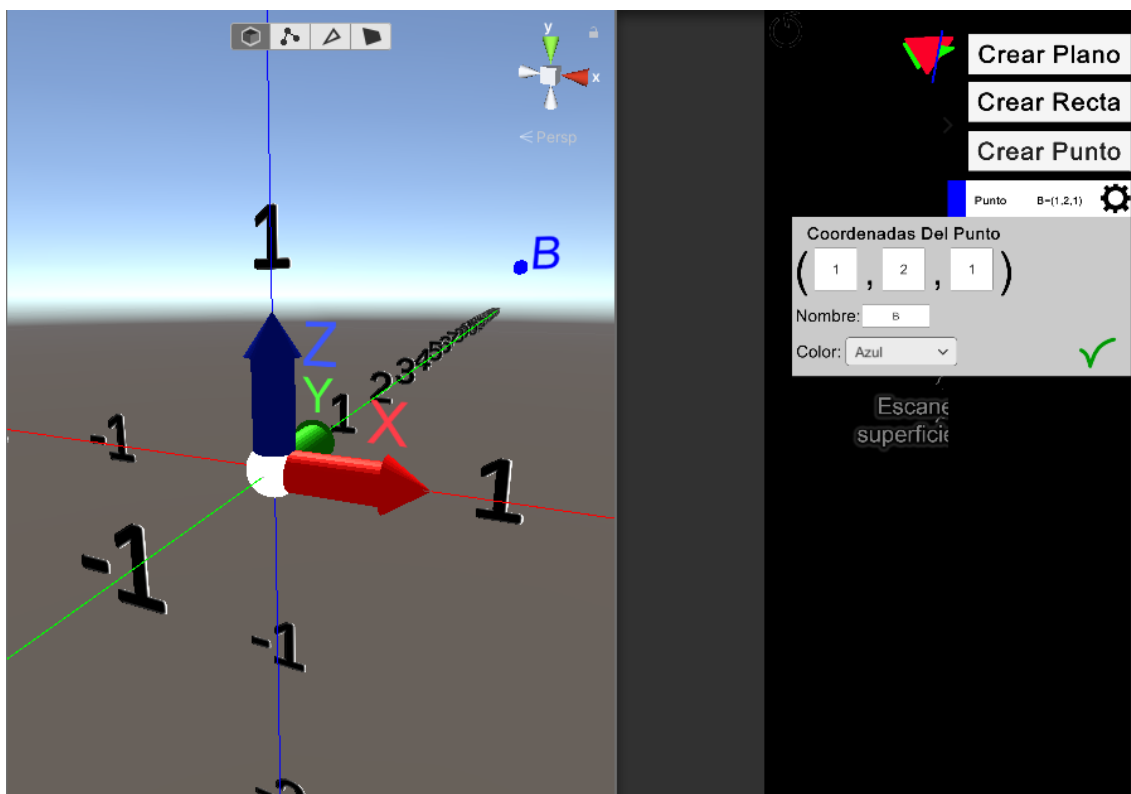


Figura 26: Captura de un punto representado en el espacio.

Fuente Propia

En cambio, si el fragmento de código activado corresponde con el nombre “Recta” uso una función de Unity llamada “Line Render” (Figura 27) que consiste en renderizar una línea de un grosor determinado entre varios puntos. Un punto es extraído de la ecuación vectorial, este punto será el origen del segmento que

representará a la recta. Los otros dos puntos los encuentro sumando el vector director a la posición del punto y multiplicando por un parámetro que en la ecuación que se muestra en el menú aparece como λ pero en el código la llamo n , este parámetro aumenta o reduce la longitud del segmento representante. Después un punto es multiplicado por 1 o -1 para hacer que escale uniformemente por ambos lados. Luego, para hacer que la recta se origine en el punto original de la recta, le resto 1 a n .

Las ecuaciones son así

$$\text{Punto 1} = (\text{Posición del punto})$$

$$\text{Punto 2} = (\text{Posición del punto}) + (\text{Vector director}) \times n \times 1$$

$$\text{Punto 3} = (\text{Posición del punto}) + (\text{Vector director}) \times (n - 1) \times -1$$

