



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

VISUALIZACIÓN Y EDICIÓN DE MODELOS REPRESENTADOS POR NUBES DE PUNTOS UTILIZANDO REALIDAD AUMENTADA

Alumno/a: Ruiz García, M^a Dolores

Tutor/a: Rueda Ruiz, Antonio Jesús
Dpto.: Departamento de Informática

Tutor/a: Ogayar Anguita, Carlos Javier
Dpto.: Departamento de Informática

Julio, 2018



**Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática**

Don Antonio Jesús Rueda Ruiz y Don Carlos Javier Ogayar Anguita, tutores del Trabajo Fin de Master titulado: Visualización y edición de modelos representados por nubes de puntos utilizando Realidad Aumentada, que presenta M^a Dolores Ruiz García, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 4 Julio de 2018

El alumno:

Tutores:

M^a Dolores Ruiz García

Antonio Jesús Rueda Ruiz
Carlos Javier Ogayar Anguita

AGRADECIMIENTOS

Me gustaría dar las a mis amigos, profesores y familiares que me han apoyado durante estos meses de trabajo. En especial, a mis tutores Antonio y Carlos, y a Rafa.

También agradecer la paciencia y apoyo moral de mi pareja, por soportarme en los días y noches de duro trabajo.

GRACIAS A TODOS

Índice

Índice de ilustraciones.....	4
Índice de tablas	6
1. Introducción	7
2. Objetivos	11
3. Software y Dispositivos de desarrollo	12
3.1. Análisis del dispositivo Microsoft Hololens.....	12
3.2. Análisis y configuración del software.....	14
3.2.1. IDE Visual Studio.....	15
3.2.1. SDK de Windows 10 y emulador Hololens	16
3.2.2. Vuforia	17
3.2.3. Unity 3D.....	18
3.2.4. MixedRealityToolkit-Unity.....	23
4. Planificación del proyecto	24
4.1. Metodología de desarrollo.....	24
4.2. Historias de Usuario	26
4.3. Estimación de recursos y costes.....	28
4.4. Evolución del proyecto.....	33
5. Producción y desarrollo.....	36
5.1. Descarga, lectura y visualizado de la nube de puntos	36
5.1.1. Elementos esenciales para Hololens	38
5.1.2. Lectura y visualizado de la nube de puntos	39
5.1.3. Descarga de la nube	44
5.1.4. Uso de entorno y gesto “Tap”	44
5.1.1. Selección de nubes.....	47
5.1.2. Compatibilidad con PLY binario.....	48
5.2. Diseño y desarrollo de herramientas	51
5.2.1. Herramientas de sistema	53
5.2.2. Herramientas de nube.....	56
5.2.3. Herramientas de puntos	57
5.3. Sonidos	59
5.4. Problemas enfrentados.....	60
5.4.1. Incompatibilidades y actualizaciones del software de desarrollo	60
5.4.1. Intérprete innecesario entre MonoDevelop y Visual Studio	61

5.4.2.	Trabajando con varias escenas.....	62
5.4.3.	Trabas entre librerías Unity y librerías UWP (internet e hilos)	62
5.4.4.	Depuración y emuladores, Holographic Emulator y Hololens Emulator mienten 63	
5.4.5.	Paso de paquete de librerías: De HoloToolKit a MixedRealityToolKit	64
5.4.6.	Los Custom Shader no son estereoscópicos ni se incluyen en la solución de forma automática.....	64
5.4.7.	Contratiempos con el gesto Double tap.....	65
5.4.8.	Buscando a UiRayCastCamera.....	65
6.	Conclusiones.....	67
7.	Anexos.....	69
7.1.	Tarjetas de historias de usuario iniciales	69
7.2.	Tarjetas de historias de usuario finales.....	74
7.3.	Diagramas y grafos.....	81
7.3.1.	Diagramas de caso de uso	81
7.3.2.	Storyboard de toda la interfaz de usuario de la aplicación.....	82
7.4.	Manual de usuario	83
7.4.1.	Escaneado de entorno.....	83
7.4.2.	Elección de una nube de puntos	84
7.4.3.	Área de trabajo y Herramientas	87
	Bibliografía	92

