



UNIVERSIDAD DE JAÉN
Escuela Politécnica de Jaén

Trabajo Fin de Grado

ESTUDIO DE LAS POSIBILIDADES DE ARCORE PARA LA INTEGRACIÓN DE LA REALIDAD AUMENTADA EN SITIOS ARQUEOLÓGICOS

Alumno: José Antonio Molina Real

Tutor: Prof. D. Antonio Jesús Rueda Ruiz

Dpto: Departamento de informática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Antonio Jesús Rueda Ruíz , tutor del Proyecto Fin de Carrera titulado: Estudio de las posibilidades de ARCore para la integración de la realidad aumentada en sitios arqueológicos, que presenta José Antonio Molina Real, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2019

El alumno:

Los tutores:

José Antonio Molina Real

Antonio Jesús Rueda Ruiz

Índice

Escuela Politécnica Superior de Jaén.....	1
Resumen.....	5
Abstract.....	5
1. Diseño Metodológico.....	7
1.1. Objetivos y formulación del problema.....	7
1.1.1. Descripción del problema.....	7
1.1.2. Formulación del problema.....	7
1.2. Justificación del problema.....	7
1.3. Objetivos.....	8
2. Realidad aumentada.....	8
2.1. Historia.....	8
2.2. Definición.....	11
2.3. Características y componentes.....	12
2.3.1. Características.....	12
2.3.2. Niveles.....	13
2.3.3. Componentes.....	14
2.3.4. Tipos de display.....	15
2.4. Software.....	17
2.4.1. Plataformas de desarrollo de realidad virtual.....	17
2.5. Aplicaciones.....	21
3. Realidad Aumentada y arqueología.....	25
3.1. Introducción.....	25
3.2. Reconstrucción yacimiento arqueológico.....	26
3.2.1. Paso en la reconstrucción de un yacimiento arqueológico.....	26
3.3. Musicalización de yacimientos arqueológicos.....	29
3.4. Histórico de aplicaciones.....	31
4. Diseño de aplicación de realidad aumentada: Google ARCore.....	37
4.1. Introducción.....	37
4.2. Historia.....	38
4.3. Funcionamiento Google ARCore.....	39
4.3.1. Seguimiento del movimiento.....	39
4.3.2. Comprensión ambiental.....	41

4.3.3. Estimación de luz	41
4.4. APLICACIONES ARCORE.....	42
4.5. Otras tecnologías	44
4.5.1. ARKit.....	44
4.5.2. Vuforia	46
4.6. Comparativa ARCore Vs ARKit Vs Vuforia.....	47
4.6.1. ARCORE VS ARKIT.....	47
4.6.2. Vuforia vs ARKit/ARCore	48
5. Aplicación de reconstrucción de yacimientos arqueológicos utilizando ARCore	51
5.1. Objetivos de la aplicación.....	51
5.2. Fases de realización	51
5.3. Análisis.....	52
5.3.1. Catálogo de Requisitos.....	52
5.4. Metodología de desarrollo	56
5.4.1. Contenido de cada prototipo	56
5.5. Diseño.....	57
5.5.1. Tecnologías usadas	57
5.5.2. Diseño de pantallas.....	58
5.5.3. Diseño de la interfaz	60
5.6 Implementación.....	65
5.7. Uso de la aplicación	68
5.8. Pruebas.....	69
5.8.1. Pruebas a la tecnología de ARCore	69
5.8.2. Pruebas en la aplicación desarrollada.....	81
5.9. YACIMIENTO RECONSTRUIDO	91
5.9.1. Historia	91
5.9.2. Edificios	93
5.8.3. Yacimiento reconstruido	99
6. Conclusiones.....	104
7. Trabajo futuro.....	106
8. Bibliografía	106
ANEXO A: Manual de usuario	108

Resumen

En este Trabajo Final de Grado (TFG) se estudia la tecnología Google ARCore y la posibilidad de adaptarla para la creación de una aplicación móvil para la reconstrucción de yacimientos arqueológicos mediante realidad aumentada.

La funcionalidad consiste en que, con el uso de esta app, un usuario pueda realizar un paseo por el yacimiento reconstruido y, con la ayuda de su dispositivo móvil, sea capaz de visualizar los edificios como eran en la antigüedad. Por otro lado, un usuario con permisos de desarrollador, con la ayuda de la app, pueda colocar los diferentes edificios del yacimiento en su lugar correspondiente y ajustarlos en escala, rotación y altura.

Para el desarrollo de esta App se ha utilizado la plataforma Unity y el modelado de los objetos 3D se ha realizado en Blender.

Se ha contado con la ayuda del arqueólogo de la ciudad de Úbeda para la obtención de materiales como planos, fotografías de los edificios e historia de los mismos.

Abstract

In this Final Degree Project (TFG) we study the Google ARCore technology and the possibility of adapting it for the creation of a mobile application for the reconstruction of archeological sites.

Using this app a user can take a walk through the reconstructed site and, with the help of their mobile device, be able to visualize the buildings as they were in ancient times. On the other hand, a user with developer permissions, can place different buildings in their corresponding place and adjust them in scale, rotation and height.

For the development I used Unity platform and the modeling o 3D object has been performed in Blender.

I of was supportd by the archaeologist of the city of Ubeda o obtain materiales such as city plans, photographs of the buildings and information about their history.

1. Diseño Metodológico

En este capítulo se detallará el diseño metodológico del proyecto, características del mismo como pueden ser descripción del problema tratado, los objetivos del proyecto y algunas más necesarias para el desarrollo del mismo.

1.1. Objetivos y formulación del problema

1.1.1. Descripción del problema

El proyecto irá enfocado a la investigación del panorama de la realidad aumentada aplicada a la arqueología.

Para ello será necesario un rápido repaso de algunos conceptos básicos de realidad aumentada y el estudio de tecnologías anteriores aplicadas al caso que nos ocupa.

1.1.2. Formulación del problema

¿Es posible diseñar una aplicación móvil que nos permita la visualización, en realidad aumentada, de yacimientos arqueológicos con una precisión de localización mayor a la obtenida por GPS?

1.2. Justificación del problema

A día de hoy la realidad aumentada se está convirtiendo en una gran herramienta compatible con muchos campos como son la medicina, la educación, etc.

Una de sus aplicaciones es la reconstrucción de yacimientos arqueológicos. Actualmente existen aplicaciones dedicadas a esto. Estas aplicaciones presentan algunos inconvenientes como puede ser el elevado coste de desarrollo o la portabilidad

Con este proyecto se investigarán nuevas tecnologías de realidad aumentada capaces de solucionar los problemas de las aplicaciones de reconstrucción de yacimientos arqueológicos de hoy día.

1.3. Objetivos

Realizar un estudio de la tecnología ARCore y evaluar su aplicación a la reconstrucción de yacimientos arqueológicos

2. Realidad aumentada

2.1. Historia

1950

-MortonHeiling escribe sobre “cine de experiencia” donde surge el concepto de integrar todos los sentidos del espectador con la actividad en la pantalla.

1962

-Construcción del prototipo “Sensorama”, el cual aumentaba la experiencia del usuario ofreciéndole no solo estímulos visuales, sino ,estímulos olfativos, táctiles y auditivos.

1968

-Ivan Sutherland y Bob Sproull construyen el primer “Head Mounted Display”(HMD). La tecnología aplicada era muy primitiva a términos de interfaz de usuario y realismo.

1985

-MyronKrueger crea “Videoplace” donde el usuario podía, por primera vez, interactuar con objetos virtuales. Podemos observar el sistema en la ilustración 1.

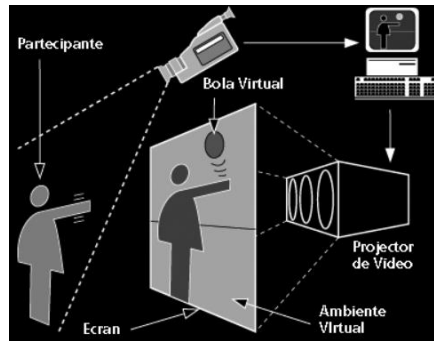


Ilustración 1

1992

Tom Caudell introduce el término “Realidad aumentada”.

1994

Steven Feiner, Blair MacIntyre y Doree Seligmann presentan el prototipo KARMA en la conferencia de la interfaz gráfica.

KARMA (Knowledge-based Augmented Reality for Maintenance Asistance). Prototipo cuyo objetivo era desarrollar gráficos 3D proyectados de una imagen fantasma para mostrar cómo arreglar una impresora. Utilizando para ello un HMD y un rastreador. Podemos observarlo en la ilustración 2.



Ilustración 2

1995

GunpeiYokoi presenta la Nintendo Virtual Boy. Ilustracion 3



Ilustración 3

1999

HirokazuKato desarrolla ARToolKit en el HitLab y se presenta en SIGGRAPH ese año.

2000

Investigadores de la universidad de Sudamérica presentan una extensión de Quake que permitía jugar en primera persona en escenarios reales. Juego representado en la ilustración 4



Ilustración 4

2001

Se presenta Archeoguide. Un sistema que ofrece información personalizada basada en el contexto y muestra reconstrucciones de edificios.

2003

Siemens lanza “Mozzies”, el primer juego de realidad aumentada para teléfonos móviles.

2007

Klein y Murray presentan el algoritmo PTAM, una adaptación del SLAM que separa el tracking y el mapping.

2013

Sony se incorpora al mundo de la realidad aumentada con el juego “ThePlayroom” para PS4.

2016

Niantic saca Pokémon GO, un juego de realidad aumentada cuyos éxitos no tienen precedentes en el genero.

2.2. Definición

Según la taxonomía descrita por Milgram y Kishino, los entornos de Realidad Mixta son aquellos en los que “*se presentan objetos del mundo real y objetos virtuales de forma conjunta en una única pantalla*”. Con esto podemos crear una definición para cada una de las aplicaciones que tiene la realidad aumentada.

Podemos apreciar la combinación de ambas tecnologías en la ilustración 5.

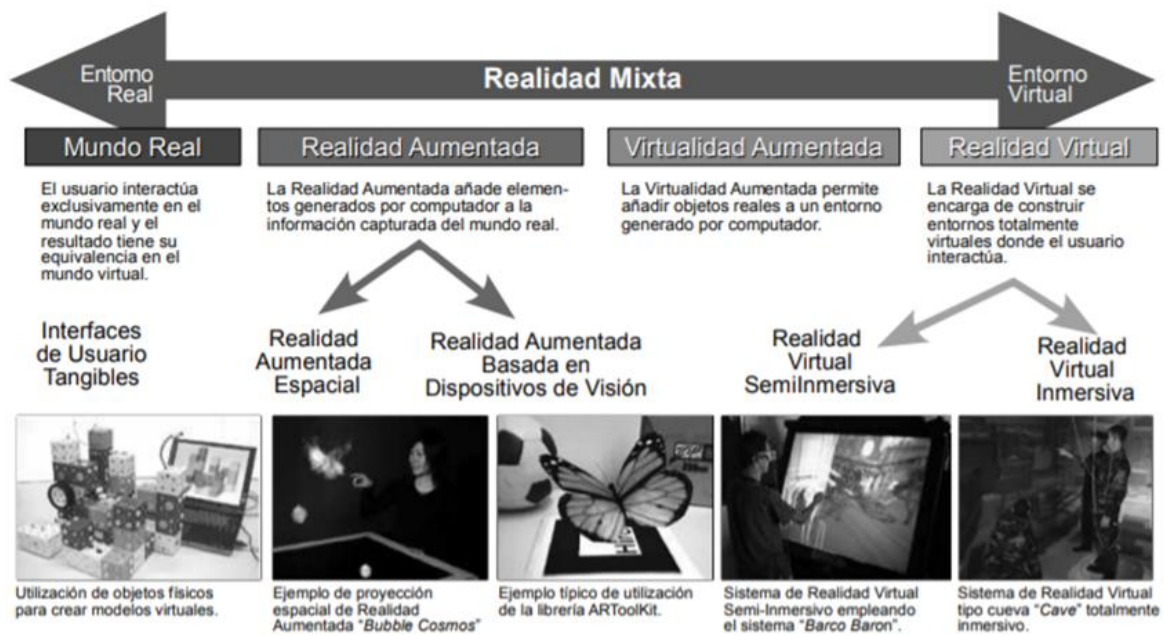


Ilustración 5

2.3. Características y componentes

2.3.1. Características

Un sistema de realidad aumentada tiene que cumplir estas características, según Azuma:

- 1.-Combinar mundo real y virtual. Sistema incorpora información sintética al mundo real.
- 2.-Interacción en tiempo real. El sistema es capaz de adaptarse a la información percibida en cada momento.
- 3.-Alineación 3D. Es muy importante que la información sintética que se le añade a las imágenes del mundo real este perfectamente alineada con este, para así conseguir un mayor realismo.

2.3.2. Niveles

Según Prendes Espinosa, una aplicación de realidad aumentada está clasificada en niveles. Estos niveles dependen de los grados de complejidad de la tecnología que implementa las diferentes aplicaciones. Estos niveles son:

-Nivel 0 (enlazado con el mundo físico). Las aplicaciones de este nivel solamente se dedican al escaneo de un código que transmite toda la información necesaria al software. Este código es un hiperenlace a otros contenidos. Ver ilustración 6.



Ilustración 6

-Nivel 1 (RV con marcadores). Aplicaciones que utilizan marcadores (imágenes en blanco y negro, cuadrangulares y con dibujos esquemáticos. Ver ilustración 7.



Ilustración 7

-Nivel 2 (RV sin marcadores). Las aplicaciones que conforman este nivel utilizan la posición GPS y la brújula para saber la ubicación del usuario y generar la realidad virtual respecto a estos datos. También son capaces de reconocer superficies. Ver ilustración 8.



Ilustración 8

-Nivel 3 (visión aumentada). Basadas en dispositivos de alta tecnología como Google Glass, HoloLens... Capaces de ofrecer una experiencia contextualizada, inmersiva y personal. Ver ilustración 9.



Ilustración 9

2.3.3. Componentes

Según Edgar Mozas Fenoli, un sistema de realidad aumentada debe estar formado, por lo general, por:

-Cámara: Elemento encargado de captar el mundo real sobre el que aplicaremos la realidad aumentada.

-Procesador: Elemento hardware capaz de combinar las imágenes captadas con la información que se le va a superponer.

-Marcador: Se encarga de reproducir las imágenes creadas por el procesador.

-Software: Programa informático específico que gestiona el proceso de la realidad aumentada.

-Pantalla: Dispositivo donde se proyecta el resultado de combinar el mundo real y la información virtual.

-Activador: Elemento del mundo real que el software reconoce y con el que se transmite información para generar el proceso de realidad aumentada como: ubicación, orientación...

2.3.4. Tipos de display

Podemos encontrar 3 técnicas principales para mostrar la realidad aumentada:

Display en la cabeza:

Una pantalla instalada en la cabeza (HMD Head-MountedDisplay) es la encargada de superponer la información virtual sobre las imágenes del mundo real que está captando el usuario en ese momento. Podemos verlo en la ilustración 10.



Ilustración 10

Display en mano

Son aquellos dispositivos que cuentan con una pantalla pequeña que cabe en la mano de un usuario.

Anteriormente estos dispositivos empleaban brújulas digitales y GPS. Más tarde con sistemas como ARToolKit, nos permitía añadir información digital a las secuencias de video en tiempo real. Hoy en día los sistemas como SLAM o PTAM son empleados para el seguimiento.

Este tipo de display promete ser el primer éxito comercial de la tecnología de realidad aumentada. Ver ilustración 11



Ilustración 11

Display espacial

Esta tecnología hace uso de proyectores digitales para mostrar información gráfica sobre los objetos físicos.

La característica clave es que la pantalla está separada del usuario del sistema.

Este sistema tiene como ventaja el hecho de que el usuario no tenga que llevar el equipo encima y un menor desgaste ocular producido por una pantalla. Ver ilustración 12.

Figura 8



Ilustración 12

2.4. Software

Por lo general, los métodos del software para generar el proceso de realidad aumentada consisten en:

1.- Reconocimiento del mundo real. En esta etapa se lleva a cabo el proceso de reconocimiento de esquinas, regiones, procesamiento de imágenes....

2.- Creación de la información virtual usando los datos obtenidos de la primera etapa. Según el nivel de la aplicación este proceso puede ser más o menos costoso.

2.4.1. Plataformas de desarrollo de realidad virtual

Según la Web (PlusUtra consultoria, 2013) la plataforma de desarrollo es *“un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones).*

Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles”.

2.4.1.1. Tipos de plataformas de desarrollo

ARGON

Es una plataforma de realidad aumentada basada en la Web. Logo en ilustración
13

Ventajas: Es una plataforma que está tecnológicamente muy avanzada.

Desventajas: Es una plataforma que solo se utiliza para iOS.

Costo: Gratuito.



Ilustración 13

ATOMIC AUTHORIZING TOOL

Ventajas: Cualquier persona que no tenga conocimientos bastos en programación puede utilizarla y crear su aplicación, es una multi-plataforma. Ver logo en ilustración 14.

Desventajas: Solo funciona para computadoras, no es utilizable en dispositivos móviles.

Costo: Gratuito.



Ilustración 14

ARTOOLKIT

Es una biblioteca que permite la creación de aplicaciones de realidad aumentada, utilizando las capacidades de seguimiento de vídeo. Ver logo en ilustración 15.

Ventajas: Es una plataforma que puede leer cualquier tipo de imagen.

Desventajas: Es un poco difícil trabajar con esta plataforma porque utiliza una programación un tanto avanzada.

Costos: Descarga gratuita, pero posteriormente se requiere licencias.



Ilustración 15

LAYAR

Es una plataforma de reconocimiento de imágenes y marcadores. Ver logo en ilustración 16.

Ventajas: Es una plataforma muy flexible y está disponible para Android e iOS.

Desventajas: El visualizador de layar es gratuito su descarga, pero para programar en él tiene un costo.

Costo: No es preciso.



Ilustración 16

LEARNAR

Es una organización que, mediante su página Web, utiliza realidad aumentada que es capaz de enseñar a la comunidad virtual conocimientos básicos en distintas áreas. Ver logo en ilustración 17.

Ventajas: No tiene costo y es una manera sencilla que los niños interactúen con esta aplicación.

Desventajas: Trabaja única y exclusivamente mediante su página Web.

Costo: Gratuito.



Ilustración 17

WIKITUDE

Es considera la plataforma de realidad aumentada más completa de todas. Ver logo en ilustración 18.

Ventajas: Ya que trabaja en muchos sistemas es muy compatible, fácil de utilizar, gratuita su descarga.

Desventajas: Por ser una plataforma muy completa, la elaboración de una aplicación en dicha plataforma es más compleja.

Costo: Gratuita la descarga del visualizador, pero la plataforma de programación tiene un costo, el cual es desconocido.



Ilustración 18

2.5. Aplicaciones

En la actualidad, la realidad aumentada ha supuesto un gran avance en muchos campos como la educación, la medicina. Con el avance de la tecnología no quedara mucho tiempo para que usemos la realidad aumentada gran parte de nuestro tiempo, ya que facilita muchos aspectos de la vida de los usuarios.

Hoy día algunos de los campos en los que está presente la realidad aumentada son:

INDUSTRIA

El desarrollo de aplicaciones de realidad aumentada en este campo cada vez es mayor, ya que ayudan a mejorar la productividad en los ciclos de trabajo.

Un ejemplo podría ser la visualización de proyectos de construcción o diseños industriales.

Aplicaciones

eKurzinfo

La aplicación se encarga de ir detallando las cualidades, calidades y cómo se usa el coche. Le dice al usuario lo que debe de hacer, como hacerlo y en qué momento realizarlo. Ejemplo de visualización en ilustración 19.



Ilustración 19

EDUCACIÓN

Según el informe *Horizon 2017* se cita a la realidad aumentada en la educación como una de las tendencias para los siguientes 3 años.

Gracias a la inclusión de la realidad aumentada en este campo los estudiantes tienen la oportunidad de explorar el contorno que les rodea de manera distinta y disfrutar de un proceso de aprendizaje interactivo y enriquecedor.

Aplicaciones

Geometry

Creada por Arloon, permite estudiar geometría observando cada cuerpo geométrico desde todas sus perspectivas. Ejemplo de aplicación en ilustración 20.