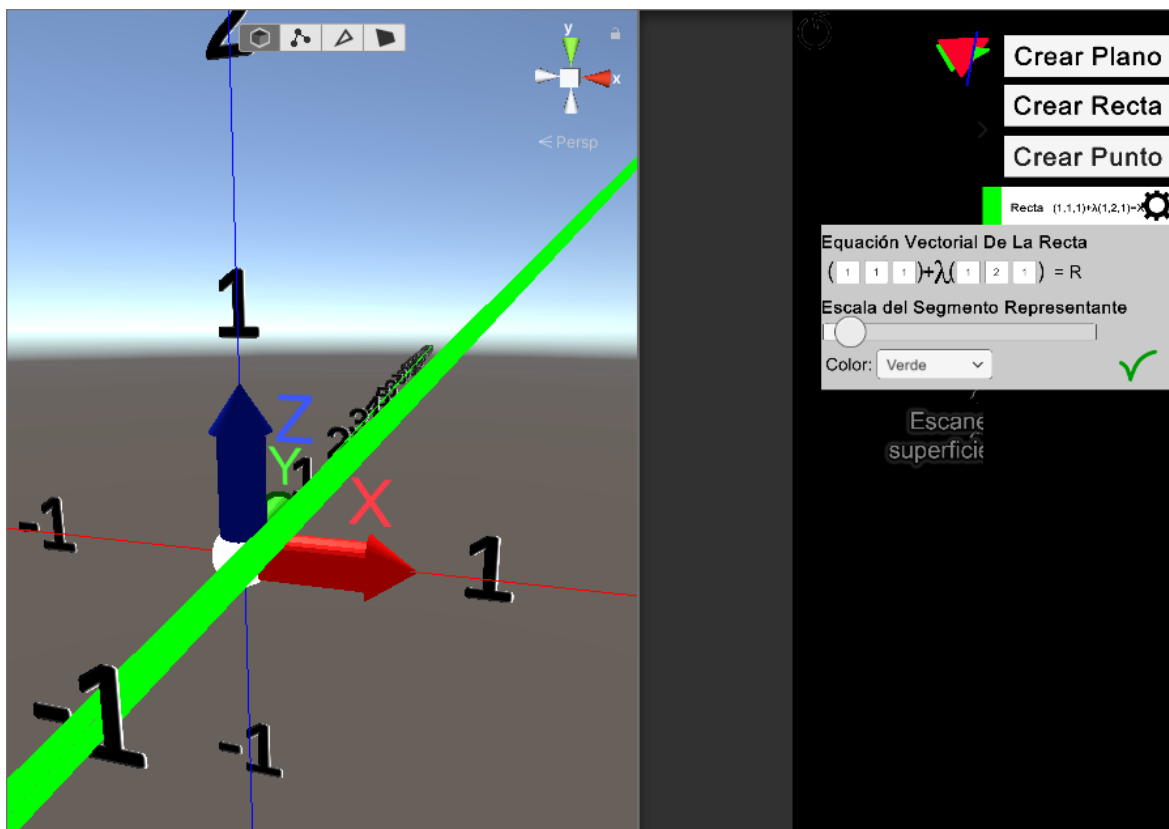


Figura 27: Captura de una recta representada en el espacio.

Figura propia.

Por último, si el fragmento de código activado es el correspondiente al nombre “Plano”, una función de Unity llamada “Mesh Renderer” usada para renderizar triángulos usando 3 puntos es activada.

Para encontrar los 3 puntos usando la ecuación general del plano y luego escalarla acorde con un parámetro que vamos a llamar e , en primer lugar, se han de hacer una serie de comprobaciones para escalar el triángulo representante de manera correcta porque el sentido de los vectores variará según si los componentes del plano son positivos o negativos debido a que un plano se ha de escalar sobre el vector que va desde el baricentro a un vértice.

(Figura 28)