Índice de ilustraciones

<i>Ilustración 1.1 - Videojuego "Paranormal Activity: The lost soul"</i>	7
<i>Ilustración 1.2 - Videojuego "Pokemon GO"</i>	7
<i>Ilustración 1.3 - Simulador nave espacial con realidad virtual</i>	8
<i>Ilustración 1.4 - Aplicación de apoyo quirúrgico con realidad aumentada</i>	8
<i>Ilustración 1.5 - Aplicación de control de calidad con realidad aumentada</i>	9
<i>Ilustración 1.6 - Nube de puntos de un puente</i>	9
<i>Ilustración 3.1 - Microsoft Hololens</i>	12
<i>Ilustración 3.2 - Placa base de Hololens</i>	13
<i>Ilustración 3.3 - Parte óptica de Hololens</i>	13
<i>Ilustración 3.4 - Sensores Hololens</i>	14
<i>Ilustración 3.5 - Módulos Visual Studio 2017</i>	15
<i>Ilustración 3.6 - SDK Windows 10</i>	16
<i>Ilustración 3.7 - Vista principal de Emulador Hololens</i>	16
<i>Ilustración 3.8 - Descarga instalador de emulador Hololens</i>	16
<i>Ilustración 3.9 - Botón "Tools" de emulador Hololens</i>	17
<i>Ilustración 3.10 - Panel parámetros de simulación</i>	17
<i>Ilustración 3.11 - Configuración Unity: Cambio de plataforma</i>	19
<i>Ilustración 3.12 - Configuración Unity: Características del proyecto</i>	19
<i>Ilustración 3.13 - Configuración Unity: XR Settings</i>	20
<i>Ilustración 3.14 - Configuración Unity: Other Settings</i>	20
<i>Ilustración 3.15 - Configuración Unity: Capabilities</i>	21
<i>Ilustración 3.16 - Configuración Unity: Quality Settings</i>	21
<i>Ilustración 3.17 - Configuración Unity: Emulador Hololens</i>	22
<i>Ilustración 3.18 - Configuración Unity: Configuración de emulador Hololens</i>	22
<i>Ilustración 3.19 - Paquete MRTK</i>	23
<i>Ilustración 3.20 - MRTK - Importando paquete a Unity</i>	23
<i>Ilustración 4.1 - Diagrama de casos de uso: Obtención de la nube de puntos</i>	26
<i>Ilustración 4.2 - Diagrama de casos de uso: Edición de la nube de puntos</i>	27
<i>Ilustración 4.3 - Gráfico: Velocidad de proyecto</i>	34
<i>Ilustración 4.4 - Gráfico: Horas por iteración</i>	34
<i>Ilustración 4.5 - Dedicación por fases</i>	35
<i>Ilustración 5.1 - Ejemplo de nube de puntos con Point Cloud Free Viewer</i>	36
<i>Ilustración 5.2 - Ejemplo de jerarquía de la nube de puntos con "Point Cloud Free Viewer"</i>	37
<i>Ilustración 5.3 - Elementos imprescindibles para Hololens</i>	38
<i>Ilustración 5.4 - Ejemplos de cabeceras de nubes de puntos PLY</i>	39
<i>Ilustración 5.5 - Seudocódigo: Lectura de cabecera PLY</i>	40
<i>Ilustración 5.6 - Seudocódigo: Hilos para lectura de puntos</i>	41
<i>Ilustración 5.7 - Seudocódigo: Estructura de la clase PointCloud</i>	42
<i>Ilustración 5.8 - Seudocódigo: Lectura de datos de puntos en ASCII</i>	42
<i>Ilustración 5.9 - Seudocódigo: Reparto de puntos y generación de objeto nube</i>	43
<i>Ilustración 5.10 - Ejemplo de nube de puntos cargada</i>	44
<i>Ilustración 5.11 - Comparativa de mala (izq.) y buena (der.) ubicación de modelos</i>	45
<i>Ilustración 5.12 - Seudocódigo: Calcula centro del entorno</i>	46
<i>Ilustración 5.13 - Interfaz de inicio</i>	46
<i>Ilustración 5.14 - Ejemplo de escaneado de entorno e interfaz de inicio</i>	47
<i>Ilustración 5.15 - Script PHP que devuelve lista archivos PLY</i>	48
<i>Ilustración 5.16 - Menú selección de nube</i>	48
<i>Ilustración 5.17 - Ejemplos de menú de carga de nube</i>	49
<i>Ilustración 5.18 - Ejemplo de nube de puntos en PLY binario</i>	51