Ilustración 20

MEDICINA

Dentro de la medicina, la realidad aumentada puede ayudar a muchos campos de la misma, un ejemplo es en la cirugía. Mediante escáneres y resonancias magnéticas es posible recopilar, de manera no invasiva, información sobre el paciente. De esta manera se puede realizar una reconstrucción y proyectarla sobre el cuerpo físico en tiempo real, haciendo las cirugías más eficientes y con mayores garantías para el paciente.

También puede resultar muy útil para enriquecer la información obtenidas de las herramientas de diagnóstico, como radiografías.

Aplicaciones

Iskull

Aplicación realizada por Maje Innovation y la universidad de navarra en 2012.

Esta aplicación permite estudiar las diferentes partes del cráneo gracias a la recreación aumentada de un cráneo. Ejemplo en ilustración 21.



Ilustración 21

ARQUITECTURA

La realidad aumentada puede usarse para la representación de edificaciones con todo lujo de detalles, además de animaciones y la posibilidad de interacción con el usuario, haciéndolas así mucho más atractivas.

Es posible recrear un edificio 3D a partir de un plano del mismo.

Aplicaciones

Visuarq

Esta aplicación permite mostrar proyectos terminado sobre plano. Ejemplo de uso en la ilustración 22.



Ilustración 22

MARKETING

Gran parte de la evolución de esta tecnología es en el ámbito comercial

Algunos ejemplos son las campañas publicitarias que muestran modelos de los objetos 3D.

Aplicaciones

Catálogo en realidad aumentada

A través de una aplicación de realidad aumentada, se puede escanear un producto del catálogo de Ikea y luego colocarlo en lugar de la casa que mejor le parezca al usuario. Ejemplo en la ilustración 23.

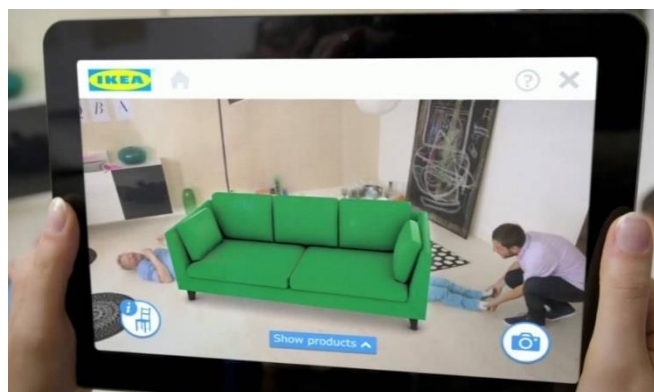


Ilustración 23

TURISMO

Gracias a la inclusión de la realidad aumentada en el turismo, el usuario puede disfrutar de una experiencia totalmente nueva añadiendo contenido visual o sonoro que proporcione información sobre la localización donde se encuentra.

De la misma manera, también puede ser útil para la reconstrucción de edificios o estatuas con algún tipo de valor histórico.

Aplicaciones

Guía Turística de Segovia

Es un nuevo concepto de guía, donde la información nos rodea y nos permite interactuar con ella. Ejemplo en la ilustración 24.



Ilustración 24

3. Realidad Aumentada y arqueología

3.1. Introducción

A día de hoy existe una revolución en la forma de interpretar la historia gracias a las nuevas tecnologías en el campo del patrimonio cultural. Gracias a lo novedoso y eficiente de estas herramientas, se han llevado a cabo importantes proyectos y experiencias cuyo objetivo ha sido potenciar las posibilidades y la difusión del objeto cultural.

Gracias a los sistemas de realidad aumentada móvil (RAM) los sitios históricos y arqueológicos son, cada vez, más atractivos para un usuario no experto en la materia. La información sobre estos lugares puede consultarse in situ y, además, se adapta al usuario, permitiéndole elegir idiomas, formatos...

3.2. Reconstrucción yacimiento arqueológico

Hoy día crear una experiencia de realidad aumentada sobre un yacimiento arqueológico es costoso, ya que se necesitan muchos procesos de recolección de datos y de creación de la realidad aumentada además específico para un lugar, es decir, todo el estudio y creación de la experiencia se realiza sobre un yacimiento concreto. Para crear una experiencia de realidad aumentada sobre otro yacimiento sería necesario repetir todo el proceso de estudio de dicho yacimiento, cosa que no siempre es posible por falta de recurso o de interés del yacimiento.

3.2.1. Paso en la reconstrucción de un yacimiento arqueológico

1.-Escaneado Láser

Esta tecnología nos ayuda a generar una nube de puntos para representar el mínimo detalle de objetos o restos arqueológicos, además de permitirnos recopilar todos esos datos sin dañar el objeto.

Con esta información se generan vistas 3D que permiten la comprensión global del conjunto u obtener una primera información dimensional del mismo.

El problema de esta tecnología es la cantidad de puntos que es capaz de capturar, siendo así necesaria una “criba” de los mismos antes de una representación. Esto se complementa con otro escáner más detallado de las partes de interés arqueológico. Ejemplos de mapas realizados con escaneado láser en la ilustración 25.

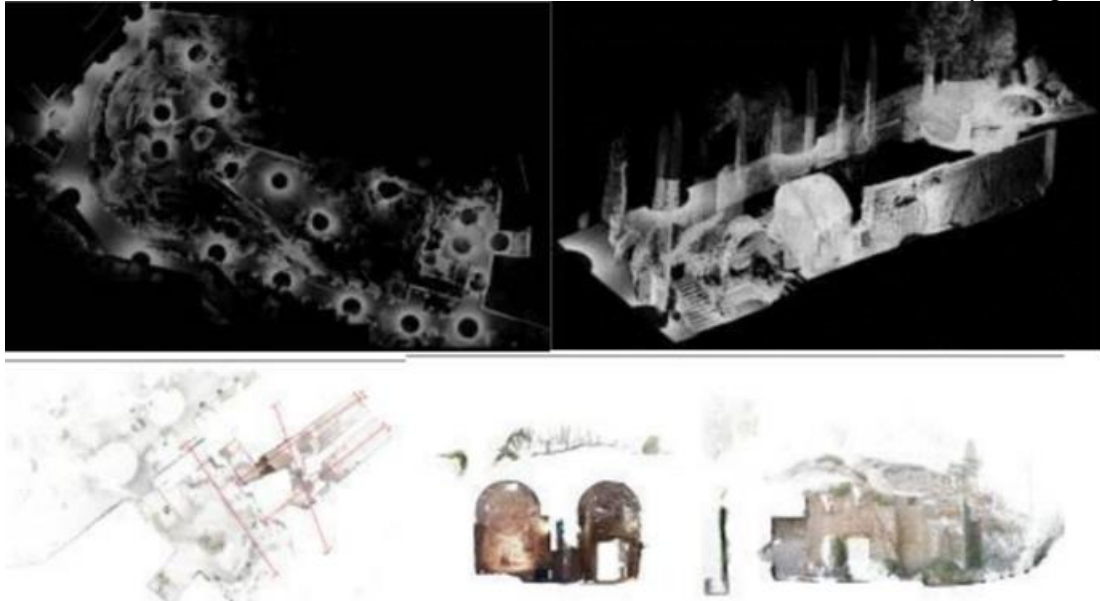


Ilustración 25

2.-Levantamiento topográfico y georeferenciación

Uno de los problemas de trabajar con nubes de puntos es establecer puntos base fiables para el posicionamiento cartográfico.

El siguiente paso para la publicación de estos datos en un Sistema de Información Geografía (SIG) pasa por la conversión de estas coordenadas al sistema Universal Transverse Mercator (UTM) en el que desaparece la coordenada Z.

3.-Edición y mallado de la nube de puntos

El primer paso consiste en optimizar la nube de puntos de manera que las formas fueran reconocibles, pero no pesadas a nivel digital. Para esto se usan varias técnicas basadas en la división del espacio en secciones (Rivest, 1974). Estos métodos miden los límites de un conjunto de coordenadas espaciales y las separan en grupos.

De cara a la generación del mallado, casi siempre se empieza por calcular las normales, de modo que luego se puedan agrupar en conjuntos para la detección de superficies.

El proceso de construcción de la malla es complicado y son necesarias varias pruebas antes de conseguir un buen resultado. El proceso de mallado usa mecanismos basados en la generación de normales para cada vértice según la disposición de los puntos cercanos, después de esos se usan algoritmos capaces de generar un conjunto de puntos que pertenecen, dentro de un margen, al mismo plano al tener una normal similar. Primeros trabajos de edición y gestión de la nube de puntos con aplicaciones comerciales ilustración 26.



Ilustración 26

4.-Representación

Existen un amplio número de sistemas y modalidades de representación, En general, la manera de representación que más información ofrece es la de modelos 3D. Ejemplos de representación en ilustración 27.



Ilustración 27

3.3. Musicalización de yacimientos arqueológicos

Podemos encontrar otra vertiente a la hora de aplicar la realidad aumentada a yacimientos arqueológicos y esta es la musicalización de yacimientos arqueológicos.

Esta técnica resulta especialmente útil para la difusión del patrimonio cultural de aquellos yacimientos de difícil acceso o para aquellos usuarios que, por diferentes motivos, les resulta dificultoso el acceso a dicho yacimiento.

1.-Diseño del sistema

El sistema se compone de cuatro unidades básicas: Un computador, una webcam HD, un monitor HD y una superficie con una serie de marcadores dispuestos de forma específica. Ejemplo del sistema de realidad aumentada en ilustración 28.

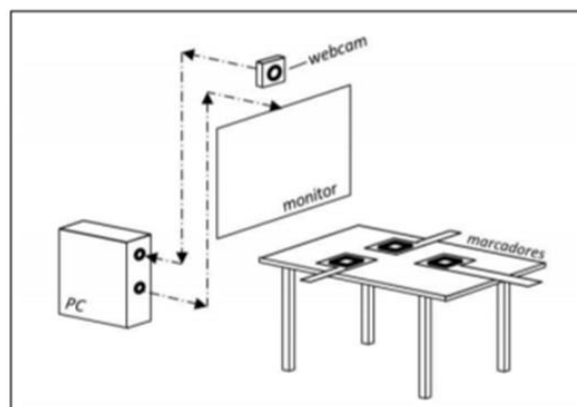


Ilustración 28

En este sistema, la cámara se encarga de grabar la mesa, donde están puestos los marcadores. Estas imágenes se envían al computador donde, junto al software específico, se encarga de procesarlas y de crear una imagen saliente que se muestra por el monitor. Estas imágenes que genera dependen de los marcadores que se encuentren sobre la mesa, pudiendo así representar desde una pieza histórica hasta grandes construcciones.

Los marcadores están adheridos a unas paletas para facilitar la manipulación por parte del usuario y así poder observar las diferentes vistas del objeto representado.

La interacción principal con el usuario consistirá en la manipulación de estas paletas. Ejemplo de interacción, ilustración 29.



Ilustración 29

Este sistema tiene un diseño económico y permite generar nuevos escenarios únicamente sustituyendo los marcadores por otros. También permite el traslado de “yacimientos” de un museo a otro simplemente trasladando los marcadores.

La característica esencial del software es la posibilidad de crear aplicaciones multimarcaador, permitiendo reconocer varios marcadores simultáneamente y posición los correspondientes modelos 3D.

El diseño del marcador es el habitual en este tipo de sistemas de realidad aumentada. El marcador tiene un marco negro que sitúa y posición, y un dibujo en el interior a escala de grises que identifica el marcador. Ejemplo de marcadores, ilustración 30.

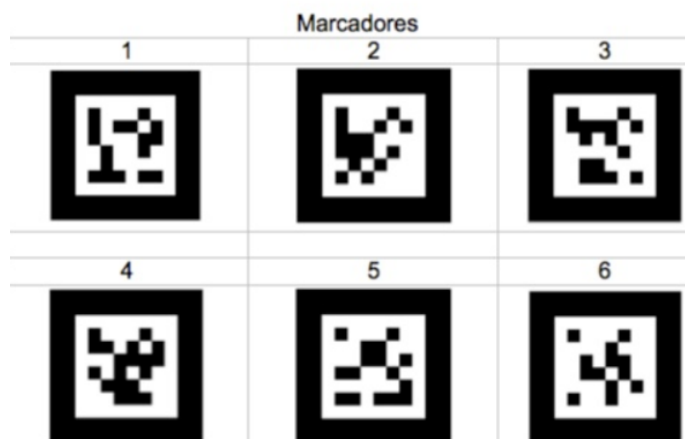


Ilustración 30

Gracias a un estudio de esta técnica de musicalización de yacimientos arqueológicos, en el que los usuarios que han probado dicho sistema lo valoran del 1 al 5, podemos observar que ha sido bien acogido, sobre todo por personas de menos de 40 años. Esto también se puede deber a la mayor predisposición por la tecnología de este colectivo. Resultados en Tabla 1.

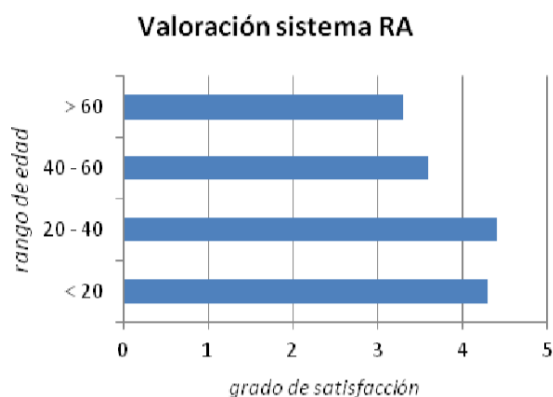


Tabla 1

3.4. Histórico de aplicaciones

1.-Archeoguide

Según los creadores de la plataforma se creó Archeoguide como una guía de un yacimiento arqueológico basada en realidad aumentada para así cerrar la brecha entre recreación, educación e investigación científica. Ejemplos de hardware necesario para su utilización y resultado de visualización en ilustración 31.



Ilustración 31

Arquitectura del sistema

Archeoguide utiliza una arquitectura cliente-servidor con tres subsistemas básicos: El servidor de información del sitio (SIS), las unidades móviles y la red. Arquitectura del sistema en ilustración 32.

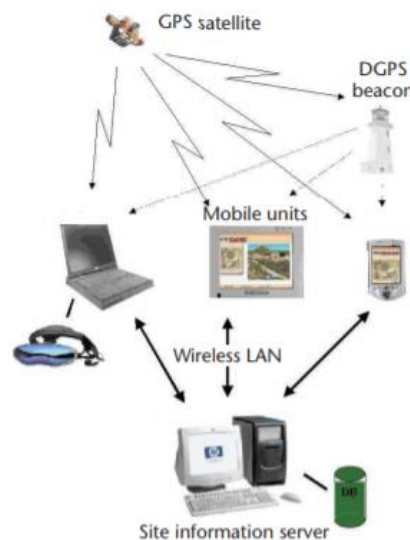


Ilustración 32

El posicionamiento es vía GPS. Esta tecnología no nos da la precisión que necesitamos, así que, se complementa con una válida de GPS diferencia (DGPS) ubicada en una posición conocida con precisión.

Seguimiento de posición y orientación

Los datos de posición DGPS dan una precisión de menos de 1 metro y estima el ángulo de visión con una precisión de 0,5 grados. Archeoguide muestra esta

información en un mapa digital del sitio arqueológico junto con las posiciones más importantes de los monumentos.

Seguimiento óptico

En aplicaciones en tiempo real el movimiento abrupto e impredecible hace enfoques secuenciales inciertos y frágiles.

Archeoguide utiliza un conjunto de imágenes de referencia calibradas desde el punto de vista del usuario.

Análisis

Se probó esta aplicación en la antigua Olimpia. Sitio donde recibió muy buenas críticas de usuarios ordinarios, personal del sitio, arqueólogos y expertos en tecnología.

A los usuarios les gustó las reconstrucciones y la información adicional que se ofrecía, sin embargo, se sentían incómodos con tecnología necesaria para visualizar la realidad aumentada, como gafas o cascos.

2.-LifeClipper

Proyecto de realidad aumentada al aire libre creado por Basilea basado en el histórico barrio de St.Alban en Basilea, Suiza.

Está compuesto por una serie de partes que son: Computadora portátil, pantalla montada en la cabeza, cámara de video, micrófono, sensor GPS, brújula y sensores de precisión.

Cuando se mueve, el sistema utiliza coordenadas de GPS para activar imágenes y sonidos para el usuario.

El principal problema de esta aplicación es que no es accesible a todos los usuarios debido a que necesita demasiada tecnología instalada en el usuario. Ejemplo de hardware de la aplicación en ilustración 33.



Ilustración 33

3.LifePlus

Uno de los proyectos más importantes sobre realidad aumentada y patrimonio, promovido por la Unión Europea y desarrollado en la Universidad de Ginebra.

El escenario escogido para poner en práctica este proyecto fue Pompeya (Nápoles, Italia). En esta experiencia se quiso insertar, dentro de los escenarios reales de Pompeya, animaciones virtuales de flora y fauna y recreaciones de escenas cotidianas de la vida romana.

Mediante el uso de HMD fue posible diseñar un sistema inalámbrico capaz de determinar los gráficos 3D que se insertaban en la imagen del entorno real que está visualizando el usuario.

Además de los gráficos, también se reproducen sonidos o narraciones relacionadas con la escena. Ejemplos de reconstrucción en ilustraciones 34 y 35.



Ilustración 34



Ilustración 35

4.-Arbela Layers Uncovered(ALU)

Basado en tecnologías usadas en proyectos anteriores como Archeoguide o Life Clipper.

Aplicación de realidad virtual basada en el situó histórico de Arblea, ubicado en el corazón de la ciudad de Arble.

La aplicación funcionaria sobre un dispositivo móvil, sin necesidad de otros dispositivos como HMD, sensor GPS...

Los objetivos de ALU son:

- Construir un sistema de AR auto guiado que permita a los visitantes acceder a la información relacionada con el contexto mientras navega alrededor del sitio histórico.

- Restaurar virtualmente edificios parcial o totalmente dañados y estructuras en un sitio histórico.

- Proporcionar un medio para recopilara experiencias de los visitantes del sitio.

Este proyecto aún está en fase de diseño.

Ejemplos de la interfaz de la aplicación en ilustraciones 36 y 37.



Ilustración 36



Ilustración 37

4. Diseño de aplicación de realidad aumentada: Google ARCore

4.1. Introducción

ARCore es la plataforma de Google para crear experiencias de realidad aumentada. Utilizando diferentes API ARCore permite que un teléfono sea capaz de ser consciente de su entorno, entender el mundo e interactuar con la información que este ofrece. Logo de ARCore en ilustración 38.

ARCore usa tres capacidades claves para integrar el contenido virtual con el mundo real que ve la cámara del teléfono:

- Seguimiento de movimiento:** permite que el teléfono entienda y rastree su posición en relación con el mundo.

- Comprensión ambiental:** permite que el teléfono detecte el tamaño y la ubicación de todo tipo de superficies.

- Estimación de luz** permite al teléfono estimar las condiciones de iluminación actuales del entorno.



Ilustración 38

4.2. Historia

ARCore es la tecnología de realidad aumentada para dispositivos móviles sucesora del obsoleto proyecto tango.

4.2.1.-Proyecto Tango

Proyecto Tango es una tecnología de realidad aumentada, enfocada a los Smartphone, desarrollada por Google. Logo proyecto tango ilustración 39.



Ilustración 39

El software del Proyecto Tango funciona mediante tres tipos de funcionalidades:

-Seguimiento de movimiento: Identificación y uso de características visuales del entorno. Estos datos, combinados con los obtenidos del acelerómetro y del giroscopio, son usados para realizar un seguimiento del movimiento del dispositivo.

-Aprendizaje de área: Creación de un mapa basándonos en los datos recogidos del entorno.

-Percepción de profundidad: Consiste en la detección de distancias, tamaños y supervisiones en el entorno.

El día 1 de marzo de 2018 Google pone fin al soporte a Proyecto tango. Quizás, una de las causas del fin de este proyecto es que los módulos necesarios para hacer funcionar el software de Tango requerían demasiado espacio en el dispositivo, haciendo poco atractivo a nivel comercial. Dispositivo con soporte para proyecto tango, ilustraciones 40 y 41.