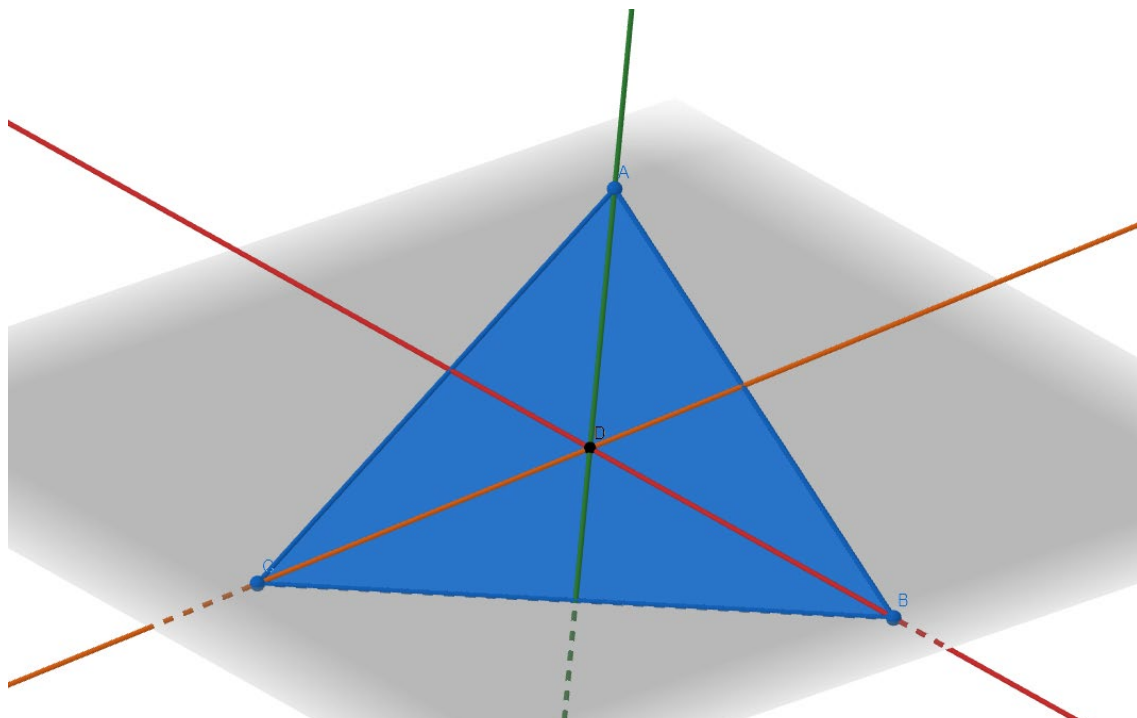


Figura 28: Representación en GeoGebra de las rectas sobre las cuales se han de desplazar los puntos para escalar el plano.

Fuente propia

Según el símbolo cada punto tendrá una ecuación un poco distinta, por ejemplo, en el caso que los componentes A, B, C tengan el mismo símbolo las ecuaciones para encontrar los punto son las siguientes:

$$\text{Punto 1} = ((-B \times e - C \times e - D) / A), e, e)$$

$$\text{Punto 2} = (e, (-A \times e - C \times e - D) / B, e)$$

$$\text{Punto 3} = (e, e, (-A \times e - B \times e - D) / C)$$

En cambio, en el caso de que el componente A sea negativo las ecuaciones son las siguientes:

$$\text{Punto 1} = ((-B \times e - C \times e - D) / A), e, e)$$

$$\text{Punto 2} = (-e, (-A \times -e - C \times e - D) / B, e)$$

$$\text{Punto 3} = (e, e, (-A \times e - B \times e - D) / C)$$

El resto de las ecuaciones se encuentran en el código fuente (ver anexo).

(Figura 29)

