

# ENTORNOS VIRTUALES

## Ejercicio 5. Aplicación de realidad aumentada.

Lidia Sánchez Mérida

De entre todas las aplicaciones de realidad aumentada existentes actualmente, las que me parecen más útiles son aquellas que te permiten comprobar cómo quedan, estéticamente, algunos objetos en tu entorno. En particular, esta es la idea que se esconde tras la aplicación [\*IKEA Place\*](#), que brinda a los usuarios la posibilidad de utilizar su dispositivo móvil para visualizar el aspecto de sus habitaciones integrando sus productos en 3D. Para ello la aplicación cuenta con un catálogo de todo tipo de muebles que poder seleccionar, posicionar y probar con la realidad aumentada, además de la posibilidad de cambiarles el color y aplicarles transformaciones, como por ejemplo rotaciones. El posicionamiento de los productos tiene en cuenta las dimensiones de la habitación, lo cual ayuda a visualizarlos a escala real de manera que el usuario pueda conocer si un determinado mueble encaja en un espacio concreto [1].

Entre las distintas tecnologías utilizadas para desarrollar esta aplicación se encuentra la herramienta de diseño gráfico *Unity3D* con la que modelar los productos de la empresa, mientras que para la realidad aumentada han utilizado el framework **ARKit** [3] desarrollado por Apple. Al ser *open source* cualquier desarrollador que desee construir una aplicación de realidad aumentada para dispositivos iOS puede utilizarlo [2].

Para integrar los objetos virtuales con la escena real, este software utiliza la técnica *Visual-Inertial Odometry* (VIO) [2]. Esta incluye métodos de visión por computador para recopilar y analizar información acerca de la escena en combinación con los sensores de movimiento del dispositivo [4]. Para acceder a sus valores, se puede utilizar otro framework denominado *Core Motion*, que procesa y almacena los valores recopilados por acelerómetros, giroscopios, etc [5]. Así, ARKit es capaz de identificar las características de la escena real para detectar superficies planas tanto verticales como horizontales, como por ejemplo suelos, techos, paredes, ventanas, puertas, mesas, etc [4]. El inconveniente de realizar una detección de características utilizando los sensores de movimiento reside en que solo se podrá calcular la distancia entre dos puntos si estos presentan altos contrastes. Lo cual no siempre sucede cuando por ejemplo la escena tiene poca iluminación, las superficies son blancas o el usuario mueve demasiado la cámara [7].

Una vez la escena real ha sido analizada, ARKit por defecto coloca los objetos virtuales en el centro de la pantalla sobre una de las superficies detectadas. Para ello utiliza una técnica denominada *Ray Casting* con la que realiza una correspondencia entre el espacio físico de la escena real y el que aparece en la pantalla para colocar el producto. Entre los distintos atributos que calcula se encuentra la profundidad y la orientación de la superficie. Además,

este framework calcula una estimación acerca de las medidas de la escena, con la que como se ha comentado anteriormente, la aplicación de IKEA puede decirle al usuario si el objeto elegido encaja o no en el espacio visualizado [6]. De igual modo, también proporciona un sistema de eventos para realizar las transformaciones correspondientes a los gestos que realiza el usuario, como mover el objeto, rotarlo, etc [5].

## Valoración personal

En función de la información que he estado leyendo acerca de *ARKit* puedo decir que me ha sorprendido bastante el hecho de que haya sido desarrollado por *Apple* y sea *open source*, puesto que esta compañía precisamente destaca por lo contrario, es decir, desarrollar bibliotecas y herramientas privativas con las que los desarrolladores construyen sus aplicaciones a cambio de pagar por su uso.

Asimismo, he estado investigando acerca de las diferencias existentes con otro de los framework para realidad aumentada más populares: el **ARCore** de Google. Ambos utilizan técnicas similares tanto para el análisis del entorno real como para el posicionamiento y la interacción con los objetos virtuales. Sin embargo, varias fuentes coinciden en que *ARCore* es capaz de identificar un mayor número de características utilizando una mayor cantidad de datos, mientras que *ARKit* solo almacena los más recientes capturados por la cámara, lo que provoca una menor precisión a la hora de mapear la escena. Por contra el framework de Apple es capaz de detectar superficies horizontales y verticales con una mayor exactitud que el de Google [8].

Tras las últimas actualizaciones de *ARKit*, como podemos visualizar en la siguiente tabla, Apple ha incluido más técnicas que *ARCore* aún no dispone, como capturar el movimiento, el uso dual de ambas cámaras y el posicionamiento de personas.