Ilustración 40

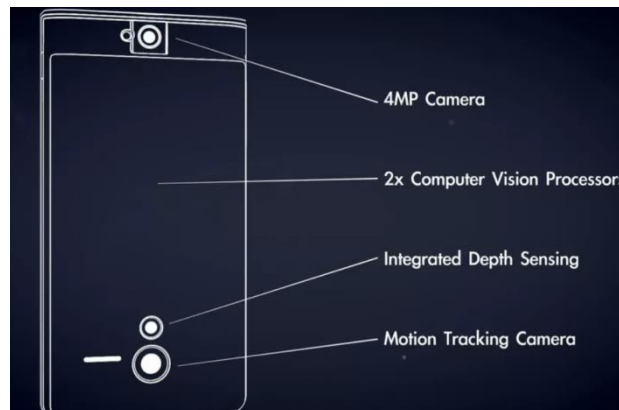


Ilustración 41

Google centra todos sus esfuerzos en una solución “solo software”. ARCore competirá directamente con ARKit de Apple, haciendo prácticamente lo mismo que Tango pero sin requerir hardware específico.

4.3. Funcionamiento Google ARCore

4.3.1. Seguimiento del movimiento

ARCore utiliza un proceso llamado odometría y mapeo concurrente (COM) para conocer la ubicación del dispositivo respecto al mundo que lo rodea.

ARCore detecta objetos basándose en características distintas en las imágenes que recoge con la cámara; Estos puntos se llaman “feature points” (puntos característicos) y son usados para calcular el cambio de ubicación. Esta información, junto con mediciones inerciales (IMU), es usada para estimar la posición del dispositivo (posición y orientación) de la cámara en relación con el entorno que la rodea.

Al alinear la posición de la cámara virtual (en la cual se generan los objetos 3D) con la posición de la cámara del dispositivo ARCore es capaz de superponer la imagen de la cámara virtual a la del dispositivo y así permitirnos ver, en la perspectiva correcta, el objeto 3D generado.

Odometría

Odometría es el estudio de la estimación de la posición de vehículos con ruedas durante la navegación.

En nuestro caso, al no tener ruedas (hecho importante en este estudio) es necesario combinar la información de la cámara con la de los sensores de movimiento, haciendo el recorrido del movimiento de nuestro dispositivo, procesando la información del acelerómetro y analizando los objetos que capta nuestra cámara. Etapas de la odometría en la ilustración 42.

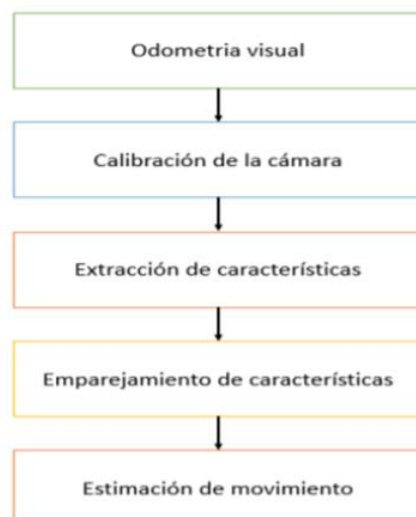


Ilustración 42

IMU

La unidad de medición inercial (inertialmeasurementunit) es un dispositivo capaz de medir una fuerza específica y la velocidad angular utilizando una combinación de acelerómetros y giroscopios.

4.3.2. Comprensión ambiental

ARCore mejora constantemente su entendimiento del entorno que le rodea mediante la detección de puntos y planos característicos.

ARCore es capaz de buscar grupos de puntos que puedan estar en superficies similares, horizontales o verticales, como pueden ser mesas, tableros, paredes. Procesando estos puntos crea planos que pone a disposición de la aplicación. Además, detecta información sobre los límites de dichos planos.

Debido a que la manera de detectar estos planos consiste en la detección de grupos de puntos, puede darse el caso en el que no encuentre dichos puntos y la detección falle como, por ejemplo, en el caso de superficies mate y amplias, como puede ser una pared blanca. Ejemplo de plano detectado en ilustración 43.

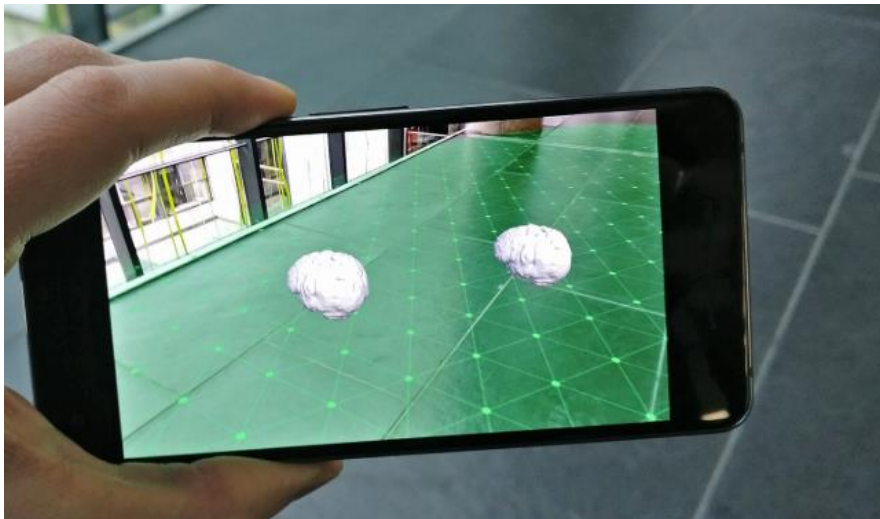


Ilustración 43

4.3.3. Estimación de luz

ARCore puede detectar información sobre la iluminación del entorno y así proporcionar intensidad y corrección del color al objeto virtual generado permitiendo iluminar este en las mismas condiciones que el entorno que le rodea aumentando así la sensación de realismo.

Para esto, ARCore utiliza HDR ambiental. Esta técnica utiliza los datos de luz disponibles de la escena real y extiende dicha luz a los objetos digitales.

El HDR ambiental que usa ARCore se basa en tres API:

-Luz direccional principal: ayuda a colocar sombras en la dirección correcta.

-Armónicos esféricos ambientales: ayuda a modelar la iluminación ambiental desde todas las direcciones.

-HDR cubemap: proporciona reflejos y reflejos especulares.

HDR

Las imágenes de alto rango dinámico(HighDynamicRange) son un conjunto de técnicas que permiten un mejor rango dinámico de luminancias entre las zonas más claras y las más oscuras de una imagen Ejemplo de creación de HDR en la ilustración 44.

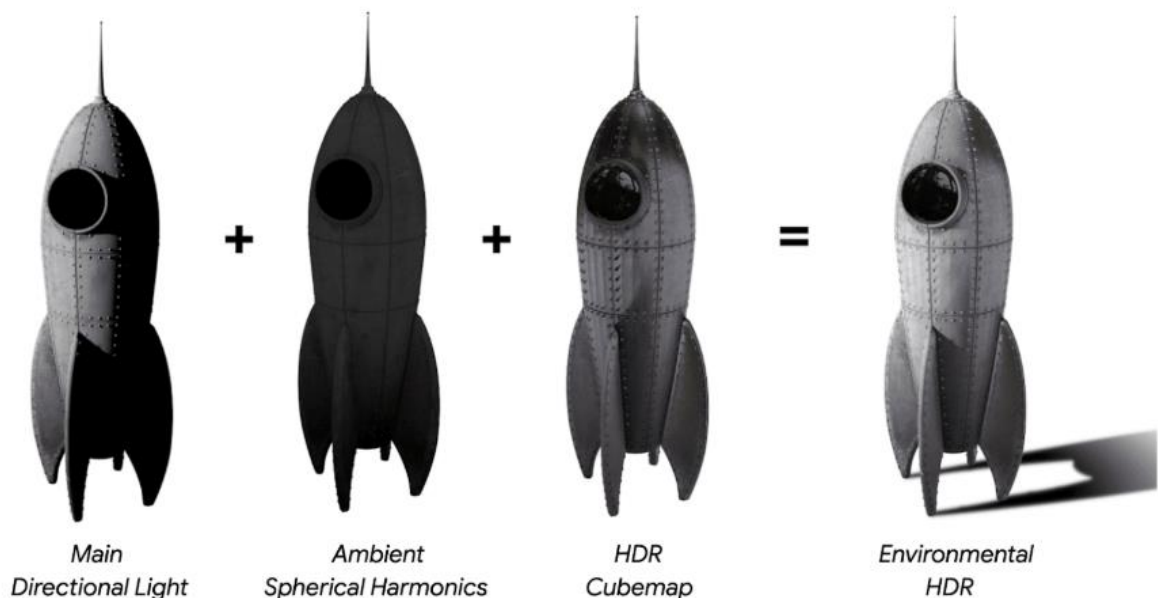


Ilustración 44

4.4. APLICACIONES ARCORE

Google nos propone una serie de funcionalidades, bastante prácticas, de su tecnología.

Superposición de modelos

-Aplicación para arquitectos que permite superponer modelos sobre el mundo real para mostrar cómo se vería una construcción terminada. Ejemplo de visualización en ilustración 45.



Ilustración 45

Entrenamiento con realidad aumentada

-Guía, por pasos, para realizar una tarea cualquiera. En esta ocasión adaptada para la tarea de realizar un café. Los usuarios fueron capaces de realizar un café solamente con la ayuda de esta aplicación. Ejemplo en ilustración 46.



Ilustración 46

Controlar la posición virtual desde el mundo real

-Forma de desplazarnos por Street View sin tener que pulsar las flechas de avance, ya que solo tenemos que andar físicamente. Ejemplo de visualización de Street View en ilustración 47.



Ilustración 47

Detectar contenido de realidad aumentada

-Consistiría en controlar la profundidad de campo y la saturación de la cámara en función al movimiento del usuario. Este experimento permite que los objetos virtuales destaquen, atrayendo la atención de los usuarios y alentándolos a explorar e interactuar con ellos. Ejemplo de este proceso en ilustraciones 48 y 49.



Ilustración 48



Ilustración 49

4.5. Otras tecnologías

4.5.1. ARKit

“Imagina que la línea que separa lo real de lo virtual dejara de existir.

Podrías convertir tu aula en el centro del universo. O vivir el pasado con tanta intensidad como si fuera el presente. Sería como volver a descubrir todo lo que ya conoces.”.

Este es el anuncio con el que se presenta la realidad aumentada de Apple.

ARKit es la plataforma de desarrollo de realidad aumentada de Apple para dispositivos móviles con IOS. Se introdujo con IOS 11 y funciona en dispositivos Core A9 y superiores. Logo de ARKit en ilustración 50.



Ilustración 50

Características

Seguimiento de movimiento: ARKit es capaz de rastrear de forma muy precisa la posición del dispositivo en referencia a los objetos reales que lo rodean usando odometría visual inercial (IVO).

Entendimiento ambiental: Con ARKit, los dispositivos Apple son capaces de escanear y analizar la escena recogida por la cámara y determinar diferentes formas.

Estimación de la luz: Gracias a la cámara, el software también es capaz de captar la luminosidad del entorno y adaptar los objetos virtuales a esa luminosidad y así conseguir una experiencia más inmersiva.

Esta tecnología de Apple es de gran detalle, por tanto, precisa de más memoria, cosa que hace que los datos antiguos desaparezcan de la memoria. Podemos ver un ejemplo del funcionamiento en la ilustración 51.



Ilustración 51

4.5.2. Vuforia

Vuforia es un Kit de desarrollo de aplicaciones de realidad aumentada (SDK) para dispositivos móviles que permite la creación de aplicaciones de realidad aumentada.

Utiliza la tecnología de visión artificial para reconocer y rastrear imágenes en tiempo real. Esta capacidad permite a los desarrolladores posicionar y orientar modelos virtuales 3D combinándolos con imágenes del mundo real.

Esta tecnología es líder en el ámbito industrial. Soluciones RA para montajes, ejemplo en ilustración 52, y soluciones RA para servicios, ejemplo en la ilustración 53.

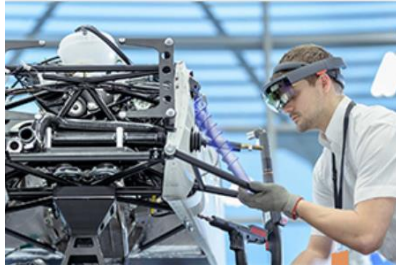


Ilustración 52



Ilustración 53

Arquitectura

Descripción de términos referentes a la arquitectura básica de vuforia:

Camera: Garantiza la captación del frame actual y su eficiente comunicación con el *Tracker*.

Image Converter: Se encarga de realizar la conversión de los datos captados por la cámara a un formato adecuado para el renderizado.

Tracker: Provisto de algoritmos de visión computacional para detectar y seguir los objetos en los fotogramas capturados por la cámara, almacenando el resultado en un objeto de estado.

Video Background Render. Procesa la imagen capturada por la cámara y la ofrece como Background de nuestra aplicación.

Application Code: el desarrollador ha de iniciar todos los componentes anteriores

User.defined Targest y Cloud Databases. Estos ficheros contienen la base de datos

4.6. Comparativa ARCore Vs ARKit Vs Vuforia

4.6.1. ARCORE VS ARKIT

Compatibilidad de los dispositivos con plataformas AR

-Hardware del sensor: los dispositivos Android son variables cuando se trata de hardware y sensores. ARCore requiere un dispositivo con una cámara orientada hacia atrás, un acelerómetro y un giroscopio. El requisito del giroscopio elimina la compatibilidad de ARCore con la mayoría de móviles Android de gama baja (los más usuales en el mercado).

Los dispositivos Apple han incluido esta tecnología de sensores desde hace ya algún tiempo.

-Sistema operativo: ARKit es compatible con todos los dispositivos lanzados desde septiembre de 2015 que ejecutan IOS 11.0, mientras que ARCore solo admite un subconjunto limitado de dispositivos con Android 7 o posterior. Además, los usuarios de Android deben descargarse una aplicación para poder usar ARCore, mientras que, los dispositivos Apple ya la llevan integrada.

-Requisitos de la aplicación: Para descargar una aplicación que utilice ARCore es necesario descargar también dicha plataforma. De modo que, si descargamos una aplicación basada en ARCore, como IKEA Place, ARCore también se descargará.

-Lanzamiento del dispositivo: Todos los dispositivos nuevos de IOS ya están listos para soportar ARKit, no ocurre lo mismo con los nuevos dispositivos Android.

Conclusión

A día de hoy, basándonos en la popularidad, ARCore es mucho más popular que ARKit, ya que los usuarios de Android son mucho mayores a los de IOS. Esto quiere decir que, a la larga, la realidad aumentada de Google será accesible para muchos más usuarios.

Sin embargo, el mejor reconocimiento y accesibilidad de ARKit mantienen a esta tecnología por delante de ARCore.

4.6.2. Vuforia vs ARKit/ARCore

En nuestros experimentos con estas herramientas, usamos las siguientes características e hicimos algunas observaciones significativas.

Objetivos de la imagen:

Vuforia: posee técnicas estándar para reconocer objetivos de imagen y es precisa en el aumento. No hay desviación en ningún ángulo, incluso con un cambio en la orientación del dispositivo.

ARCore y ARKit: bastante nuevo para el reconocimiento de imágenes y no está estandarizado. El aumento tuvo sacudidas y derivas muchas veces. Estos problemas ocurrieron incluso en su aplicación de demostración.

Rastreo de movimiento

Vuforia: Mientras caminaba alrededor del aumento, el objeto parecía girar a 90 grados y la imagen de la cámara resultante no era estable.

ARCore y ARKit: el aumento se desplaza un margen cuando cambiamos la orientación del dispositivo o cuando calibramos el enfoque del plano de tierra.

Detección del plano de tierra:

Vuforia: no detecta el suelo dinámicamente, sino que tiene un plano predefinido, en el que se coloca el modelo y reacciona al ángulo de la cámara y al giroscopio, lo que da una ilusión de profundidad y ubicación en el suelo. No es compatible con ciertos dispositivos y versiones de IOS.

ARCore y ARKit: detecta el plano de tierra con la ayuda de una nube de puntos de características. Es comparativamente mejor y estable a menos que se altere la orientación del dispositivo. El aumento se desplaza por un margen cuando cambiamos la orientación del dispositivo o volvemos a enfocar el plano de tierra.

VuforiaFusion : después de seguir los pasos de integración mencionados en la integración de Vuforia ARCore , no pudimos encontrar ninguna diferencia en los resultados. Se supone que solo se utilizaron las técnicas de Vuforia sobre ARCore / ARKit.

La ilustración 54 ofrece un resumen de lo anteriormente nombrado.

Herramientas / Técnicas	Objetivos de imagen	Rastreo de movimiento	Detección de plano de tierra
Kit AR Core y AR	No estable, altas derivas inesperadas.	Estable la mayoría de las veces.	Estable, pero deriva debido a problemas de anclaje cuando orientamos la pantalla.
Vuforia	Muy estable	No estable, se produce disparidad en la orientación.	Parece que detecta en función de la orientación y no con puntos de característica.
Vuforia Fusion	Muy estable	No se pudo ver ninguna mejora significativa, los resultados son los mismos que los de Vuforia.	

Ilustración 54

Conclusiones

En total, cada herramienta tiene sus pros y sus contras. Se recomienda más bien elegir una herramienta con características que satisfagan los requisitos específicos. Aquí están nuestras inferencias:

Vuforia es la mejor herramienta para AR basada en imágenes.

ARCore / ARKit es apto para el seguimiento de movimiento y la detección del plano de tierra.

VuforiaFusion no parece combinar las técnicas de Vuforia y ARCore / ARKit. Esperamos el próximo lanzamiento con pasos y técnicas de integración improvisados.

5. Aplicación de reconstrucción de yacimientos arqueológicos utilizando ARCore

5.1. Objetivos de la aplicación

El desarrollo de una aplicación móvil para la reconstrucción, en realidad aumentada, de yacimientos arqueológicos viene motivada por la posibilidad de hacer mas accesible la historia y el patrimonio a un mayor número de personas.

Hasta ahora la única forma que existía para ver un yacimiento era ir al lugar del mismo y, en la mayoría de los casos, contratar a un guía para poder entender como era antes ese yacimiento, su historia, sus edificios....

Con esta aplicación se pretende que el usuario pueda visualizar el yacimiento en su móvil, tal y como era en la época a la que pertenecía.

Además de la visualización del mismo, la aplicación ofrece la posibilidad de consultar la historia del yacimiento.

5.2. Fases de realización

Estudio: Investigación sobre el sistema al que va dirigida la aplicación. Indagación sobre aplicaciones similares, estudiando pros y contras de las mismas.

Análisis: Definición de los requisitos de nuestra aplicación con el objetivo de centrar el trabajo en lo que realmente se quiere conseguir.

Diseño: Una vez concluidas las fases previas, y con los objetivos de la aplicación claros, procedemos a realizar diagramas de funcionamiento, interfaz...

Implementación: Programación de la aplicación en C#.

Pruebas: Una vez que tenemos un prototipo de nuestra aplicación, realizamos pruebas para ver errores y posibles mejoras.

Documentación: Según una guía sobre desarrollo software [3], la documentación es algo indispensable para el correcto mantenimiento de cualquier producto software. Tanto a nivel de código como de documento, con el fin de dejar

constancia por escrito de las fases llevadas a cabo en el ciclo de vida del producto software.

Mantenimiento: La guía de principios de desarrollo software de Microsoft Ibérica [4] afirma que el mantenimiento de un producto software supone el 80% del ciclo de vida. En este proyecto puede no tener mucha cabida, pero se orienta a un trabajo futuro con bastante proyección con el fin de mejorar y ampliar la aplicación.

5.3. Análisis

En esta fase inicial de todo proyecto software se definen los requisitos de la aplicación y una visión general de la arquitectura de la misma.

A la hora de diseñar la aplicación se tiene en cuenta el catálogo de requisitos. Es por esto que la fase de análisis es tan importante para la correcta implementación de cualquier aplicación.

En este punto desglosaremos la funcionalidad y las características de nuestra aplicación a modo de catálogo de requisitos.

5.3.1. Catálogo de Requisitos

Este catálogo supone la especificación del comportamiento que se espera de cualquier aplicación software.

Basándonos en nuestras necesidades, se ha creado una lista enumerada de los requisitos que se consideran imprescindibles para el correcto funcionamiento de la aplicación.

5.3.1.1 Requisitos funcionales

Estos requisitos describen todas las interacciones que tendrán los diferentes usuarios con nuestra aplicación.

a) Gestión de usuarios

1.-Modo desarrollador

*registro: El registro se llevará a cabo de forma externa a la aplicación, ya que no todos los usuarios pueden ser desarrolladores.