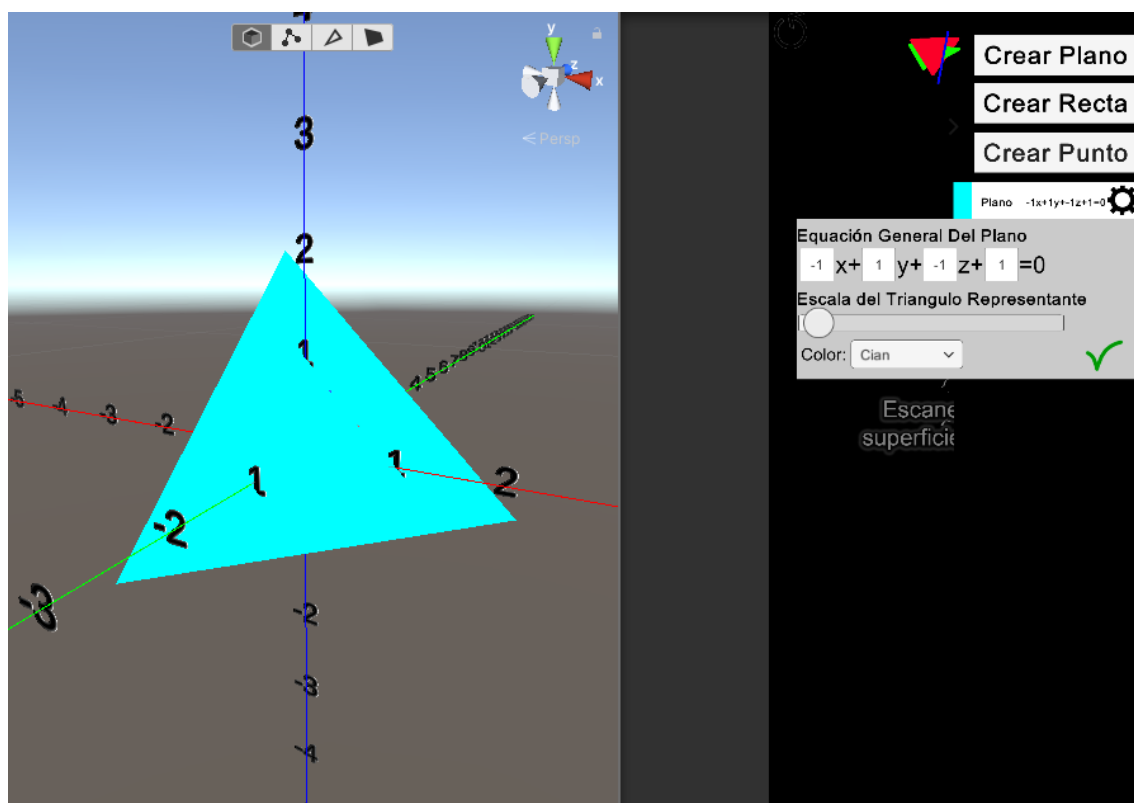


Figura 29: Captura de un plano representado en el espacio.

Fuente propia

Una vez contrasté mi aplicación con la calculadora grafica 3D de GeoGebra me di cuenta de que el motor grafico tenía el eje vertical como el eje Y en cambio,

GeoGebra tenía el eje vertical Z para arreglar esta discrepancia cambie de orden los componentes de las ecuaciones.

3.3.5 Hacer estos elementos visibles en RA.

Una vez los elementos son creados una herramienta del SDK simula la iluminación que capta de la cámara mediante una herramienta llamada “Environment Probe Manager” i simula la iluminación del espacio digital acorde con la iluminación captada por la cámara.

3.3.6 Crear un sistema que marque la intersección de dos planos como una recta en un espacio RA

Una vez diseñado el sistema de creación tanto de planos como de rectas tenía que combinar los sistemas para poder representar una recta como la intersección de dos planos. Para hacer esto tenía que seguir los siguientes pasos:

- Detectar los dos planos
- Archivar los componentes de los planos
- Crear la recta con la ecuación que represente la intersección

3.3.6.1 Detectar los dos planos

Para este paso experimente con varios métodos el primer método que intente fue detectar la yuxtaposición de dos triángulos representantes de los planos, pero me di cuenta de que eso no iba a funcionar debido a que no siempre se han de intersecar los triángulos representantes cuando los planos intersecan,

además los triángulos son elementos en 2 dimensiones y debido a eso el motor tiene problemas al detectar la intersección.

El segundo método con el cual experimente consistía en que el usuario seleccionara en el espacio los dos planos que intersecan. Pero el sistema Ray casting (hace referencia al uso de la intersección rayo-superficie para solucionar una variedad de problemas en gráficos por ordenador y geometría computacional (Roth, 1982)) que usaba para detectar que planos había seleccionado el usuario daba problemas debido a que a veces detectaba elementos erróneos, así que busque otro método.

El tercer método con el cual experimente fue el que decidí implementar en la aplicación final. Este método consiste en que el usuario después de pulsar un botón en la UI pueda seleccionar, desde el mismo inspector de los elementos, los dos planos que van a ser detectados para el cálculo de la intersección. Primero implemente la función del botón que consiste en la transformación de los elementos del inspector de elementos gráficos a elementos interaccionales, por suerte el motor de Unity y su sistema UI tiene herramientas muy prácticas para llevar a cabo esta función.

Una vez el usuario selecciona un elemento este vuelve a transformarse en un elemento gráfico, y cuando el usuario selecciona el segundo elemento todos los elementos son transformados otra vez en elementos gráficos.

3.3.6.2 Archivar los componentes de los planos

Una vez el sistema de selección de los planos estaba terminado pasé a crear un sistema que archivara los componentes de los planos para poder operar con ellos más tarde. Para llevar a cabo esto tenía que almacenar todos los componentes posibles que el usuario pueda seleccionar. Para ello use una lista que almacenaba cada componente a medida que los planos son creados y/o modificados. Una vez los componentes son almacenados, el usuario ha de seleccionar los planos, una vez los planos han sido seleccionados los componentes de cada plano son almacenados en un componente que almacena y hace los cálculos de la intersección de planos.