<i>Ilustración 5.19 - Menú de herramientas.....</i>	<i>52</i>
<i>Ilustración 5.20 - Ejemplo de uso de BoundingBoxRig.....</i>	<i>53</i>
<i>Ilustración 5.21 - Menú Herramientas de sistema desplegado.....</i>	<i>53</i>
<i>Ilustración 5.22 - Panel de guardado de nube con teclado desplegado.....</i>	<i>54</i>
<i>Ilustración 5.23 - Script PHP que recoge el fichero PLY y lo almacena en el servidor</i>	<i>55</i>
<i>Ilustración 5.24 - Menú de Herramientas de nube desplegado</i>	<i>56</i>
<i>Ilustración 5.25 - Pivote de manipulación de nube</i>	<i>56</i>
<i>Ilustración 5.26 - Ejemplo de nube de puntos en modo manipulación</i>	<i>57</i>
<i>Ilustración 5.27 - Menú Herramientas de puntos desplegado</i>	<i>58</i>
<i>Ilustración 5.28 - Ejemplo de puntos seleccionados.....</i>	<i>59</i>
<i>Ilustración 5.29 - Error necesario Windows 10 Fall Creators</i>	<i>61</i>
<i>Ilustración 5.30 - Ejemplo de incompatibilidad entre librerías.....</i>	<i>63</i>
<i>Ilustración 5.31 - Listado de shaders del proyecto</i>	<i>65</i>
<i>Ilustración 5.32 - Código para encontrar cámara uiRaycast.....</i>	<i>66</i>
<i>Ilustración 7.1 - Ejemplo de gesto básico "Tap"</i>	<i>83</i>
<i>Ilustración 7.2 - Ejemplo de escaneado del entorno</i>	<i>83</i>
<i>Ilustración 7.3 - Ejemplo de mínimo de superficie escaneada alcanzado</i>	<i>84</i>
<i>Ilustración 7.4 - Ejemplo de menú de selección de nubes</i>	<i>84</i>
<i>Ilustración 7.5 - Ejemplo de gesto básico "Hold"</i>	<i>85</i>
<i>Ilustración 7.6 - Ejemplo selección de una nube</i>	<i>85</i>
<i>Ilustración 7.7 - Ejemplo de menú principal</i>	<i>86</i>
<i>Ilustración 7.8 - Ejemplo de nube cargada y menú de herramientas.....</i>	<i>86</i>
<i>Ilustración 7.9 - Ejemplo de edición de icono 3D del menú de Herramientas.....</i>	<i>87</i>
<i>Ilustración 7.10 - Ejemplo de menú de herramientas de sistema y botones desplegados.....</i>	<i>88</i>
<i>Ilustración 7.11 - Ejemplo Panel guardar nube y teclado 3D</i>	<i>88</i>
<i>Ilustración 7.12 - Ejemplo de icono de Herramientas de nube.....</i>	<i>89</i>
<i>Ilustración 7.13 - Ejemplo de manipulación de nube con dos manos</i>	<i>89</i>
<i>Ilustración 7.14 - Ejemplo de menú de herramientas de puntos y botones desplegados</i>	<i>90</i>
<i>Ilustración 7.15 - Ejemplo de inicio de la selección de puntos.....</i>	<i>90</i>
<i>Ilustración 7.16 - Ejemplo de puntos marcados</i>	<i>91</i>