*Identificación:

1.-Para iniciar sesión en modo desarrollador, el usuario deberá identificarse mediante su nombre y una contraseña.

2.-El sistema se encargará de validar los datos y permitir, o no, el acceso al usuario.

3.-En el caso de que los datos introducidos sean erróneos, aparecerá una ventana de error.

4.-Si los datos son correctos, se redirigirá al desarrollador a la ventana de trabajo.

*Cierre de sesión

1.-Una vez que el desarrollador haya concluido su trabajo, simplemente se tendrá que salir de la ventana de trabajo para así cerrar su sesión.

2.-Modo Usuario

-El modo usuario no tendrá validación de ningún tipo, ya que cualquier persona podrá acceder a esta aplicación y disfrutar de ella.

*Inicio

-Al seleccionar el modo usuario, la aplicación abrirá una ventana donde podremos seleccionar los diferentes mapas, previamente creados, dependiendo del lugar donde estemos.

*Fin

-Para salir, el usuario solamente tendrá que presionar el botón de salida y este será redirigido a la pantalla de inicio.

b) Aplicación

1.-Pantalla principal

-En esta pantalla podremos encontrar 3 botones.

1.-Usuario: La aplicación entrará en el modo usuario.

2.-Desarrollador: La aplicación abrirá un sistema de login para redirigir, o no, al usuario a la pantalla de desarrollo.

3.-Idioma: El usuario podrá elegir el idioma de la aplicación.

2.-Pantalla usuario

-Podremos encontrar 2 botones:

1.-Salir: salimos de la pantalla de visualización del usuario y somos redirigidos a la pantalla principal.

2.-Historia: Con este botón se abrirá una ventana donde podremos consultar la historia del yacimiento que estamos visitando.

3.-Pantalla desarrollador

-Podemos encontrar 2 botones.

1.-Objetos: Con este botón habilitamos o deshabilitamos un scrollbar donde podemos encontrar todos los objetos que podemos ubicar en el yacimiento.

-Una vez que seleccionemos un objeto, este botón desaparece y aparece uno de confirmar. Este sirve para validar la posición, rotación y escalado del objeto que acabamos de posicionar.

2.-Salir: salimos de la pantalla de desarrollador, cerramos sesión del mismo y volvemos a la pantalla principal.

5.3.1.2 Requisitos no funcionales

Requisitos complementarios o atributos de calidad.

a) Documentación

-Manual de usuario de la aplicación.

-Codificación de la aplicación clara y bien documentada para que cualquier otro programador pueda introducir funcionalidades extras.

b) Seguridad

-Para poder crear y modificar los mapas, será necesario identificarse.

-La identificación permanecerá activa mientras no se cierre la sesión.

c) Portabilidad

-La aplicación será compatible con versiones superiores a Android 7.0.

-Solo estará disponible para aquellos dispositivos con soporte para Google ARCore.

d) Interfaz y usabilidad

-La interfaz deberá ser lo más intuitiva posible, con pocos botones y con códigos de colores reconocibles por el usuario.

5.4. Metodología de desarrollo

Se trata de un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de software. Una metodología de desarrollo software se basa en:

- Herramientas, modelos y métodos para ejecutar dicho proceso de desarrollo.

Para este proyecto se usará la metodología *Rapid Application Development (RAD)*, ya que se centra en la experiencia con usuario y a que su construcción debe ser en poco tiempo.

En este modelo, el prototipo debe ser construido en poco tiempo y sin muchos recursos. La creación del prototipo se basa en la representación de los aspectos software visibles por el usuario. La rápida creación del prototipo permite al usuario poder utilizar la aplicación y gracias a esto, permite al equipo obtener mucha información y guiar su trabajo hacia los aspectos más importantes, derivados del uso de la aplicación por el usuario.

Las etapas que recorre un software construido con esta metodología son:

- Plan rápido, modelado, desarrollo, entrega, retroalimentación, comunicación y entrega final.

5.4.1. Contenido de cada prototipo

Prototipo 1: En este prototipo aplicamos el diseño del menú principal y creamos las vistas de desarrollador y usuario. Poniendo los botones para la interacción del usuario con la aplicación.

Prototipo 2: Creación de sistema de login para el perfil de desarrollador, desarrollo de modelos 3D e integración de los mismos en la aplicación. Sistema de elección de idioma.

Prototipo 3: Modificación de objetos 3D, tanto de origen de coordenadas, como escalado.

Prototipo 4: Modificación de objetos 3D, creando 2 variantes de ubicación de los mismos:

- ubicar cada objeto 3D en su posición correspondiente
- ubicar un conjunto con todos los objetos 3D de la aplicación
- Inclusión ventana de selección de yacimiento para visualizar
- Inclusión botones de borrado de objetos ya colocados

5.5. Diseño

En este punto se aborda el diseño de la aplicación software basándonos en los requisitos previamente establecidos.

El diseño de la aplicación se ha ido modificando a lo largo del desarrollo de la misma, debido a:

- Pruebas no satisfactorias
- adquisición de nuevos conocimientos
- Plazos de entrega

5.5.1. Tecnologías usadas

Enumeramos las diferentes tecnologías y herramientas usadas durante el desarrollo del proyecto:

-Google ARCore: Es la base del proyecto, sobre esta tecnología basamos el funcionamiento de nuestra aplicación.

-Android SDK: Necesario para el funcionamiento de Google ARCore.

-Blender: programa de diseño de objetos en 3D. Con este programa se han modelado todos los edificios disponibles en la aplicación.

-Unity: entorno de desarrollo de la aplicación. Ha sido necesario añadir algunos plugins para el correcto funcionamiento de Google ARCore.

-**Visual Studio**: Entorno de programación de los scripts necesarios en la aplicación.

-**NinjaMock**: Plataforma online para la creación de maquetas para simular la interfaz gráfica de una aplicación.

5.5.2. Diseño de pantallas

Primer prototipo

En este primer prototipo encontramos las pantallas de inicio, visualización en modo usuario y visualización en modo desarrollador.

En este prototipo solo se pretende diseñar la interacción del cambio de pantallas. Representación en la ilustración 55.

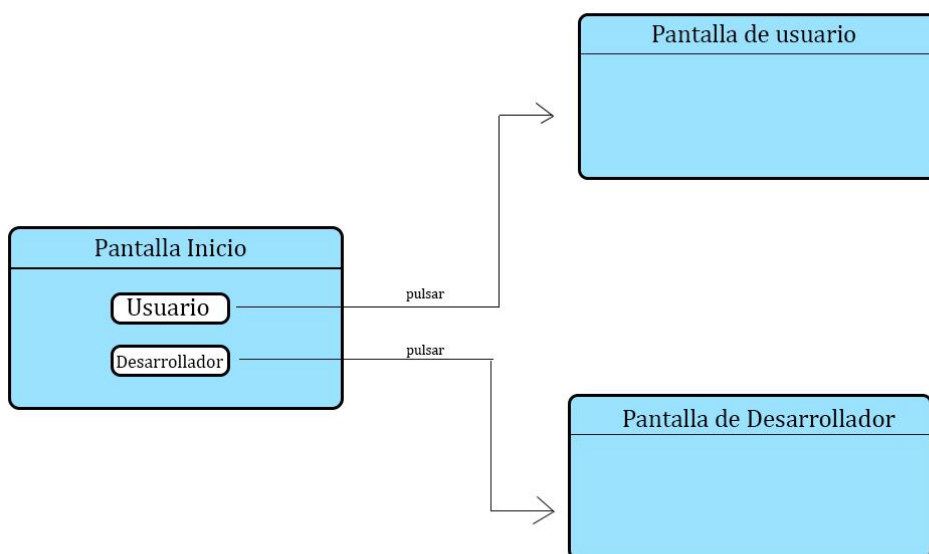


Ilustración 55

Segundo prototipo

En este segundo prototipo introducimos el sistema de login del usuario de tipo desarrollador y un botón para el cambio de idioma. Podemos apreciar los cambios en la ilustración 56.

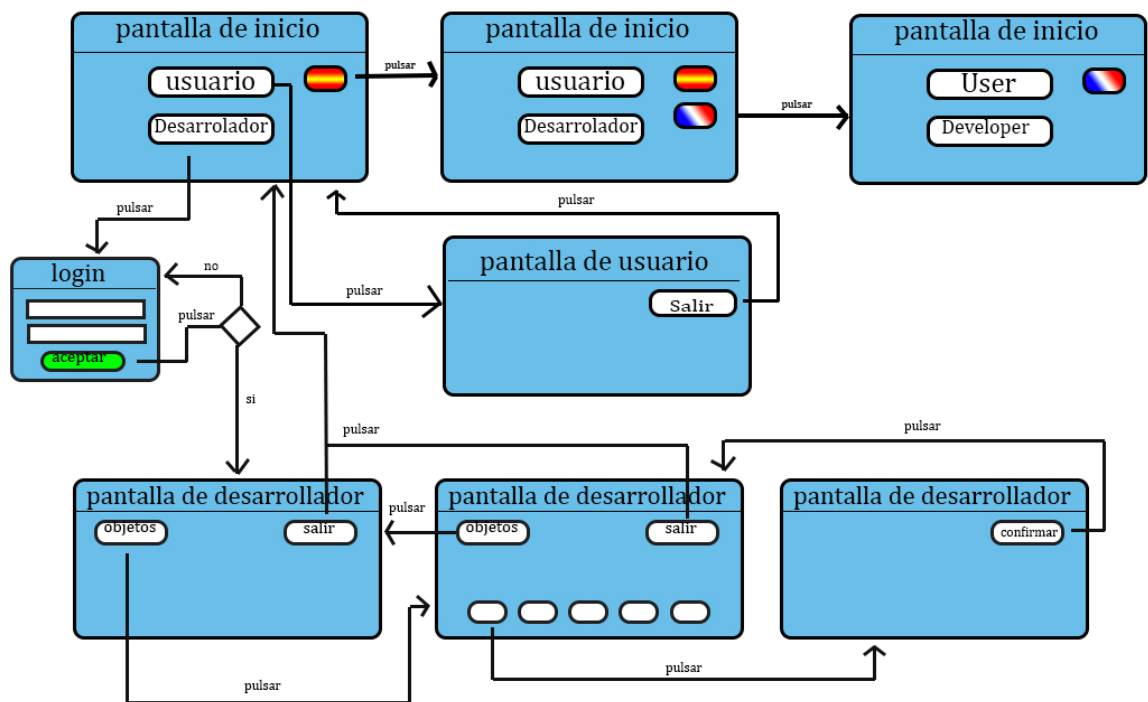


Ilustración 56

Tercer prototipo

En este prototipo no se cambia nada de la interfaz.

Cuarto prototipo

Gracias a la retroalimentación con las ejecuciones anteriores, nos percatamos de la necesidad de una ventana para la selección del yacimiento que queremos visualizar y la necesidad de un botón para borrar el elemento seleccionado en el perfil de desarrollador. Representado en la ilustración 57.

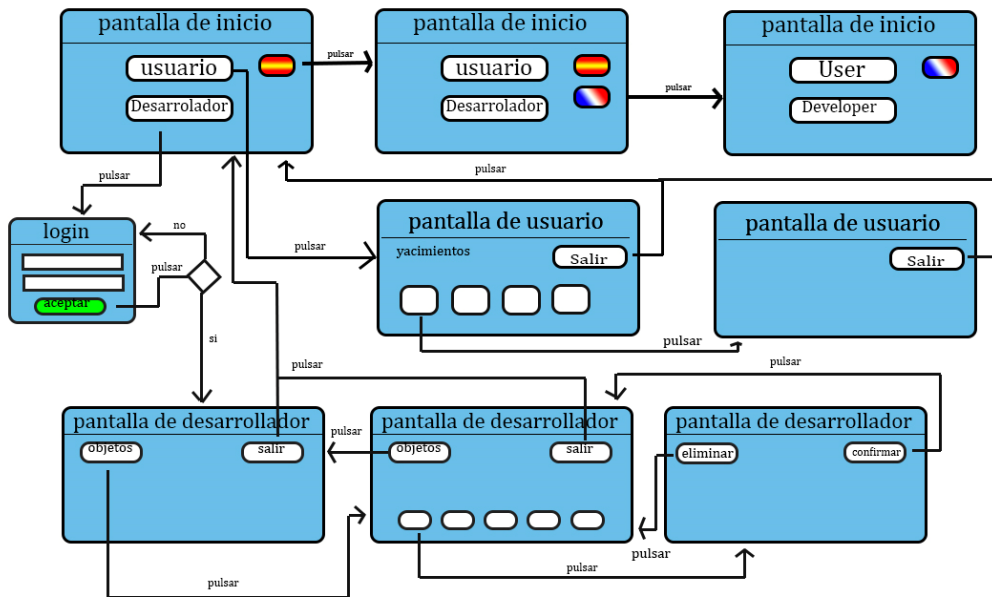


Ilustración 57

5.5.3. Diseño de la interfaz

Es muy importante que la interfaz sea lo más clara e intuitiva posible. Poniendo la mínima cantidad de botones y especificando bien la función de cada uno que si esta puesto, podemos lograr esos objetivos.

El diseño de las maquetas ha sido realizado con NinjaMock.

Cuando abrimos la aplicación encontramos el menú inicial, en el cual podremos cambiar de idioma y elegir entre modo usuario o desarrollador. Ilustración 58.



Ilustración 58

En el caso de que elijamos el modo usuario, nos redirigirá a una ventana donde podremos elegir el yacimiento que queramos ver. Ilustración 59.

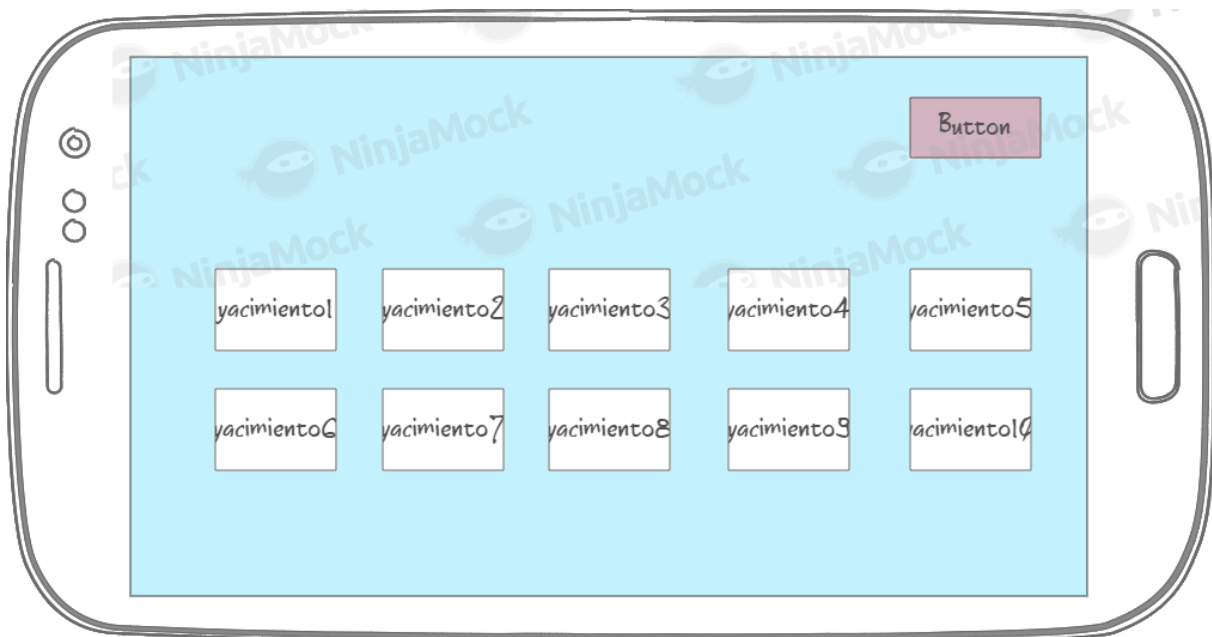


Ilustración 59

Una vez seleccionado el yacimiento, este aparecerá en nuestra pantalla.

En esta ventana podremos visualizar el yacimiento mientras caminamos por él. También tendremos la opción de consultar la historia del mismo. Ilustración 60 representa la visualización en modo usuario y la ilustración 61 representa el desplegable para el modo historia.

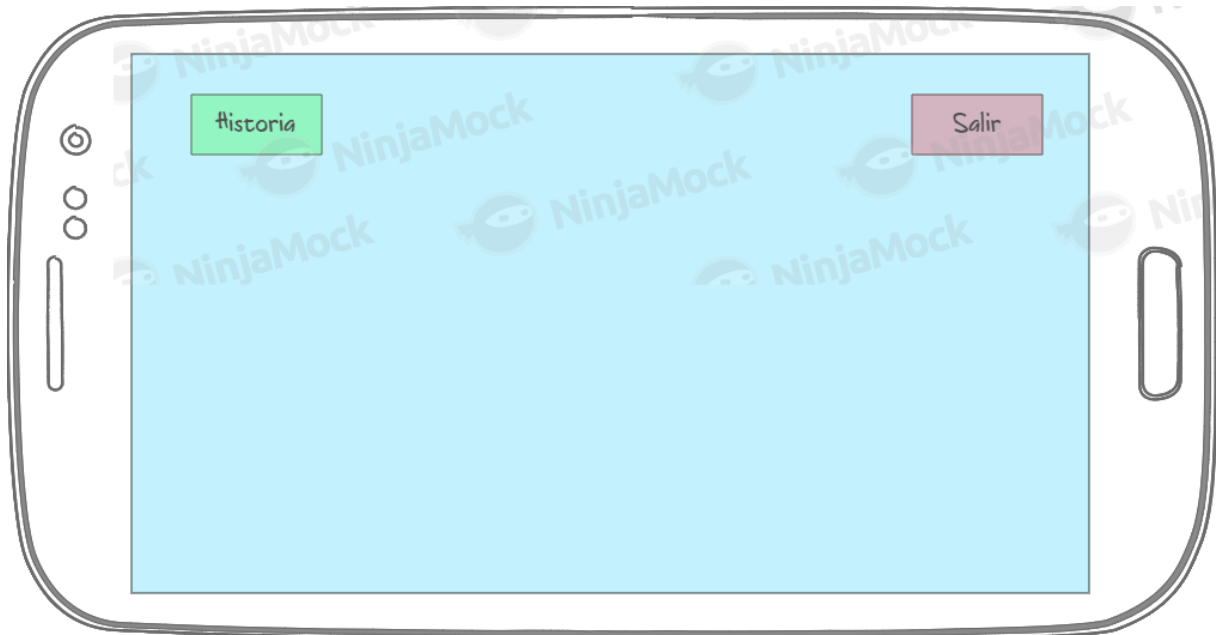


Ilustración 60

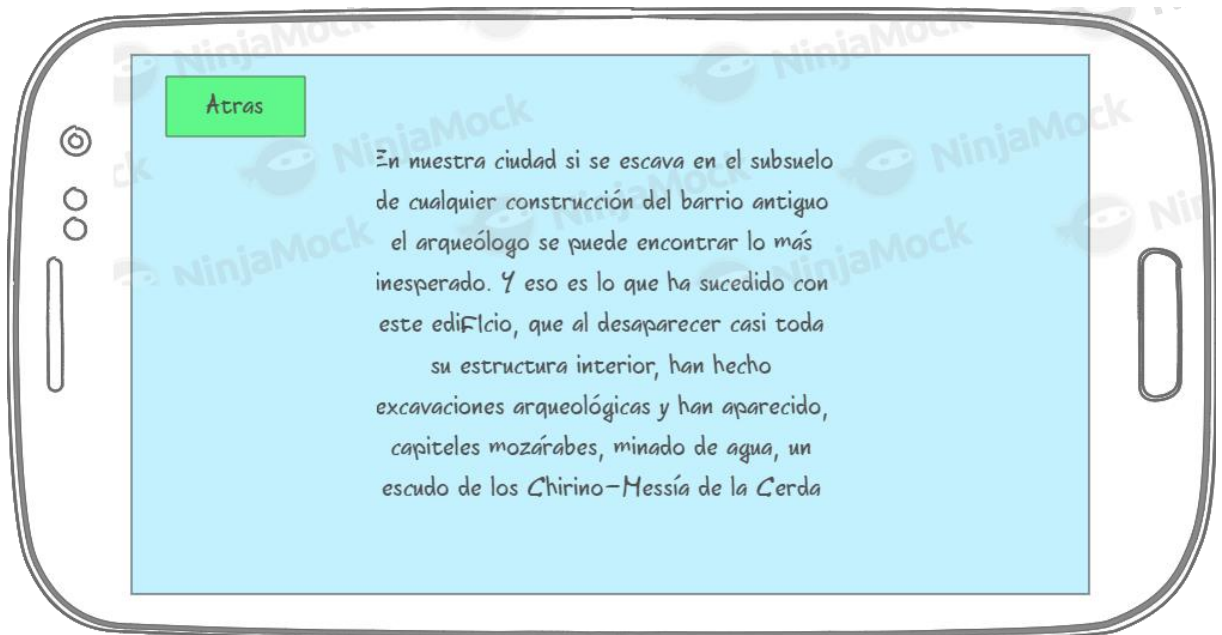


Ilustración 61

En el caso de que seamos desarrolladores, una vez pulsemos el botón de desarrollador nos aparecerá una ventana para introducir el usuario y la contraseña. Esto es así para que no todas las personas puedan modificar la composición de los yacimientos reconstruidos. Ilustración 62.