3.3.6.3 Crear la recta con la ecuación que represente la intersección

Una vez almacenados los componentes de los planos hacía falta organizarlos. Así que decidí nombrar los componentes de los planos con las letras 8 primeras letras del abecedario.

Las ecuaciones de los planos resultan de la siguiente forma:

$$\text{Plano 1} = Ax + By + Cz + D = 0$$

$$\text{Plano 2} = Ex + Fy + Gz + H = 0$$

Por lo tanto, la recta “**r**” intersección, se encuentra en el siguiente sistema:

$$r = \begin{cases} Ax + By + Cz + D = 0 \\ Ex + Fy + Gz + H = 0 \end{cases}$$

Aunque matemáticamente esta recta ya está expresada por completo, mi aplicación solo puede representar rectas expresadas con la ecuación vectorial.

Por lo tanto, lo primero que se ha de hacer es pasar el sistema a la ecuación

vectorial. Para hacer esto experimenté con varios métodos para resolver el sistema, como el método de Gauss o el método de reducción, pero después de investigar un poco más me di cuenta de que el método más práctico para resolver el sistema en código es el método de Cramer debido ya que permite de manera sencilla encontrar si el sistema tiene solución.

Estos son los pasos que seguí para transformar el sistema de ecuaciones a la ecuación vectorial con el método de Cramer:

Primero ordené los componentes tomando \mathbf{z} como el parámetro al cual llamaremos λ de esta forma pasa de ser un sistema con 3 incógnitas a un sistema con 2. Es necesario apuntar que, si la recta no admite que λ adquiera todos los valores reales, \mathbf{x} pasa a ser el parámetro λ y entonces todo el cálculo cambia, pero para evitar la repetición solo explicaré el proceso que seguí en el caso que \mathbf{z} sea el parámetro:

$$r = \begin{cases} Ax + By = -C\lambda - D \\ Ex + Fy = -G\lambda - H \end{cases}$$

Después convertí el sistema en matrices para poder continuar con el método de Cramer.

$$a = \begin{bmatrix} A & B \\ E & F \end{bmatrix} \quad a' = \begin{bmatrix} A & -C\lambda - D \\ E & -G\lambda - H \end{bmatrix} \quad a'' = \begin{bmatrix} -C\lambda - D & B \\ -G\lambda - H & F \end{bmatrix}$$

Después aplique el método de Cramer.

$$y = \frac{|a'|}{|a|} \quad x = \frac{|a''|}{|a|}$$

Una vez calculados los determinantes y operándose obtienen las siguientes ecuaciones equivalentes:

$$y = \frac{(D \times E) - (A \times H)}{(A \times F) - (E \times B)} + \frac{(C \times E) - (G \times A)}{(A \times F) - (E \times B)} \lambda$$

$$x = \frac{(H \times B) - (D \times F)}{(A \times F) - (E \times B)} + \frac{(G \times B) - (C \times F)}{(A \times F) - (E \times B)} \lambda$$

Por último falta extraer la ecuación vectorial de estas dos ecuaciones Para ello se usan los coeficientes de los parámetros como los componentes del vector y los componentes que lo suman se usan como los componentes del punto. (Es necesario recordar que $z = \lambda$) Por lo tanto la ecuación vectorial resulta así:

$$r = \left(\frac{(H \times B) - (D \times F)}{(A \times F) - (E \times B)}, \frac{(D \times E) - (A \times H)}{(A \times F) - (E \times B)}, 0 \right) + \lambda \left(\frac{(G \times B) - (C \times F)}{(A \times F) - (E \times B)}, \frac{(C \times E) - (G \times A)}{(A \times F) - (E \times B)}, 1 \right)$$

Finalmente, la ecuación vectorial es integrada en un elemento creando así una representación gráfica en RA de la recta interacción. (Figura 30)