Features	ARCore	ARKit
Motion tracking	✓	✓
Environmental understanding	✓	✓
Environmental light estimation	✓	✓
Collaborative Sessions	✓	✓
People Occlusion	✓	✓
Motion Capture	✗	✓
Simultaneous Front and Back Camera	✗	✓
Multiple Face Tracking	✗	✓
Cross-platform experiences across both Android and iOS	✓	✗
Tracking an extended number of feature points	✓	✗

Asimismo, en esta otra fuente aparece que *ARKit* es más utilizado por los desarrolladores en términos de preguntas relacionadas y repositorios en *GitHub* [9], lo que da que pensar que se trata de un *framework* de código libre bastante completo para desarrollar aplicaciones de realidad aumentada para dispositivos iOS. Además, en este mismo estudio explica que el mercado de Apple es más estable debido a la poca variación *hardware* que presentan los diferentes modelos, mientras que dependiendo de la gama a la que pertenezca un dispositivo Android, sus componentes pueden ser muy distintos. De este modo, si utilizas *ARKit* para desarrollar una aplicación AR te puedes asegurar que más gente pueda consumirla aunque no dispongan de teléfonos recientes. De una forma similar ocurre con las versiones de los sistemas operativos, requiriendo para *ARCore* Android 7.0 o superior [10]. Sin embargo, conforme más novedoso es el dispositivo mejores técnicas incorpora como es el caso de los *iPad Pro*, que integran el sistema *LiDAR (Light Detection and Ranging)*, que genera mapas de las escenas 3D más precisos, lo cual ayuda a que la interacción con los objetos virtuales sea más realista [11].

En resumen, me parece que *ARKit* es un *framework* muy completo para el desarrollo de aplicaciones AR, dispone de una comunidad extensa que cada día parece aumentar y multitud de códigos de ejemplo para realizar cada una de las funcionalidades que proporciona. Asimismo, el hecho de que sea *open source* posibilita que más desarrolladores puedan crear sus proyectos y contribuir a la mejora de esta herramienta. De igual modo, en referencia a los dispositivos a los que va dedicado, creo que *ARKit* brinda una gran oportunidad para comercializar tus aplicaciones dentro de un mercado más extenso debido a que podrán ser utilizadas por la mayoría de usuarios iOS, lo cual representa una gran ventaja en comparación por ejemplo con el de Google.

## Bibliografía

1. MINDSTUDIOS, Svitlana Varaksina, *AR Apps Boost Business: How to Make an App Like IKEA Place*, 2019, <https://themindstudios.com/blog/how-to-make-an-app-like-ikea-place/>
2. <sup>TM</sup>, Nadia Kovach, *How much does it cost to create an app like IKEA*, <https://thinkmobiles.com/blog/how-much-cost-make-app-like-ikea/>
3. Apple Developers, *ARKit*, <https://developer.apple.com/documentation/arkit/>
4. Apple Developers, *Understanding World Tracking*, [https://developer.apple.com/documentation/arkit/world\\_tracking/understanding\\_world\\_tracking](https://developer.apple.com/documentation/arkit/world_tracking/understanding_world_tracking)
5. Apple Developers, *Core Motion*, <https://developer.apple.com/documentation/coremotion>
6. Apple Developers, *Placing Objects and Handling 3D Interaction*, [https://developer.apple.com/documentation/arkit/world\\_tracking/placing\\_objects\\_and\\_handling\\_3d\\_interaction](https://developer.apple.com/documentation/arkit/world_tracking/placing_objects_and_handling_3d_interaction)
7. Developers, Osama Abdelkarim Aboulhassan, *iOS ARKit Tutorial: Drawing in the Air with Bare Fingers*, <https://www.toptal.com/swift/ios-arkit-tutorial-drawing-in-air-with-fingers>

8. Inflexion Blog, Tatyana Shavel, *ARCore vs. ARKit: Which Is Better for Building Augmented Reality Apps?*, 2019, <https://www.inflexion.com/blog/arcore-vs-arkit>
9. exyte, Daniil Manin, *Augmented reality software: ARKit vs. ARCore in 2020*, <https://exyte.com/blog/augmented-reality-software>
10. Vieyra Software, *Comparing Google ARCore and Apple ARKit*, 2018, <https://medium.com/@vieyrasoftware/comparing-google-arcore-and-apple-arkit-81b4727132ad>
11. Medium, Vardhan Agrawal, *How ARKit 3.5 Enables Immersive Augmented Reality Experiences on iOS*, <https://heartbeat.fritz.ai/how-arkit-3-5-enables-immersive-augmented-reality-experiences-on-ios-50d45ae352ef>