Ilustración 62

Si la autenticación es correcta, podremos acceder al modo desarrollador donde tendremos las herramientas necesarias para ubicar los diferentes objetos 3D en su lugar correspondiente en el yacimiento. Ilustración 63.

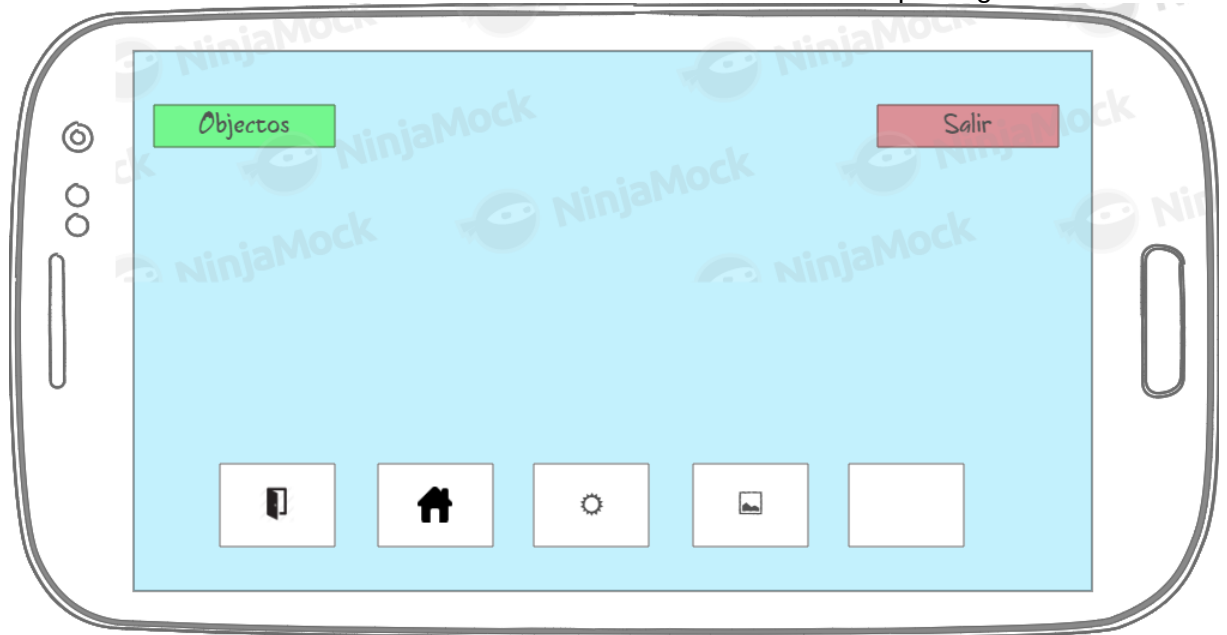


Ilustración 63

Una vez seleccionado el objeto que queremos ubicar, se cerraran todos los botones y aparecerán 2 nuevos que nos servirán para confirmar la ubicación del objeto o para borrarlo en el caso de que no nos interese. Ilustración 64.

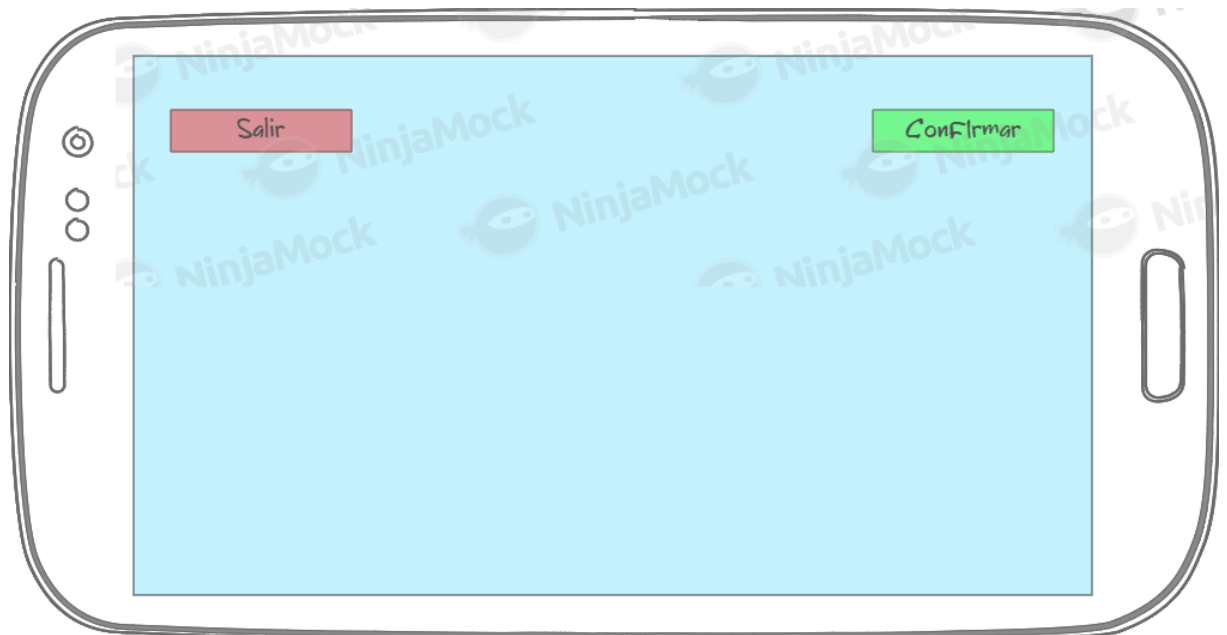


Ilustración 64

5.6 Implementación

La aplicación ha sido desarrollada sobre un paquete Unity, ofrecido por Google, en el cual podemos probar su aplicación en nuestro dispositivo.

Esta aplicación ofrece 2 escenas de prueba, correspondientes a dos tipos de aplicación.

Para el uso y entendimiento de este paquete se ha consultado la API que ofrece la plataforma Google ARCore en su página web oficial.

Dado que la explicación del funcionamiento de cada clase la podemos observar en dicha API, en este apartado podemos encontrar los UML de las clases modificadas o útiles para el desarrollo de la aplicación. Diagramas UML en las ilustraciones 65,66,67,68,69 y 70

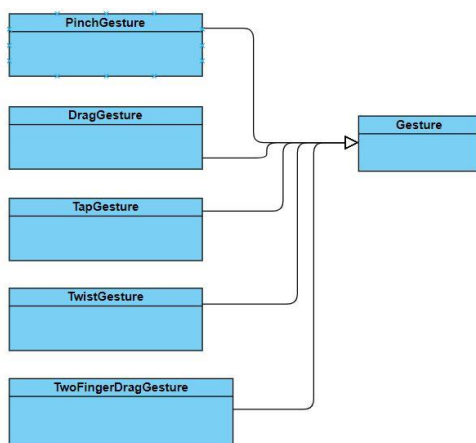


Ilustración 65

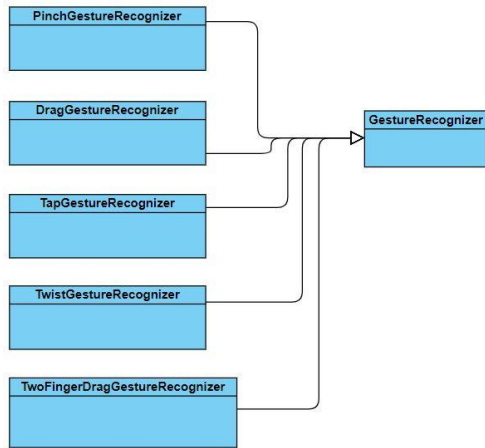


Ilustración 66

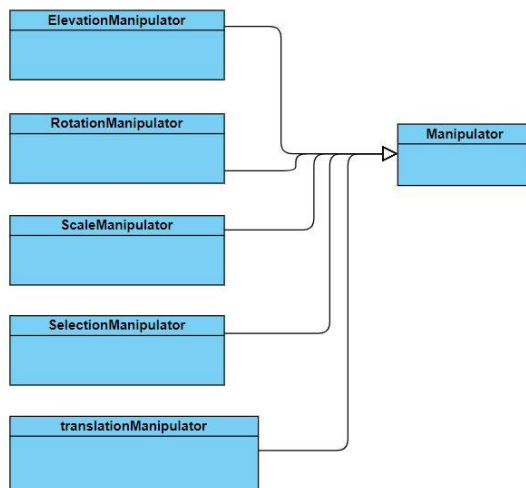


Ilustración 67

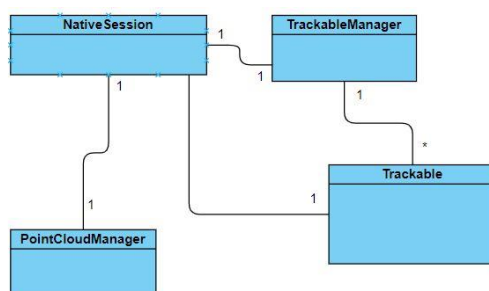


Ilustración 68

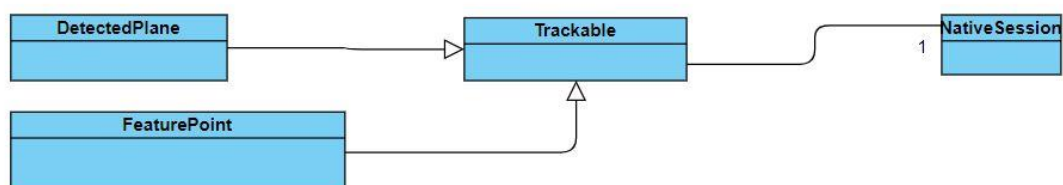


Ilustración 69

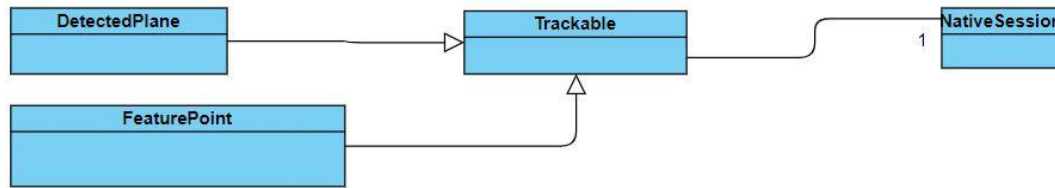


Ilustración 70

Para gestionar el uso de nuestra aplicación se han implementado dos clases significativas:

- Buttons: En esta clase podemos encontrar la funcionalidad de los botones de la aplicación.

- ShowHistoryCanvas: Función de mostrar el canvas para la historia del yacimiento.

- ExitHistoryCanvas: Función para ocultar canvas de historia.

- Update: Comprueba si hay algún objeto seleccionado para mostrar, o no, el botón de borrar objeto.

- ShowLoginCanvas/ExitLoginCanvas: Muestra/oculta el canvas para hacer login al intentar acceder al perfil de desarrollador.

- ShowDeveloperCanvas: Muestra las pantallas de uso de la aplicación. Dependiendo de si se ha intentado entrar en modo usuario o desarrollador, se activarán unos botones u otros, además de activar Raycast y la visualización de los planos detectados.

- ShowExitScrollBar/ExitScrollBar: Muestra/oculta la selección de los diferentes modelos 3D para generar en el yacimiento.

- ChangePrefab: Cambia el objeto a visualizar, dependiendo de la elección en la selección de los modelos 3D.

- Confirm: fija el modelo 3D al punto donde lo dejamos.

- Delete: Borra el modelo 3D seleccionado.

-ExitViewCanvas: salimos de cualquier pantalla de uso de la aplicación para volver al menú principal.

-Variables: Clase estática utilizada por todas las clases que la necesiten para conocer el estado de la aplicación en un momento determinado, y así, ajustar su comportamiento.

5.7. Uso de la aplicación

En los siguientes Workflow podemos apreciar el funcionamiento de la aplicación.

En la ilustración 71 podemos observar el funcionamiento del modo usuario de nuestra aplicación.

En este modo podemos consultar la historia del yacimiento reconstruido y visualizar el trabajo realizado en el modo desarrollador.

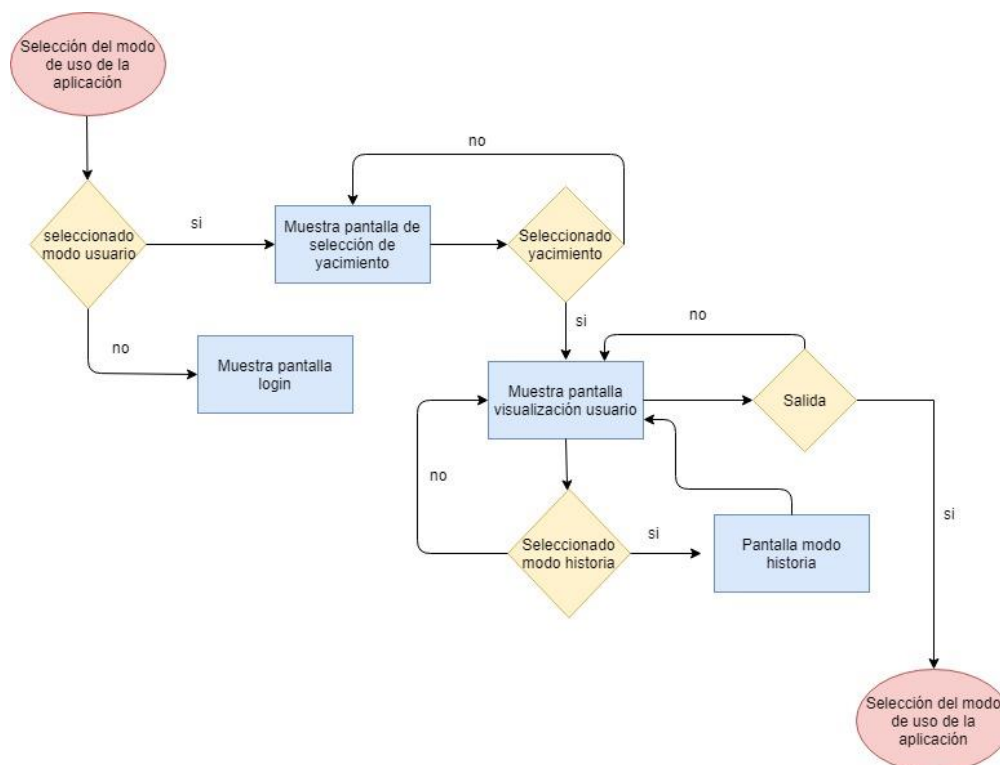


Ilustración 71

En la ilustración 72 podemos contemplar el funcionamiento del modo desarrollador.

En este modo tenemos la posibilidad de ubicar modelos 3D y reconstruir el yacimiento.

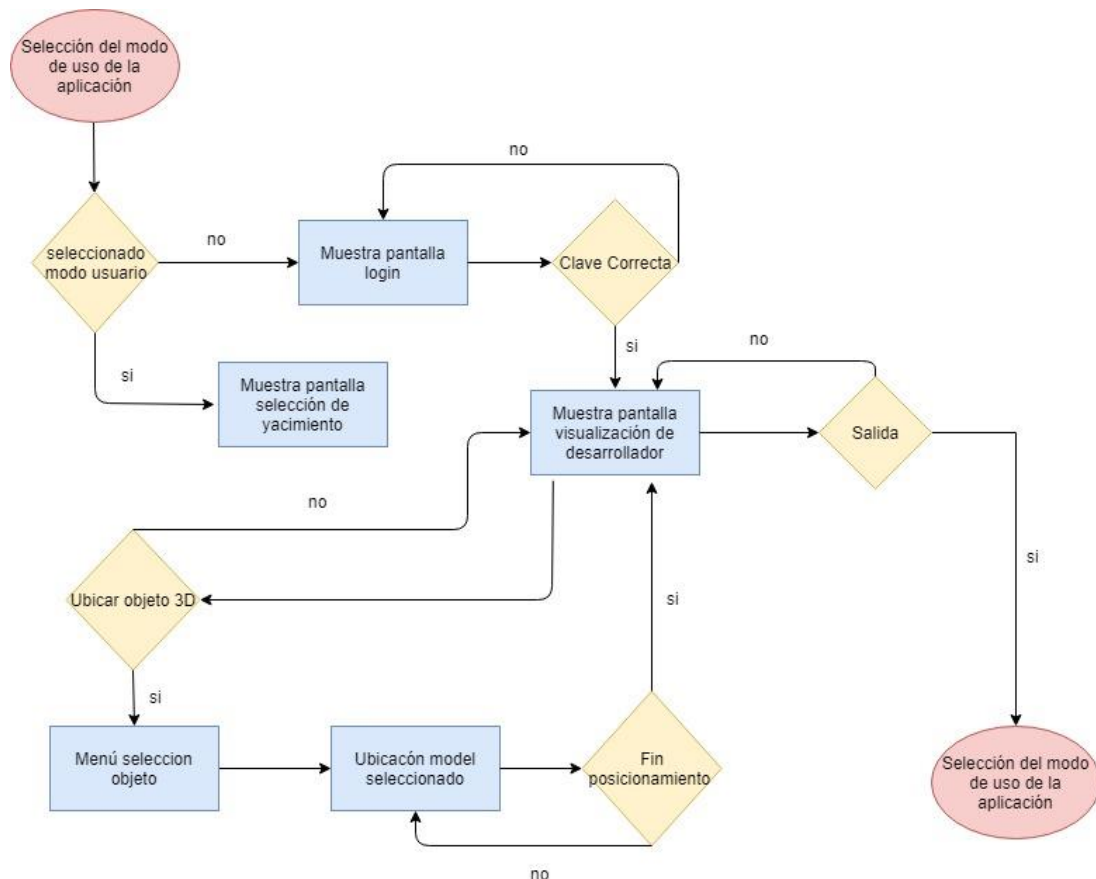


Ilustración 72

5.8. Pruebas

5.8.1. Pruebas a la tecnología de ARCore

5.8.1.1. -Maxima distancia generación de plano

Para esta prueba he colocado el dispositivo a diferentes alturas y he medido sobre la superficie las diferentes marcas a donde ha llegado la captación de puntos y la generación de plano. Podemos ver los resultados en la tabla 2.