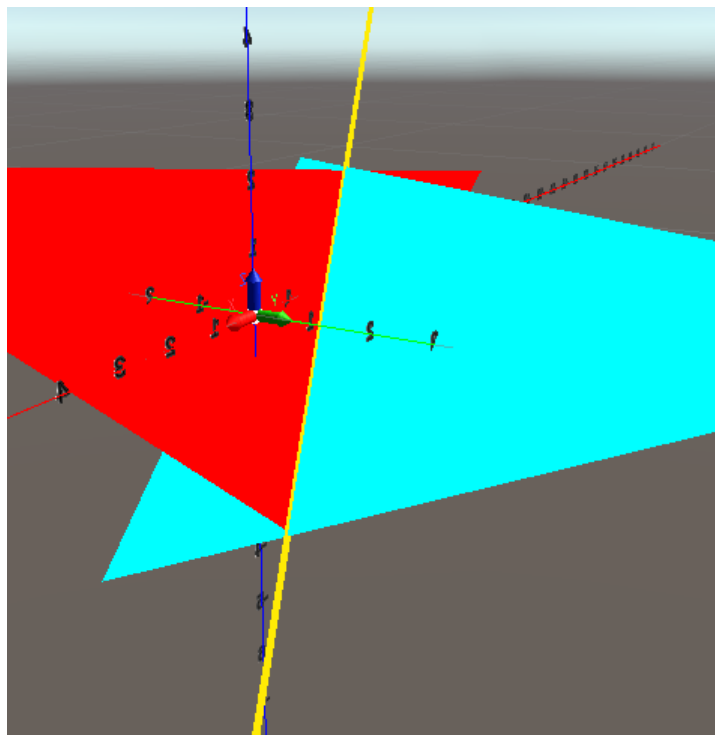


Figura 30: Captura de la representación de una intersección entre planos.

Fuente propia.

3.3.7 Resultado

A pesar de que ya existen aplicaciones RA educativas, no todas son accesibles ya sea por requerimientos de hardware o por la dificultad de uso de dichas aplicaciones. Sin embargo, esta prueba de concepto demuestra que es posible la creación de una herramienta de conocimiento accesible en RA (Figura 31). Cabe mencionar que la accesibilidad de mi aplicación podría mejorarse, pero como se ha comentado antes esta es una prueba de concepto debido a que no disponía de los suficientes recursos para desarrollar una aplicación comercial.

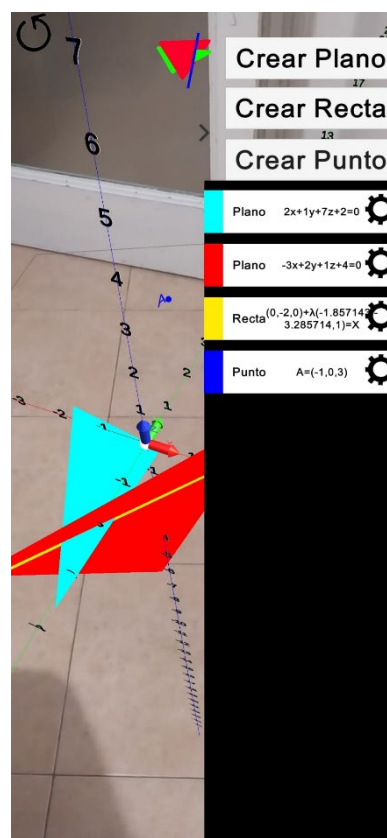


Figura 31: Resultado de la aplicación en RA.

Fuente propia.

La aplicación tiene un número importante de fallos y carece de algunas funciones vitales como la capacidad de poder suprimir elementos en el espacio, pero sin embargo esta aplicación realiza su trabajo y podría ser usada en la clase para representar elementos gráficos en el espacio. En conclusión, esta aplicación demuestra que la RA puede ser accesible y usado en un entorno como el aula.

4. CONCLUSIÓN

En conclusión, después de investigar sobre la RA se hace evidente que vez tiene mayor incidencia en nuestra sociedad y resulta beneficiosa cada vez en más disciplinas, pero también se hace evidente que la RA no consigue instaurarse como una tecnología de gran influencia a causa de su difícil accesibilidad. Tanto las empresas como el público en general son reticentes a esta tecnología debido a que presenta una barrera tanto de hardware como de accesibilidad.

Para sobrepasar estas barreras muchas desarrolladoras de software RA se han lanzado a desarrollar aplicaciones RA para teléfono móvil ya a que es la herramienta tecnológica más accesible con soporte RA que actualmente hay en el mercado. Este es el principal motivo por el cual la parte práctica se centra en este tipo de hardware.

Aunque el móvil soluciona la barrera del hardware, para el software RA el soporte del teléfono móvil ha de mejorar en muchos aspectos. Es necesario mencionar que las empresas de telefonía están trabajando para hacer del teléfono móvil un mejor soporte RA, pero aún queda muchas funciones que añadir y optimizar por lo que la RA aún no ha incidido con éxito en el mercado actual.

En contraste, actualmente el futuro de la RA se presenta muy brillante debido al avance de las tecnologías sobre las cuales se soporta. Este avance implica que la RA será cada vez más cómoda, robusta y popular ya que este progreso tecnológico derribará las barreras que actualmente impiden que la RA se consolide entre el público en general.