Índice de tablas

<i>Tabla 4-1 - Fechas de inicio de iteraciones.....</i>	<i>25</i>
<i>Tabla 4-2 - Historias de usuario iniciales.....</i>	<i>27</i>
<i>Tabla 4-3 - Estimación de recursos digitales.....</i>	<i>29</i>
<i>Tabla 4-4 - Estimación de recursos materiales.....</i>	<i>31</i>
<i>Tabla 4-5 - Estimación de coste de personal.....</i>	<i>32</i>
<i>Tabla 4-6 - Estimación de coste total de proyecto.....</i>	<i>32</i>
<i>Tabla 5-1 - Bytes e identificadores de tipos de datos.....</i>	<i>50</i>
<i>Tabla 7-1 – Historia de usuario inicial ID: 1.....</i>	<i>69</i>
<i>Tabla 7-2 – Historia de usuario inicial ID: 2.....</i>	<i>69</i>
<i>Tabla 7-3 – Historia de usuario inicial ID: 3.1.....</i>	<i>70</i>
<i>Tabla 7-4 - Historia de usuario inicial ID: 3.2.....</i>	<i>70</i>
<i>Tabla 7-5 – Historia de usuario inicial ID: 3.3.....</i>	<i>70</i>
<i>Tabla 7-6 – Historia de usuario inicial ID: 3.4.....</i>	<i>71</i>
<i>Tabla 7-7 – Historia de usuario inicial ID: 3.5.....</i>	<i>71</i>
<i>Tabla 7-8 – Historia de usuario inicial ID: 4.....</i>	<i>71</i>
<i>Tabla 7-9 – Historia de usuario inicial ID: 5.1.....</i>	<i>72</i>
<i>Tabla 7-10 – Historia de usuario inicial ID: 5.2.....</i>	<i>72</i>
<i>Tabla 7-11 – Historia de usuario inicial ID: 5.3.1.....</i>	<i>72</i>
<i>Tabla 7-12 – Historia de usuario inicial ID: 5.3.2.....</i>	<i>73</i>
<i>Tabla 7-13 – Historia de usuario inicial ID: 5.2.....</i>	<i>73</i>
<i>Tabla 7-14 – Historia de usuario final ID: 1.....</i>	<i>74</i>
<i>Tabla 7-15 - Historia de usuario final ID: 2.....</i>	<i>74</i>
<i>Tabla 7-16 - Historia de usuario final ID: 3.....</i>	<i>75</i>
<i>Tabla 7-17– Historia de usuario final ID: 4.....</i>	<i>75</i>
<i>Tabla 7-18 – Historia de usuario final ID: 5.....</i>	<i>75</i>
<i>Tabla 7-19– Historia de usuario final ID: 6.....</i>	<i>76</i>
<i>Tabla 7-20 – Historia de usuario final ID: 7.....</i>	<i>76</i>
<i>Tabla 7-21 – Historia de usuario final ID: 8.....</i>	<i>76</i>
<i>Tabla 7-22– Historia de usuario final ID: 9.....</i>	<i>77</i>
<i>Tabla 7-23– Historia de usuario final ID: 10.....</i>	<i>77</i>
<i>Tabla 7-24– Historia de usuario final ID: 11.....</i>	<i>78</i>
<i>Tabla 7-25– Historia de usuario final ID: 12.....</i>	<i>78</i>
<i>Tabla 7-26– Historia de usuario final ID: 13.....</i>	<i>78</i>
<i>Tabla 7-27– Historia de usuario final ID: 14.....</i>	<i>79</i>
<i>Tabla 7-28– Historia de usuario final ID: 15.....</i>	<i>79</i>
<i>Tabla 7-29– Historia de usuario final ID: 16.....</i>	<i>79</i>
<i>Tabla 7-30– Historia de usuario final ID: 17.....</i>	<i>79</i>
<i>Tabla 7-31– Historia de usuario final ID: 18.....</i>	<i>80</i>

1. Introducción

En los últimos años han surgido diferentes tecnologías que están aportando una gran cantidad de avances en el desarrollo de la industria y mejorando la calidad de los servicios que ofrecen.