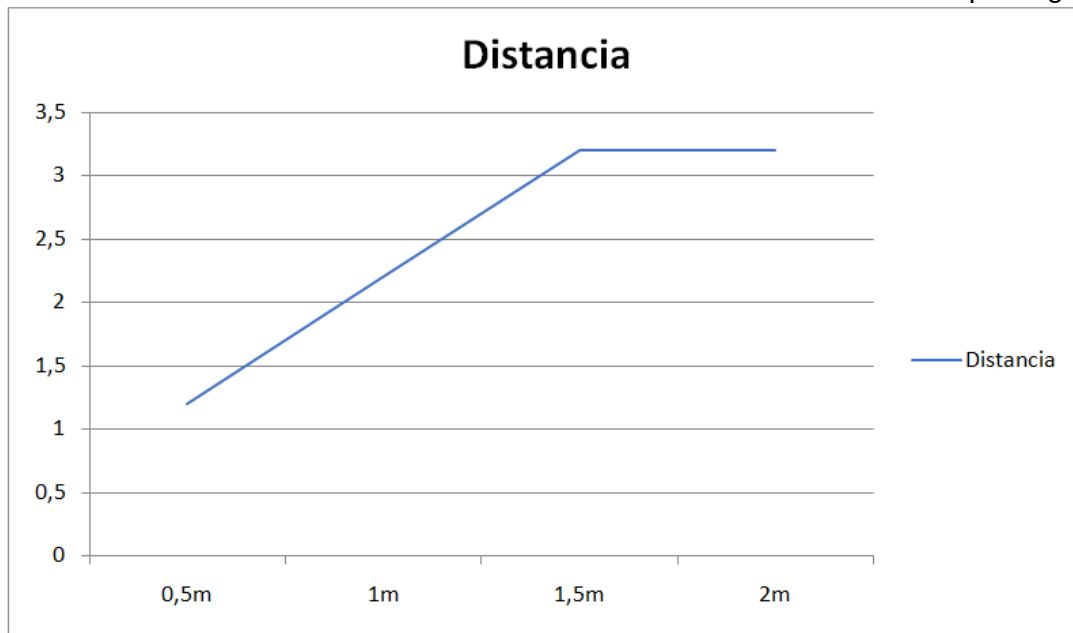


Tabla 2

Conclusión prueba: Podemos observar que conforme aumentamos la distancia a la que se encuentra el dispositivo del suelo, la distancia de captación de puntos aumenta linealmente, hasta que alcanza su máximo a los 3,2 m. Con esto podemos concluir que la distancia máxima a la que podemos capturar puntos es de 3.2 m.

5.8.1.2.-Detección de superficies

Esta prueba la realizaremos solamente en superficies lisas, ya que en una superficie que presente irregularidades, en condiciones óptimas de iluminación y a una distancia adecuada, ARCore es capaz de captar puntos y generar un plano sin ningún tipo de problema.

La tecnología ARCore no funciona bien con superficies lisas ya que para encontrar puntos clave y poder generar un plano, esta lo que hace es convertir la imagen captada por la cámara en una escala de grises y captar los puntos donde hay diferencia de tonalidad.

Para esta prueba he ido poniendo puntos de referencia en una superficie totalmente lisa.

-1 punto de referencia. Ilustración 73.

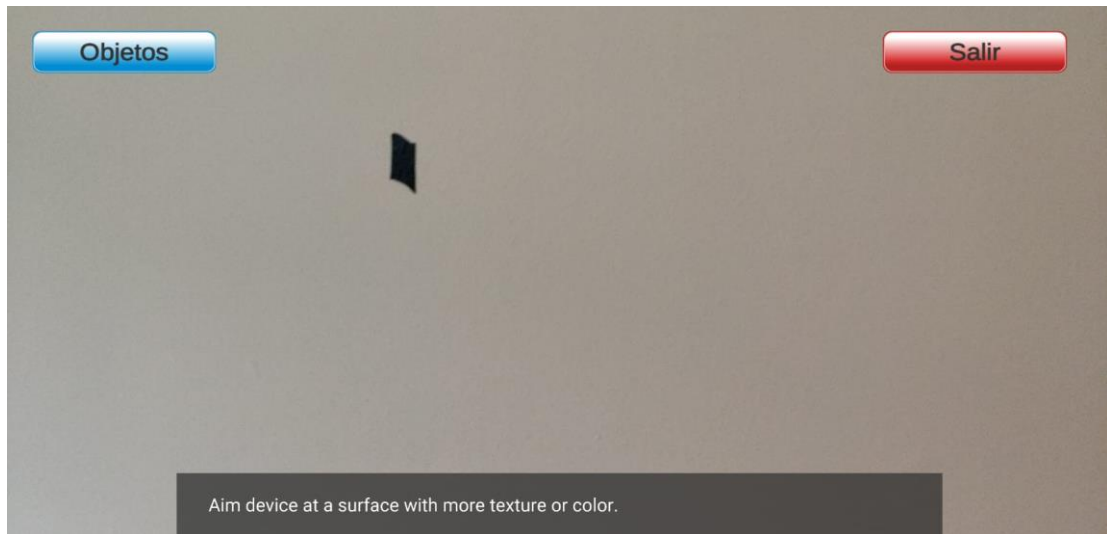


Ilustración 73

ARCore no es capaz de capturar puntos si en la superficie hay solo un objeto de referencia de un mismo color.

-2 puntos de referencia. Ilustración 74.

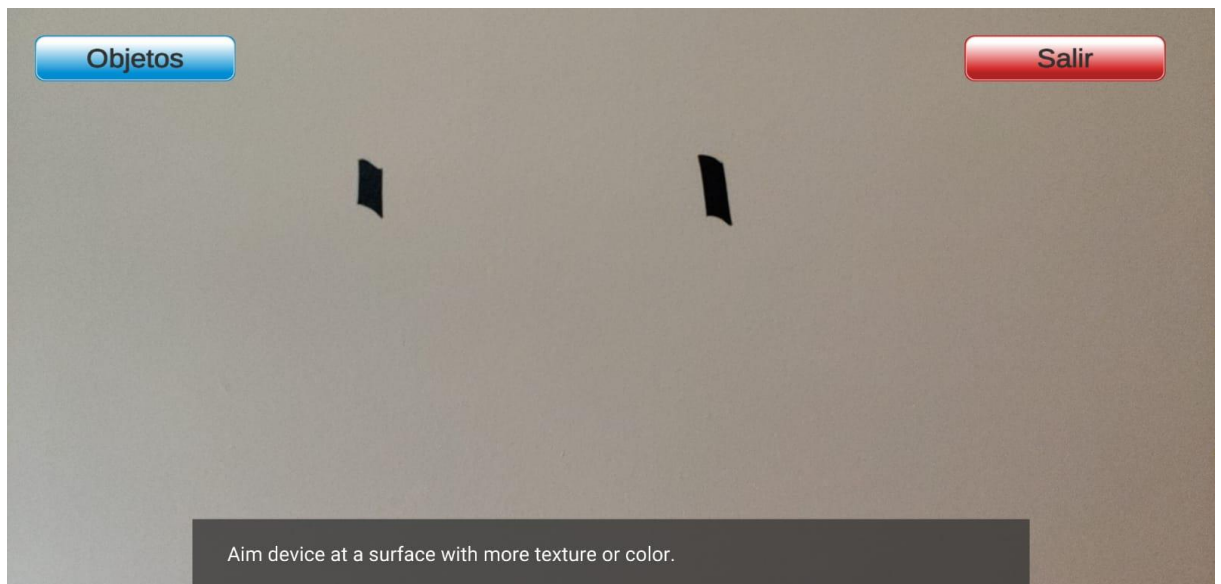


Ilustración 74

Mismo resultado que con un solo punto de referencia.

-3 puntos de referencia. Ilustración 75.



Ilustración 75

De nuevo mismo resultado.

-4 puntos de referencia. Ilustración 76.



Ilustración 76

ARCore es capaz de empezar a captar puntos, pero aun no identifica ningún punto clave para la generación de un plano.

-5 puntos de referencia. Ilustración 77.

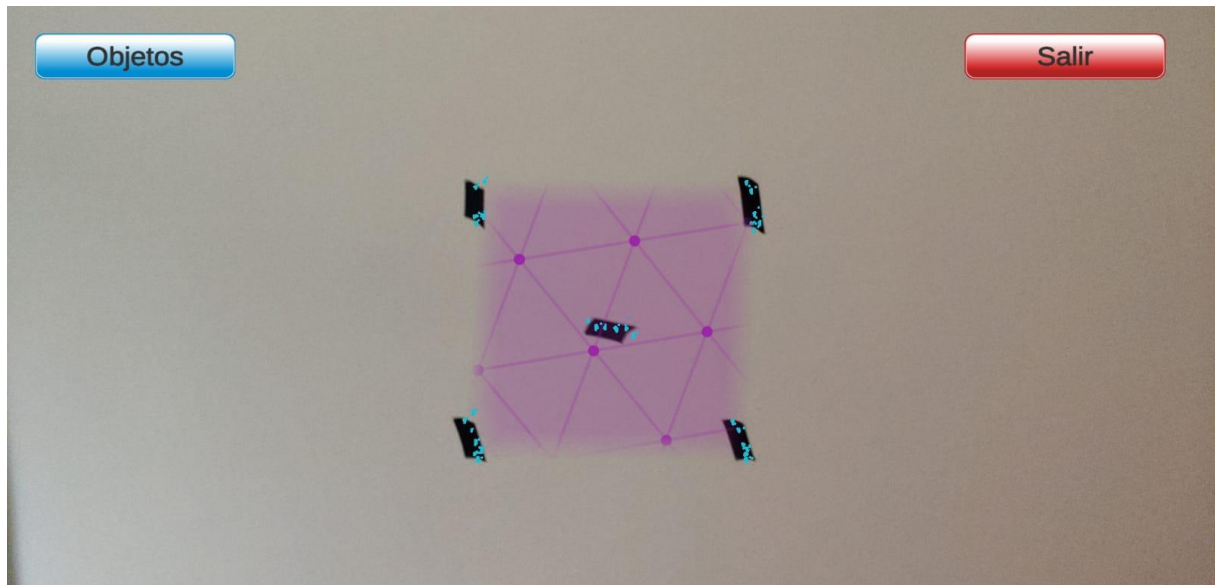


Ilustración 77

Con 5 puntos de referencia ARCore es capaz de conseguir la suficiente información como para crear un plano.

-Errores de las superficies totalmente lisas. Ilustración 78.

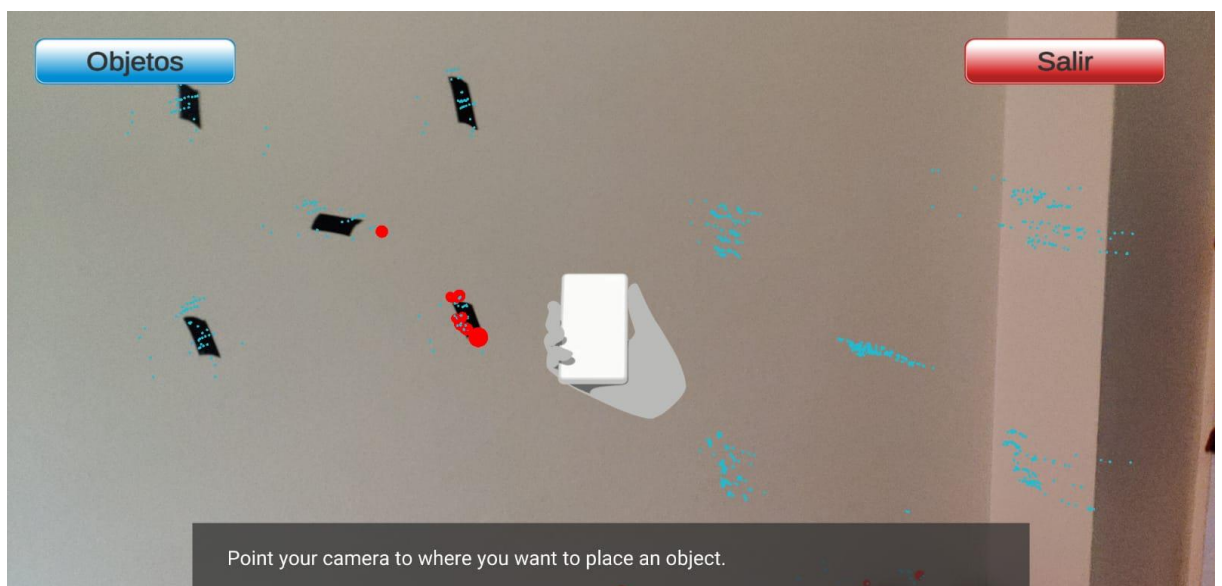


Ilustración 78

Al no tener más puntos de referencia, un problema muy habitual (basándome en las pruebas realizadas) es que el dispositivo pierde totalmente la ubicación del mismo y con que lo rotemos un poco, ARCore interpreta que el dispositivo se está trasladando continuamente y los puntos captados comienzan a trasladarse continuamente hasta que volvemos a captar otros puntos y ARCore deduce que el dispositivo ya está parado

Conclusión prueba Basándonos en los datos obtenidos de esta prueba, podemos concluir que son necesarios bastantes puntos de referencia (en el mundo real) para que ARCore sea capaz de generar un plano.

Debido al error comentado anteriormente, se pueden presentar problemas importantes debido a la desubicación del dispositivo.

5.8.1.3. -Rendimiento y temperatura

Para esta prueba he realizado un escaneo del terreno durante 10 minutos, tomando datos sobre el uso de la CPU, el estado de la memoria y la temperatura del dispositivo en intervalos de 2 minutos.

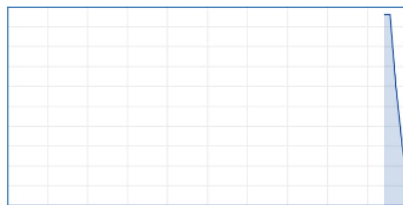
Para esto, las pruebas se han realizado a las 9:48 A.M con una temperatura de 19°C. Ilustración 79 muestra las condiciones en las que se realizó la prueba.

Las primeras pruebas las he realizado sin generar ningún modelo 3D, solamente con la visualización de los planos generados.



Ilustración 79

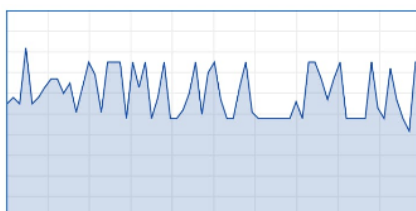
Estado inicial del dispositivo. Ilustración 80.



CPU		
Utilization	Current avg. freq.	Max frequency
9%	700 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	28 °C	
Common		
Battery temp	Available memory	
26 °C	2175 MB	

ilustración 80

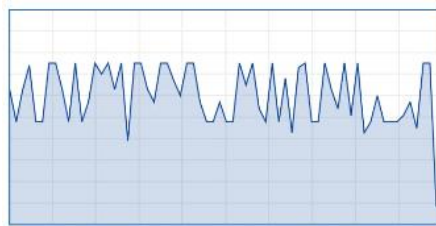
Pasados 2 minutos del inicio de la prueba. Ilustración 81.



CPU		
Utilization	Current avg. freq.	Max frequency
75%	1804 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	45 °C	
Common		
Battery temp	Available memory	
35 °C	1729 MB	

Ilustración 81

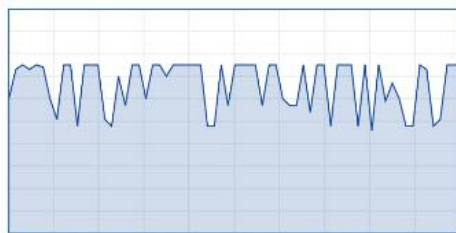
Pasados 4 minutos de ejecución. Ilustración 82.



CPU		
Utilization	Current avg. freq.	Max frequency
9%	700 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	43 °C	
Common		
Battery temp	Available memory	
35 °C	1704 MB	

Ilustración 82

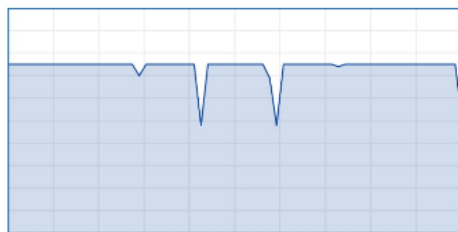
Pasados 6 minutos de ejecución. Ilustración 83.



CPU		
Utilization	Current avg. freq.	Max frequency
75%	1804 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	54 °C	
Common		
Battery temp	Available memory	
36 °C	1712 MB	

Ilustración 83

Pasados 8 minutos de ejecución. Ilustración 84.



CPU		
Utilization	Current avg. freq.	Max frequency
50%	1296 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	49 °C	
Common		
Battery temp	Available memory	
37 °C	1699 MB	

Ilustración 84

Pasados 10 minutos de ejecución. Ilustración 85.



CPU		
Utilization	Current avg. freq.	Max frequency
75%	1804 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	52 °C	
Common		
Battery temp	Available memory	
36 °C	1619 MB	

Ilustración 85

Resultados de la prueba en tablas 3 y 4.

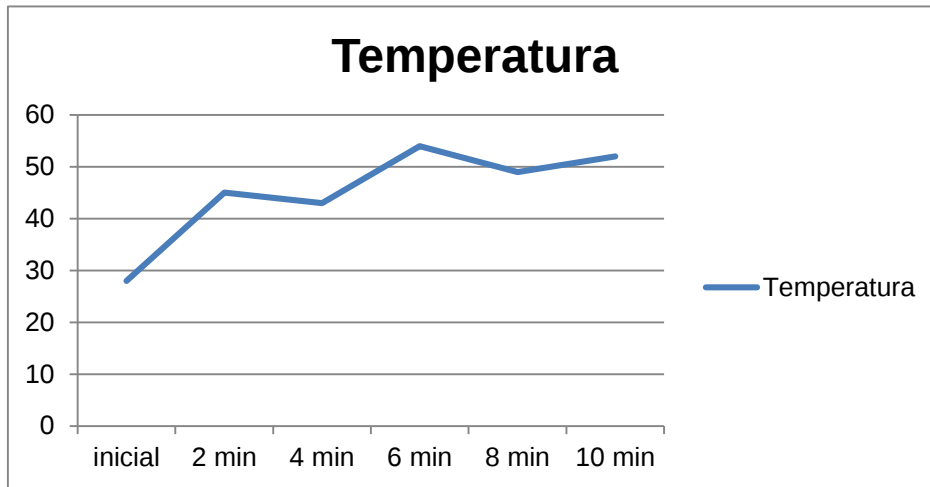


Tabla 3

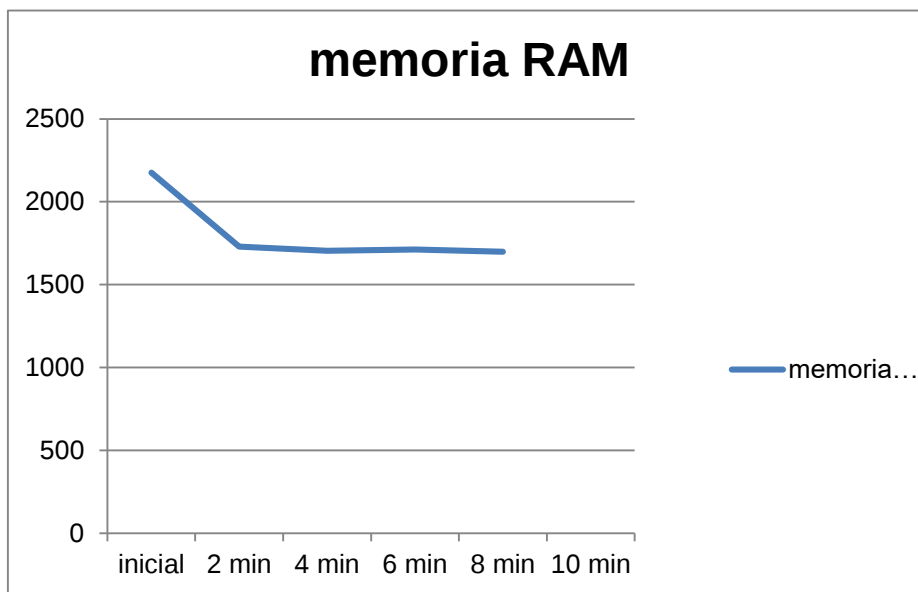


Tabla 4

Análisis de datos prueba

A lo largo de la prueba, podemos observar que el dispositivo se mantiene bastante estable.

***CPU**

Su pico de actividad es del 75%, independientemente del tiempo de escaneado.

Este pico lo alcanza cuando está reconociendo los puntos y enlazándolos con un plano ya existente o con uno nuevo.

***Temperatura**

Podemos observar que, debido al uso de la CPU, la temperatura aumenta rápidamente al inicio hasta los 6 minutos de uso. Después de estos minutos, la temperatura se mantiene.

***Memoria RAM disponible**

Al inicio de la aplicación encontramos un decremento significativo de la memoria RAM disponible, posiblemente causado por la ejecución de la aplicación y la reserva de recursos para la misma.

A lo largo de la ejecución observamos que la memoria RAM disponible disminuye muy lentamente.

Conclusiones

Podemos concluir que la aplicación, solo con el escaneado, a temperatura exterior relativamente baja y con condiciones de luz propicias, es muy estable.

El único problema que nos puede surgir sería quedarnos sin memoria, que, si el ritmo de decremento de la memoria es constante, podemos predecir que esta memoria la agotaremos en 173 min, es decir, en aproximadamente 3 horas de escaneo.