En cuanto a los objetivos que me he planteado en este trabajo he podido cumplir con mayor o menor éxito cada uno de ellos.

En primer lugar, he encontrado aplicaciones más o menos útiles a nuestro día a día, tal como muestra la aplicación de representación de películas y la aplicación de representación de elementos gráficos en el espacio.

En segundo lugar, he conseguido dominar los conceptos necesarios para entender y manejar la RA. Además, he sido capaz de adquirir los conocimientos necesarios para crear una aplicación para teléfono móvil.

Por último las dos aplicaciones desarrolladas demuestran que es posible la creación de herramientas RA accesibles y con aplicación en campos cotidianos como la búsqueda de información y la educación, aunque, estas dos aplicaciones, aún muestran algunos fallos a causa de la falta de recursos y de desarrollo de tecnologías en las cuales se basa la RA. Pese a estos fallos he sido capaz de llevar a cabo estas aplicaciones como prueba de concepto.

Recapitulando, este trabajo de investigación me ha llevado a confirmar la hipótesis inicial siendo posible crear herramientas accesibles de RA aunque faltaría desarrollar la tecnología para que dichas herramientas influyeran de forma importante nuestro día a día.

5. BIBLIOGRAFÍA

- (1) *MIT Explains: How Does Virtual Reality Work? - YouTube*. (n.d.). Retrieved from https://www.youtube.com/watch?time_continue=48&v=-Kovxf6g0mo&feature=emb_logo
- (38) *DHL tests augmented reality in warehouse order-picking - YouTube*. (n.d.). Retrieved July 4, 2020, from <https://www.youtube.com/watch?v=HKhhJxlbiuM>
- Alamparambil, J., & Mowrer, T. (n.d.). *Unity at GDC - First look at Unity 2018 mobile AR key features - YouTube*. Retrieved from <https://www.youtube.com/watch?v=3WvgLECD73c>
- Anamorphic Now*. (n.d.).
- Apple, I. (2019). ARKit | Apple Developer Documentation. Retrieved November 22, 2019, from Apple Inc. website: <https://developer.apple.com/documentation/arkit/>
- Behringer, R., Tam, C., McGee, J., Sundareswaran, S., & Vassiliou, M. (2000). A wearable augmented reality testbed for navigation and control, built solely with commercial-off-the-shelf (COTS) hardware. *Proceedings - IEEE and ACM International Symposium on Augmented Reality, ISAR 2000*, 12–19. <https://doi.org/10.1109/ISAR.2000.880918>
- Casebook, A. (2014). *Digital theatre*. Retrieved from https://en.wikipedia.org/w/index.php?title=Digital_theatre&oldid=638045784
- Cavas-Martínez, F., Eynard, B., Fernández Cañavate, F. J., Fernández-

Pacheco, D. G., Morer, P., & Nigrelli, V. (n.d.). *Advances on mechanics, design engineering and manufacturing II: proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2018)*.

CCNA Data Center DCICT 640-916 Official Cert Guide - Navaid Shamsee, David Klebanov, Hesham Fayed, Ahmed Afrose, Ozden Karakok - Google Llibres. (n.d.). Retrieved July 19, 2020, from https://books.google.es/books?id=0GkBgAAQBAJ&pg=PA934&redir_esc=y#v=onepage&q&f=false

Cui, N., Kharel, P., & Gruev, V. (2017). Augmented reality with Microsoft HoloLens holograms for near infrared fluorescence based image guided surgery. *Molecular-Guided Surgery: Molecules, Devices, and Applications III*, 10049, 1004901. <https://doi.org/10.1117/12.2251625>

Delgado, F. J., Abernathy, M. F., White, J., & Lowrey, W. H. (1999). <title>Real-time 3D flight guidance with terrain for the X-38</title> In J. G. Verly (Ed.), *Enhanced and Synthetic Vision 1999* (Vol. 3691, pp. 149–156). <https://doi.org/10.1117/12.354416>

ECMA-334 5 th Edition / C# Language Specification. (2017). Retrieved from www.ecma-international.org

FilmAffinity. (n.d.). Retrieved August 15, 2020, from <https://www.filmaffinity.com/es/main.html>

GeoGebra AR – GeoGebra. (n.d.). Retrieved August 13, 2020, from <https://www.geogebra.org/m/kcwfax5a>