Algunas de ellas, como las tecnologías inmersivas y las tecnologías de escáneres láser 3D, están demostrando un gran potencial que puede ser aprovechado para múltiples fines en ámbitos como por ejemplo la medicina o industria.

En el caso de las tecnologías inmersivas, capaces de alterar la realidad del usuario con elementos virtuales por medio del internet de las cosas (IoT [1]), han ganado mucha popularidad en la industria del entretenimiento, ejemplos de ello son títulos como “Pokemon Go” [2] y “Paranormal Activity: The Lost Soul” [3].

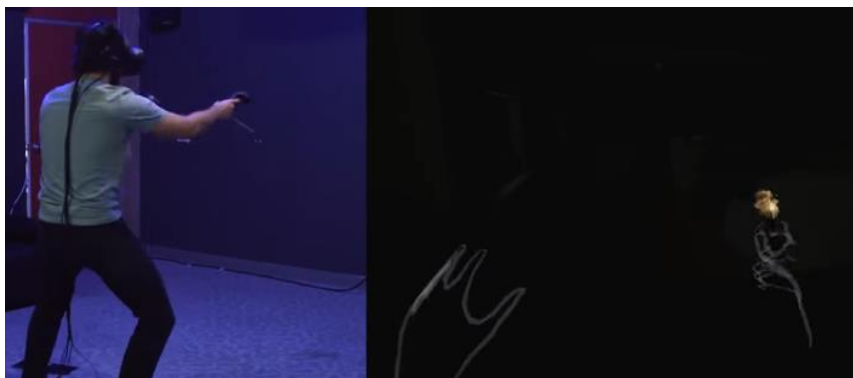


Ilustración 1.1 - Videojuego “Paranormal Activity: The lost soul”



Ilustración 1.2 - Videojuego "Pokemon GO"

Estas tecnologías junto con las actuales plataformas, no solo están dominando el sector del entretenimiento, sino que además han comenzado a introducirse en ámbitos como la medicina, facilitando cirugías, y la educación, permitiendo, por ejemplo, a los alumnos experimentar en primera persona eventos históricos.



Ilustración 1.3 - Simulador nave espacial con realidad virtual



Ilustración 1.4 - Aplicación de apoyo quirúrgico con realidad aumentada

También pueden ser muy útiles para la industria 4.0 [4], incorporando dichas tecnologías en las empresas para mejorar su rendimiento y desarrollo.

Un ejemplo de ello es Renault Trucks, que está incorporando la realidad aumentada a sus procesos de control de calidad en las operaciones de ensamblaje de motores [5].



Ilustración 1.5 - Aplicación de control de calidad con realidad aumentada

Por otro lado, tenemos a las tecnologías de escáneres láser 3D de las que podemos obtener nubes de puntos, Point Clouds [6]. Por ejemplo, la Ilustración 1.6 es resultado de la combinación de múltiples tomas de un puente.