5.8.1.4. Ubicación

Tras varias pruebas de escaneo podemos decir que la ubicación del dispositivo, basándose únicamente en una nube de puntos, es algo inestable.

Gran parte de este problema es ocasionado por el desnivel del terreno, ocasionando problemas como el de la imagen.

En esta imagen podemos apreciar que los planos, generados anteriormente, han ascendido desde el suelo debido a que encontramos cierto desnivel en el pavimento.

Este error se ocasiona cuando nos alejamos de los planos captados con mayor antigüedad. Este error se suele dar bastante a menudo, en la prueba de 10 min del rendimiento se daba en cada instante que había un desnivel en el acerado.

5.8.1.5. Conclusiones sobre tecnología ARCore

Basándome en las pruebas realizadas Considero que la tecnología Google ARCore es aún una tecnología poco madura. Lleva poco tiempo en el mercado y aún faltan algunos aspectos que solucionar.

Uno de estos problemas es la incompatibilidad entre los entornos de desarrollo y la pre visualización en el dispositivo.

En este caso se ha usado Unity como plataforma de desarrollo. Esta plataforma cuenta con la posibilidad de ejecutar la aplicación en el dispositivo móvil directamente. Esto se llama “instant preview” pero, más que ayudarnos a depurar y comprender el funcionamiento de nuestra aplicación, es una fuente continua de problemas ya que si, por algún casual, iniciamos la aplicación con el dispositivo desconectado o con la pantalla bloqueada, esto causara que no se reconozca la cámara de nuestro dispositivo y que, por mucho que intentemos arreglarlo, la mayoría de las veces sea necesario restaurar el dispositivo.

Además de esto, con “instant preview” no podremos interactuar con la aplicación, ya que en el dispositivo se abrirá una ventana cuando se ejecute desde Unity. Esta ventana tiene un menú de opciones que impide que, al pulsar en la pantalla, Unity reconozca el gesto ya que se abrirá el menú desplegable de la pantalla del dispositivo.

Otro problema importante es la falta de información. Al llevar tan poco tiempo en el mercado, aun no se ha experimentado lo suficiente con ella y surgen errores que no tienen solución aparente. En foros propios de Unity y Google se plantean muchos de estos problemas, pero no hay ninguna solución. Por otra parte, la información que ofrece google sobre su tecnología es bastante escueta, diciéndote, como máximo, como funcionan algunas de sus clases y métodos.

Creo también que esta tecnología podría ayudar muchísimo en ámbitos educativos, debido a la facilidad absoluta de obtenerla y manejarla. Nos permite tener “frente a nosotros” cualquier objeto que queramos ver y estudiar.

Gracias a su reconocimiento de gestos, podemos separar las diferentes partes de un objeto, rotarlas, escalarlas y ver, con todo lujo de detalle, cualquier parte del mismo.

También podría ser muy útil en el mundo del entretenimiento, ya que, cualquier usuario podría fabricar, fácilmente, un juego que se adapte al sitio donde se encuentre, al permitirnos introducir modelos 3D basándose en el entorno que nos rodea esto no pone límites a los escenarios donde podemos jugar. (por ejemplo, un “pasadizo del terror” realizado en tu casa en pocos minutos).

5.8.2. Pruebas en la aplicación desarrollada

En estas pruebas nos centraremos en analizar los posibles problemas que pueden surgir en la aplicación desarrollada, muchos de ellos heredados de problemas de la tecnología ARCore.

Analizaremos como afectan cada uno de los parámetros de ARCore analizados anteriormente.

5.8.2.1. Influencia de la luz

Además de problemas de desplazamiento de los planos y puntos captados, como pasaba cuando detectamos una superficie lisa con pocos puntos de referencia, encontramos otro error que es la generación de planos a diferentes alturas en una misma superficie. Podemos ver un ejemplo en la ilustración 86.

Esto nos dificulta significativa la creación de un entorno de realidad aumentada ya que incapacita la labor del desarrollador para ubicar correctamente un modelo 3D.

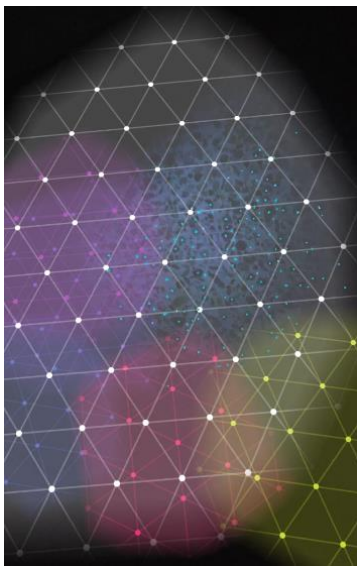


Ilustración 86

5.8.2.2. -Máxima distancia generación de plano

Debido a la “corta” distancia de captado de puntos dificulta el escaneado de superficies inaccesibles a más de 3 metros de distancia.

Una posible solución es, a la hora de modelar el objeto que queremos poner, establecer su centro de coordenadas en el punto más lejano que seamos capaces de escanear. Así, escaneada la superficie, al colocar el modelo 3D, este se generará a la distancia correcta desde el punto establecido.

El problema de este enfoque es que sería muy difícil ajustar escala, altura y posición del objeto. Sería necesario establecer estos parámetros de una

manera perfecta en el modelado del mismo. Podemos ver un ejemplo del sistema en la ilustración 87.

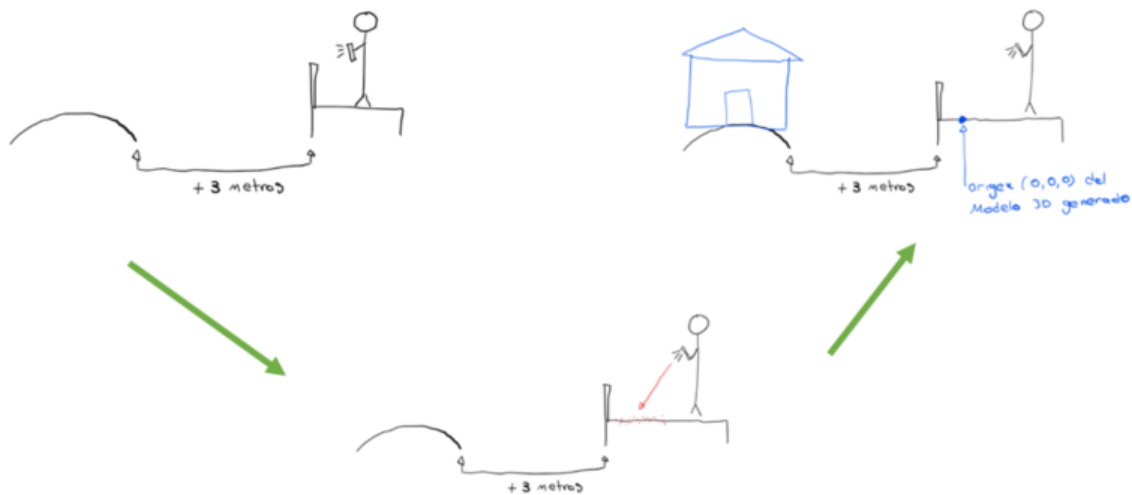


Ilustración 87

5.8.2.3.-Detección de superficies

Respecto a la detección de superficies, siempre que la hagamos con unas condiciones de luz aceptables, no encontraríamos ningún problema ya que en un yacimiento encontraremos superficies con suficientes irregularidades como para que ARCore sea capaz de captar y generar planos.

5.8.2.4.-Rendimiento y Temperatura

En esta prueba emularemos las mismas condiciones que para la prueba anterior, solo que esta vez visualizaremos un modelo 3D cada 30 segundos y analizaremos como influye este hecho en el uso de la CPU, la temperatura del dispositivo y la cantidad de memoria RAM disponible.

Pasados 2 minutos de ejecución. Ilustración 88.

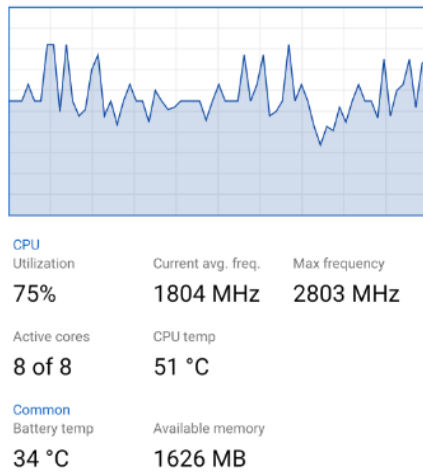


Ilustración 88

Pasados 4 minutos de ejecución. Ilustración 89.

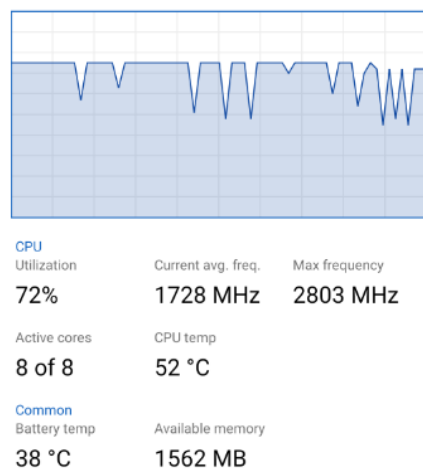
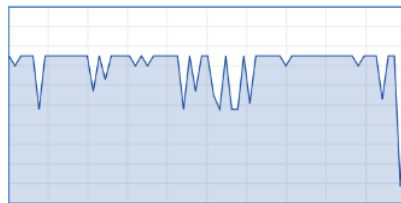


Ilustración 89

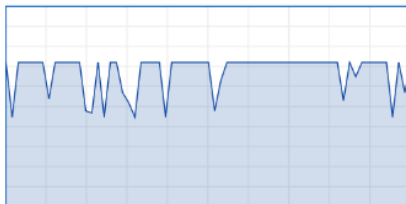
Pasados 6 minutos de ejecución. Ilustración 90.



CPU		
Utilization	Current avg. freq.	Max frequency
53%	1372 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	44 °C	
Common		
Battery temp	Available memory	
37 °C	1775 MB	

Ilustración 90

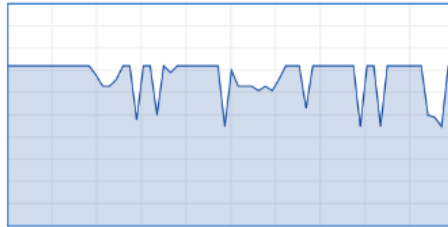
Pasados 8 minutos de ejecución. Ilustración 91.



CPU		
Utilization	Current avg. freq.	Max frequency
72%	1728 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	52 °C	
Common		
Battery temp	Available memory	
40 °C	1655 MB	

Ilustración 91

Pasados 10 minutos de ejecución. Ilustración 92.



CPU		
Utilization	Current avg. freq.	Max frequency
72%	1728 MHz	2803 MHz
Active cores	CPU temp	
8 of 8	53 °C	
Common		
Battery temp	Available memory	
41 °C	1522 MB	

Ilustración 92

Con estos datos recogidos podemos observar que la diferencia, entre tener y no tener modelos 3D visualizándose en la escena, es mínima.

Podemos ver que la temperatura de la CPU y de la batería es algo superior debido, posiblemente, al mayor esfuerzo que están desarrollando para visualizar los modelos 3D.

También podemos observar que la memoria disponible, tras 10 minutos, es algo menor que la memoria que teníamos disponible sin la visualización de modelos 3D.

Con esto podemos deducir que podremos visualizar un yacimiento extenso sin excesivo problema de temperatura ni de memoria.

Lo que si causaría problema sería la visualización de las mallas de triángulos que componen los planos y los edificios, que esto haría que la aplicación no fuese todo lo fluida que nos gustaría.

5.8.2.5. -Ubicación

Debido a los problemas de ARCore para mantener la altura y ubicación de los planos generados, se hace muy difícil colocar cualquier objeto teniendo otros ya colocados de referencia.

ARCore puede captar puntos a una misma distancia de la superficie y generar un plano. A medida que avancemos ARCore sigue capturando puntos y aumentando la extensión del plano añadiendo esos puntos que acabamos de captar.

El problema de esto es que los primeros puntos captados del plano, al alejarnos, cambian de ubicación y el plano se modifica, de forma que, si tenemos 2 modelos 3D ubicados en este plano, uno al inicio y otro al final, nunca estarán a la misma altura, ya que el plano se ajusta al suelo, conforme vamos acercándonos a los puntos que forman parte de él. Por este motivo, sería un problema tener un plano demasiado grande, ya que, aunque “teóricamente” los modelos estarían a la misma altura, su visualización sería errónea hasta el momento que nos acercásemos a cada uno de ellos.

Este hecho dificulta la creación y visualización de un entorno de realidad aumentada, en este caso, un yacimiento arqueológico.

En el caso de la ilustración 93, el plano rosa (el de los puntos más cercanos a la cámara) fue captado a la misma altura que el plano morado que se encuentra debajo, solo que al alejarnos del primer plano captado, este cambia su altura y cualquier objeto que este sobre este plano, también cambia dicha altura con él.



Ilustración 93

Sin embargo, como podemos ver en la ilustración 94, observamos que ambos planos vuelven a su misma altura una vez que nos acercamos al plano que ha perdido su referencia.



Ilustración 94

5.8.2.6 -Visualización

Dado que la imagen que capta la cámara se usa como background de nuestro canvas, no podremos crear una sinergia entre los objetos reales de yacimiento y nuestra reconstrucción.

Un ejemplo de esto sería la imposibilidad de crear un patio y utilizar las columnas reales existentes en el mismo. En el yacimiento reconstruido existe este ejemplo y la única solución sería crear el edificio sin una de las paredes.

En la ilustración 95 podemos observar este problema. La ilustración 96 representa el estado actual del patio con las columnas.



Ilustración 95



Ilustración 96

Otro ejemplo de este problema surge en la fachada del edificio, podemos apreciarlo en las ilustraciones 97 y 98.



Ilustración 97



Ilustración 98

Este problema también se nos presenta con elemento colocados en diferentes habitaciones, en el ejemplo presentado, he colocado dos modelos 3D en una habitación y me he salido de ella, el resultado ha sido que se puede ver todos los modelos 3D a través de las paredes. Podemos observarlo en la ilustración 99.

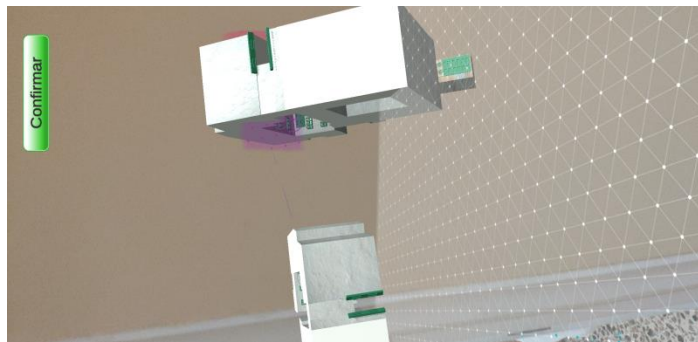


Ilustración 99

5.8.2.7.-Reconocimiento de los gestos

Pese a que la aplicación proporciona reconocimiento de gestos (trasladar, rotar, elevar y escalar) su comportamiento no es fiable al 100%.

Algunos problemas que se han presentado a la hora de transformar los modelos 3D usando el reconocimiento de gestos son:

Reconocimiento de gesto de elevar (tocando la pantalla con 2 dedos y desplazándolos hacia el extremo superior), el de rotar (tocando la pantalla con 2 dedos, manteniendo uno fijo y moviendo el otro) y el de escalar (tocando la pantalla con 2 dedos y moviéndolos los dos a la vez). Este movimiento no suele detectarlo con claridad ya que es muy similar al de rotación y escalado. La

interpretación que hace la aplicación de estos gestos es algo aleatoria ya que la mayoría de las veces confunde los movimientos y ocasiona modificaciones no deseables en el modelo 3D, dificultando así la labor del desarrollador .

Posibles soluciones:

-establecer un sistema de modificación de los modelos 3D basado en botones en vez de en gestos.

-establecer la rotación, elevación y escalado del modelo antes de visualizarlo, en la etapa de modelado.

5.9. YACIMIENTO RECONSTRUIDO

Para este prototipo he escogido la reconstrucción del yacimiento de santa clara (Úbeda).

5.9.1. Historia

El solar que actualmente ocupa el Antiguo Cuartel de Santa Clara, encierra un importante relleno arqueológico, que engloba restos de época prehistórica, romana, medieval islámica, medieval cristiana, moderna y contemporánea.

PERIODO I: La ocupación prehistórica.

El registro arqueológico nos informa de que, al menos, desde inicios del III milenio a. n. e. la ocupación prehistórica se extendía fuera de los límites del antiguo Alcázar.

PERIODO II: La ocupación romana. Siglo IV D.N.E.

Tras esta ocupación prehistórica se constata un largo vacío hasta época islámica que pueden fecharse entre los siglos XII y XIII. No obstante, se han recuperado algunos productos cerámicos de época iberorromana.

PERIODO III: La ocupación islámica. Siglos XII-XIII D.N.E.

En 1233 se toma de la ciudad de Ubbadat al'arab por parte de las tropas de Fernando III. Los restos arqueológicos documentados en la parcela de época musulmana se sitúan a finales del siglo XII, principios del XIII.

PERIODO IV. La ocupación cristiana. Siglos XIII-XV D.N.E.

Una vez conquistada la ciudad por las tropas cristianas se produce la ocupación de las viviendas, reformándolas, reutilizando los muros de carga, y las adaptándolos a sus costumbres.

Al igual que para la etapa anterior, las estructuras las podemos agrupar en espacios de habitación, estructuras de carácter productivo y fosas de planta circular excavadas en el sustrato geológico.

PERIODO V. La ocupación moderna. Siglos XVI-XVIII D.N.E.

Palacio de los Chirinos Narváez, siglo XVI.

La construcción de nueva planta del palacio en el siglo XVI por parte de la familia Chirinos - Narváez implicó la agregación y demolición de varias viviendas de la manzana.

Palacio del Conde de Gavia, siglo XVIII.

El origen del Palacio del Conde de Gavia hacia principios del siglo XVIII. Se construiría sobre la edificación anterior que, al menos desde el siglo XVI, estaría delimitando la mitad Sur de la Plaza de Santa Clara.

PERIODO VI. La ocupación contemporánea. Siglos XIX-XXI D.N.E. Declive del edificio palatino, siglo XIX. Cuartel de la Guardia Civil, siglo XX.

La adquisición del Palacio del Conde Gavia por el Ayuntamiento de Úbeda en el año 1926 y su posterior cesión al Estado con destino a Cuartel de la Guardia Civil, supuso una nueva etapa de continuas remodelaciones del inmueble hasta su definitivo abandono.

Demolición del inmueble, siglo XXI.

Las escasas obras de mantenimiento y el abandono por ruina del Cuartel en el año 1998 significaron el inicio de un acelerado proceso de deterioro que culminó con la demolición incontrolada del edificio en el año 2006, a excepción de la fachada principal, el patio de columnas en planta baja y el muro perimetral este del edificio.

5.9.2. Edificios

El yacimiento a reconstruir cuenta con 3 edificios principales: El cuartel, las viviendas y los talleres.

Basándonos en los planos y las fotografías cedidas por el arqueólogo de Úbeda he conseguido reconstruir, lo más fielmente posible, los edificios que antes formaban parte de este antiguo cuartel de la guardia civil. Plano usado para la reconstrucción, ilustración 100.