Heilig, M. L. (1962). *U.S. Patent No. 3,050,870. 870.*

How Museums are using Augmented Reality - MuseumNext. (n.d.). Retrieved April 18, 2020, from <https://www.museumnext.com/article/how-museums-are-using-augmented-reality/>

IBM, & Feigenblatt, R. O. (2005). *Absolute Display Window Mouse / Mice An IP . com Prior Art Database Technical Disclosure Authors et . al .: IBM Feigenblatt , RI Original Publication Date : March 01 , 1987 Original Disclosure Information : Research Disclosure n275 03-87 IP . com Number :*

L. Frank Baum. (n.d.). "The Master Key": L. Frank Baum envisions augmented reality glasses in 1901 | Mote & Beam by Joel Johnson. Retrieved February 14, 2020, from <https://web.archive.org/web/20130522153011/http://moteandbeam.net/the-master-key-l-frank-baum-envisions-ar-glasses-in-1901>

Lee, G., & Billingham, M. (2012). *CityViewAR outdoor AR visualization. 97.* <https://doi.org/10.1145/2379256.2379281>

Lintern, G. (1980). Transfer of landing skill after training with supplementary visual cues. *Human Factors*, 22(1), 81–88. <https://doi.org/10.1177/001872088002200109>

Lintern, G., Roscoe, S. N., & Sivier, J. E. (1990). Display principles, control dynamics, and environmental factors in pilot training and transfer. *Human Factors*, 32(3), 299–317. <https://doi.org/10.1177/001872089003200304>

Mann, S. (1996). Wearable Tetherless Computer-Mediated Reality: WearCam

- as a wearable face-recognizer, and other applications for the disabled.
IEEE Transactions on Reliability, (February).
- Milgram, P., & Kishino, F. (1994). Taxonomy of mixed reality visual displays.
IEICE Transactions on Information and Systems, E77-D(12), 1321–1329.
- Miller, R. W. (2008). WOLF — A computer expert system for sunspot
classification and solar flare prediction. In *Knowledge-Based Systems in
Astronomy* (Vol. 83, pp. 107–120). https://doi.org/10.1007/3-540-51044-3_20
- More, M. (2019). ¿Qué es la UX y la UI? *Thinking for Innovation*. Retrieved from
<https://www.iebschool.com/blog/que-es-la-ux-y-la-ui-analitica-usabilidad/>
- Rosenber, L. (1992). *The use of virtual fixtures as perceptual overlays to
enhance operator performancew in remote enviroments*. (p. 50). p. 50.
Retrieved from <http://www.dtic.mil/dtic/tr/fulltext/u2/a292450.pdf>
- Rosenberg, L. B. (1992). *The Use of Virtual Fixtures as Perceptual Overlays to
Enhance Operator Performance in Remote Environments*. Retrieved from
<https://apps.dtic.mil/docs/citations/ADA292450>
- Roth, S. D. (1982). Ray casting for modeling solids. *Computer Graphics and
Image Processing*, 18(2), 109–144. [https://doi.org/10.1016/0146-664X\(82\)90169-1](https://doi.org/10.1016/0146-664X(82)90169-1)
- Rotten Tomatoes: Movies | TV Shows | Movie Trailers | Reviews - Rotten
Tomatoes. (n.d.). Retrieved August 15, 2020, from
<https://www.rottentomatoes.com/>
- Schueffel, P. (2017). *The Concise Fintech Compendium*. Retrieved from

https://web.archive.org/web/20180425130029/http://www.heg-fr.ch/FR/HEG-FR/Communication-et-evenements/evenements/Documents/Schueffel2017_The-Concise-FINTECH-COMPENDIUM.PDF

Steure, J. (1993). Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42(4), 73–93.

Sutherland, I. E. (1968). *A head-mounted three dimensional display**. 5–10.

Tractica Omdia. (n.d.). Installed Base of Mobile Augmented Reality Apps to Reach 2.2 Billion by 2019 | Omdia | Tractica. Retrieved July 1, 2020, from <https://tractica.omdia.com/newsroom/press-releases/installed-base-of-mobile-augmented-reality-apps-to-reach-2-2-billion-by-2019/>

What is Build? - Definition from Techopedia. (n.d.). Retrieved July 19, 2020, from <https://www.techopedia.com/definition/3759/build>

Yen, J.-C., Tsai, C.-H., & Wu, M. (2013). Augmented Reality in the Higher Education: Students' Science Concept Learning and Academic Achievement in Astronomy. *Procedia - Social and Behavioral Sciences*. <https://doi.org/10.1016/j.sbspro.2013.10.322>