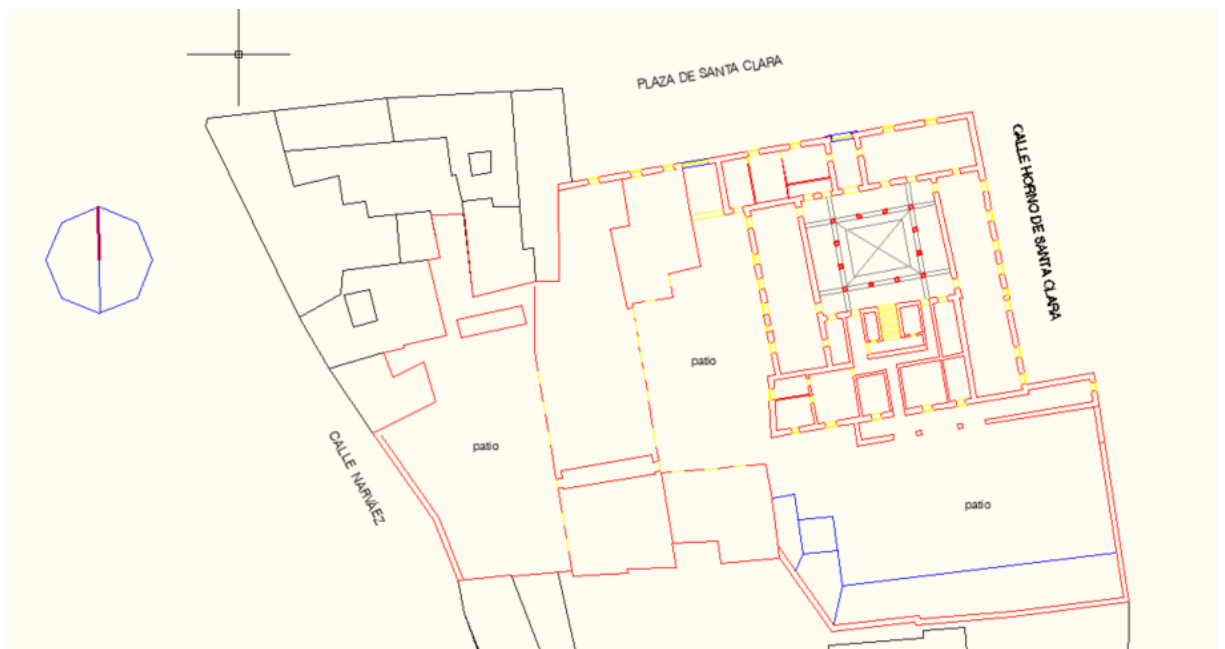


Ilustración 100

Imágenes usadas para la reconstrucción del cuartel, ilustraciones 101, 102,
103, 104, 105 y 106



Ilustración 101



Ilustración 102



Ilustración 103



Ilustración 104



Ilustración 105



Ilustración 106

Modelo 3D, realizado en Blender, del cuartel, ilustración 107 y 108.

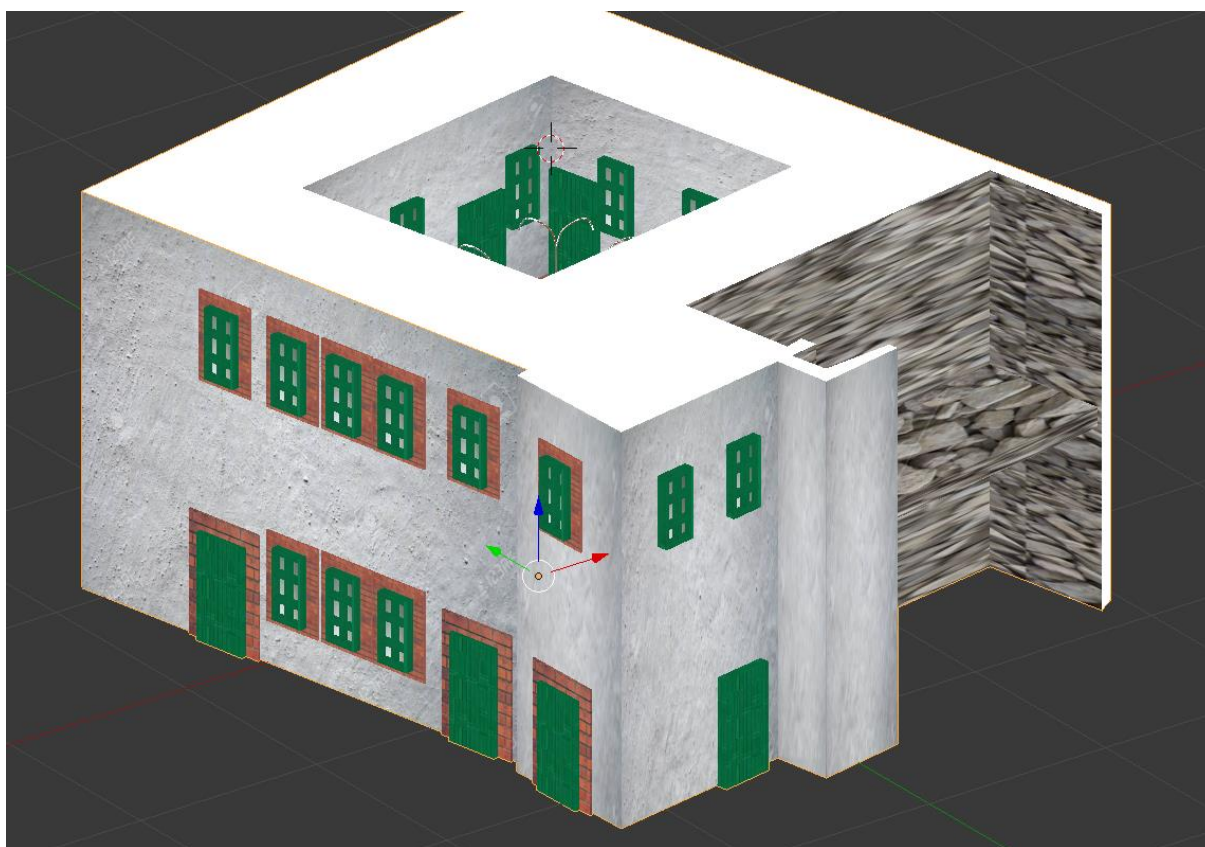


Ilustración 107

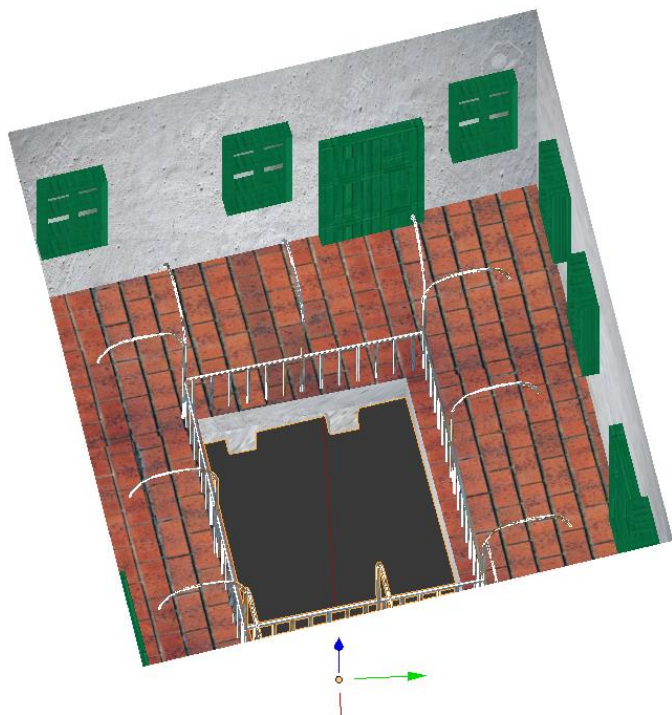


Ilustración 108

Imágenes usadas para la reconstrucción de las viviendas, ilustraciones 109 y 110.



Ilustración 109



Ilustración 110

Modelo 3D, realizado en Blender, de las viviendas, ilustración 111.

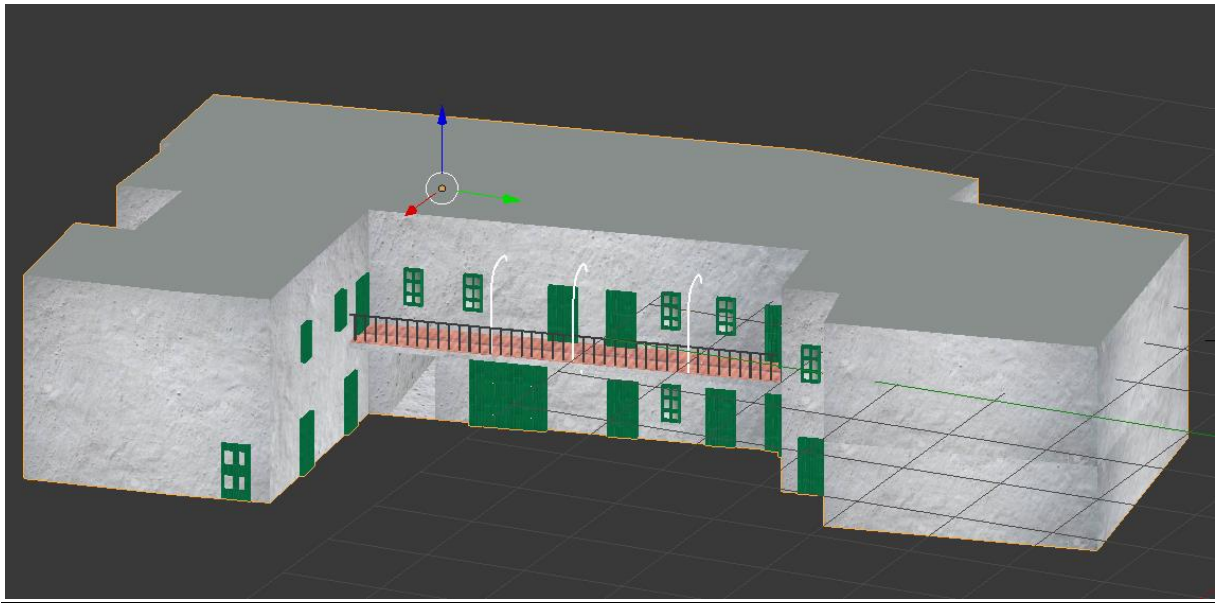


Ilustración 111

Imágenes usadas para la reconstrucción del taller, ilustraciones 112 y 113.



Ilustración 112



Ilustración 113

Modelo 3D, realizado en Blender, del taller. Ilustración 114.

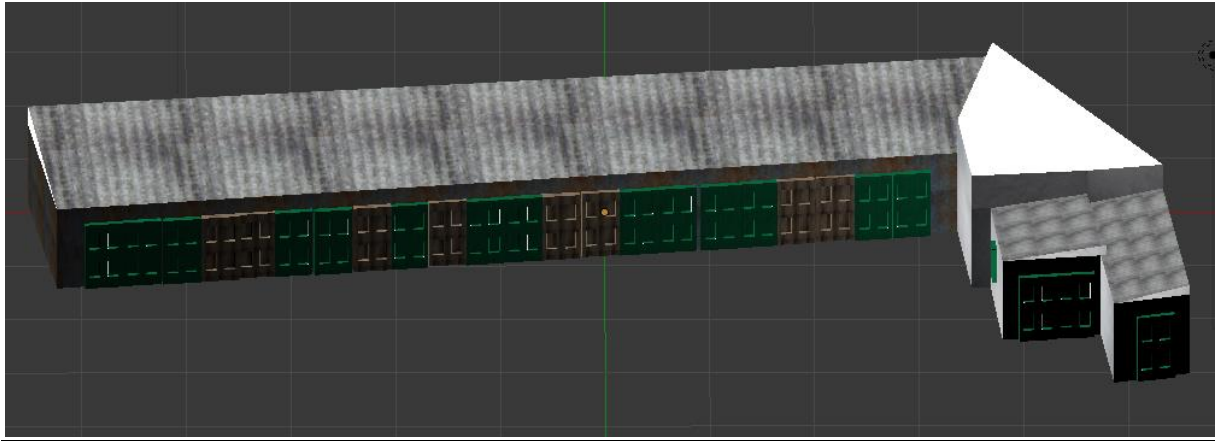


Ilustración 114

5.8.3. Yacimiento reconstruido

Para ubicar los modelos 3D en el yacimiento podemos hacerlo de dos maneras diferentes:

-Manera 1: escanear todo el yacimiento, generando los planos del mismo, e ir colocando cada modelo 3D independientemente en su lugar.

Esta manera de hacerlo no se la mejor elección, ya que: no se pueden ubicar bien los objetos debido a sus dimensiones y a la precisión de Google ARCore.

La mejor manera para poder reconstruir el yacimiento de esta manera sería colocar el objeto a una escala menor, rotarlo y ubicarlo. Por ultimo lo escalaremos.

Esta manera de reconstrucción la podemos observar en las ilustraciones 115, 116, 117, 118, 119, 120, 121 y 122.

-Manera 2: Modelar el conjunto de todo el yacimiento. En el yacimiento, únicamente, tendríamos que escanear el lugar donde se encuentra el origen de coordenadas de nuestro modelo 3D. Con lo cual nos ahorramos tiempo y memoria del dispositivo. Podemos ver un ejemplo en las ilustraciones 123 y 124.



Ilustración 115



Ilustración 116



Ilustración 117



Ilustración 118



Ilustración 119



Ilustración 120



Ilustración 121



Ilustración 122



Ilustración 123



Ilustración 124

6. Conclusiones

Respecto a la creación de una aplicación de reconstrucción de yacimientos arqueológicos usando Google ARCore podemos concluir que:

-Debido a algunos fallos, de la propia tecnología ARCore, se vuelve algo ardua la tarea de posicionamiento y modificación de los edificios que forman parte de un yacimiento arqueológico.

Como solución a este problema, en la fase de modelado, se pueden preparar todos los modelos 3D que forman parte del yacimiento arqueológico y ubicarlos en las posiciones donde se encontrarían en el yacimiento arqueológico, de tal forma que, al llegar al yacimiento, solamente necesitaríamos escanear el punto donde hemos colocado el origen de coordenadas de nuestros modelos enlazados y , al colocar el modelo en ese punto, se colocarían todos los modelos 3D en su lugar correspondiente.

Esto solucionaría el problema de la corta distancia de escaneo de puntos y de largas jornadas de reconocimiento del terreno.

-Hemos observado los problemas que tiene la tecnología al intentar enfocar a superficies lisas, como puede ser una pared o el cielo, de modo que, no podríamos enfocar al cielo, por ejemplo, para ver el techo de una casa, ya que el dispositivo perdería su ubicación con respecto a los planos ya generados y se perderían todos los modelos del yacimiento.

-Encontramos un gran problema con el hecho de que Google ARCore no se adapte bien a las paredes reales que podemos encontrar entre un objeto y otro, haciéndonos ver, por ejemplo, objetos que hemos colocado en una habitación, en otra.

Esto es un problema ya que no podríamos modelar partes de edificios e integrarlos con las partes que aún quedan de él en el mundo real.

Creo que este tipo de aplicación tiene futuro para yacimientos de los cuales no quede prácticamente nada, que las construcciones de los mismos no sean excesivamente altas, o que tengamos alguna manera de ubicar el dispositivo cuando enfoquemos al cielo para ver las partes altas de los modelos 3D.

También veo un gran futuro en el ámbito educativo, ya que, podríamos visualizar una ciudad entera en nuestra mesa. Este hecho no pone límites a la divulgación de la historia. Se podría modelar cualquier cosa que se nos ocurra: un mercado, una batalla o lo que se nos ocurra.

Tras probar Blender para modelar edificios creo que no es el mejor programa para realizar este cometido. Tiene una interfaz compleja, posiblemente difícil de

entender para usuarios que no estén familiarizados con programas de modelado en 3D.

Creo que una buena alternativa sería Autocad y utilizar Blender para asignar texturas y algunos elementos más creativos que no requieran medidas exactas.

7. Trabajo futuro

Dentro del trabajo futuro de este proyecto podemos destacar:

-Implementación de un modelo cliente-servidor para aljora los mapas y los modelos 3D en un servidor (por parte del desarrollador) y permitir al usuario descargarse el mapa que necesite, en función del yacimiento que quiera visitar

-Creación de base de datos de registro de desarrolladores

-Establecer medidas de seguridad de acuerdo con la conexión entre el cliente y servidor

-Estudio de la posibilidad de vincular la posición de los modelos 3D con coordenadas GPS. Esto permitiría un posicionamiento más fiel del modelo, independientemente de plano detectado.

Esta idea tendría que valorarse en función del objetivo de la aplicación, si es para visitas de yacimientos históricos, sería una buena opción, pero si es para educación, no sería tan buena idea, ya que privaríamos a cualquier docente o alumno de descargar los mapas en su aula, o casa, para poder observar cualquier yacimiento sin necesidad de acudir al lugar del mismo

Modelado más realista de los objetos de los yacimientos

Mejora del sistema de internacionalización, haciéndolo externo a la aplicación e introduciendo más idiomas

8. Bibliografía

-Gonzales Morcillo, Carlos; Vallejo Fernandez, David; Albusac Jimenez, Javier A.; Costro Snachez, José Jesús: *Realidad aumentada. Un enfoque Practico con ARToolKit y Blender*

-Ruiz Torres, David: *Realidad aumentada y Patrimonio cultural: Nuevas perspectivas para el conocimiento y la difusión del objeto cultural*

-Sanchez Riera, Alberto: *Tesis doctoral: Evaluacion de la tecnología de realidad aumentada móvil en entornos educativos del ámbito de la arquitectura y edificacacion*

-Arreguin Gonzales, Jacqueline Fabiola [en linea]: *Tesina para obtener el título de licenciado en informática administrativa*

- <http://ri.uaemex.mx/bitstream/handle/20.500.11799/59266/Tesis+1-split-merge.pdf;jsessionid=DFB34E6359D6EAF6C61D96066BCABD5B4?sequence=3>

.Redondo Ernesto; sanchez, Alberto; Narcis Moya, Joaquín; Regot Joaquin; *La ciudad como aula digital. En señando urbanismo y arquitectura mediante mobile learning y la realidad aumentada, un estudio de viabilidad y de caso*

-Esclapés, Javier; Tejernia Daniel; Bolufer, Joaquin; Esquembre Marco Aurrlio: *Sistema de realidad Aumentada para la musicalización de yacimientos arqueológicos*

-ARCore:

Disponible en: <https://developers.google.com/ar/>

- *Uso y aplicaciones de la realidad aumentada.*

Disponible en : <http://totemcat.com/usos-y-aplicaciones-de-la-realidad-aumentada/>

-*Uso de la realidad aumentada en las aulas:*

Disponible en: <https://www.educaciontrespuntocero.com/noticias/usos-realidad-aumentada-aulas/91867.html>

- Información en línea

(Educ@contic, *el uso de las TIC en las aulas*, 2012)

- Información En línea.

(Información obtenida de *Mahei, What's next?*)

-*Campos de aplicación de la realidad aumentada:*

Disponible en: [http://www.francisortiz.biz/francis-ortiz-
realidad-aumentada/campos-de-aplicacion-de-la-re](http://www.francisortiz.biz/francis-ortiz-realidad-aumentada/campos-de-aplicacion-de-la-re)

-*Arqueología y realidad aumentada.*

Disponible en: [https://evemuseografia.com/2018/04/23/arqueologia-y-
realidad-aumentada/](https://evemuseografia.com/2018/04/23/arqueologia-y-realidad-aumentada/)

-*Arqueología y técnicas digitales en educación patrimonial.*

Disponible en: <https://revistas.um.es/educatio/article/view/222511/174761>

-*Paquete Unity de Google ARCore.*

Disponible en: <https://github.com/artoolkit/arunity5>

ANEXO A: Manual de usuario

Para usar esta aplicación es necesario disponer de la aplicación “*Google ARCore*” instalada previamente en el dispositivo.

Una vez iniciada la aplicación encontraremos 3 botones:

-Cambio de idioma: presionándolo se desplegarán botones con los diferentes idiomas disponibles. Basta con presionar en el que queremos.

-Usuario: este botón nos redirige a una ventana donde encontraremos un catálogo de todos los yacimientos disponibles, en el elegiremos el que queramos visitar.

-Una vez seleccionado, la aplicación abrirá la cámara y el usuario deberá enfocar a un punto en concreto del yacimiento especificado al abrir el mapa.

-en esta vista encontramos un botón de salir, el cual nos llevara a la pantalla de inicio.

-Desarrollador: al presionar este botón aparecerá una ventana de login, donde los usuarios con permisos de desarrollador deberán ingresar su nombre y contraseña.

Una vez hecho esto y dándole a “aceptar” se redirigirá al usuario a la ventana de creación, en la cual podrá encontrar dos botones:

-objetos: al presionarlo se visualizará un desplegable donde encontrara todos los modelos disponibles para ese yacimiento.

-al presionar un modelo, desaparecerá todo lo anterior de la ventana y aparecerán dos botones:

-borrar: borra el modelo generado y nos redirige a la ventana anterior.

-Confirmar: ancla el objeto y nos redirige a la ventana anterior.

-salir: al presionarlo nos redirige al menú principal.