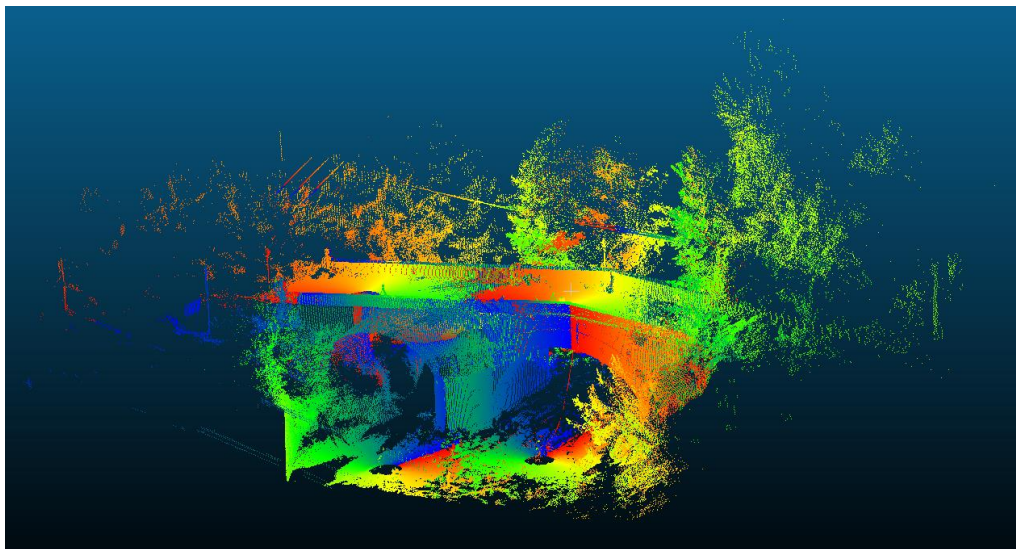


Ilustración 1.6 - Nube de puntos de un puente

Las nubes de puntos se han estado utilizando actualmente en la industria de la ingeniería, aportando precisión y mayor calidad de diseño, con una menor carga gráfica que aumenta el rendimiento al carecer de geometría.

Pero para ello requieren de una edición previa para eliminar ruido y elementos innecesarios, como vegetación, vehículos, etc, lo cual no es una tarea sencilla.

La edición de puntos con medios tradicionales es compleja, ya que lo habitual es realizarla en un ordenador, y la percepción espacial con puntos es dificultosa en una pantalla, donde no se aprecia la profundidad.

Uno de los problemas más habitual, en la edición de puntos, es que estos están desconectados, y a menos que estén cerca y tengan colores similares, no es fácil identificar objetos y su ubicación.

De modo que, dada la calidad y precisión que nos puede aportar las nubes de puntos, la necesidad de facilitar la edición de las mismas y las posibilidades que nos ofrecen las tecnologías inmersivas, estudiaremos el desarrollo de una aplicación inmersiva que permita una visualización en tres dimensiones de una nube de puntos y editarla fácilmente.

2. Objetivos

Este proyecto tiene como fin diseñar e implementar un prototipo de aplicación de realidad mixta [7] , utilizando la plataforma Microsoft HoloLens [8], que permita al usuario visualizar y editar nubes de puntos.

Para ello, será necesario que durante el desarrollo de este proyecto cumplir con los siguientes objetivos generales:

- Investigar las tecnologías disponibles y profundizar en el SDK de Microsoft HoloLens, para hallar las posibilidades para el proyecto.
- Realizar un análisis del sistema, obtener los requerimientos del mismo, determinar una estimación temporal y costes del proyecto.
- Desarrollar una aplicación basada en el SDK de Microsoft HoloLens que permita superponer nubes de puntos de escenas reales obtenidas mediante dispositivos de captura.
- Implementar operaciones de selección y edición de nubes de puntos utilizando una interfaz basada en gestos que proporciona Microsoft HoloLens.
- Diseñar y aplicar pruebas de evaluación y de usuario.

Para realizar los anteriores objetivos, se requerirá por parte del alumno disponer de conocimientos de diseño y programación de software, experiencia en desarrollo de aplicaciones gráficas y capacidad investigadora.

3. Software y Dispositivos de desarrollo

Para desarrollar del prototipo de este proyecto, se necesitan unas Microsoft Hololens y software de programación adecuado.

También se hará uso de otras herramientas software, como Photoshop [9] y 3DS MAX [10] de Autodesk, ya que se precisará de modelos 3D y texturas para la interfaz de usuario, y son herramientas con las que el alumno ya ha trabajado anteriormente.

3.1. Análisis del dispositivo Microsoft Hololens

Hololens son unas gafas de realidad mixta diseñadas por Microsoft de las que actualmente están disponibles al público español en dos versiones, la versión de desarrollo y la versión comercial, cuyo coste es de 3299€ y 5489€ respectivamente [11]. En nuestro caso, contaremos con la edición de desarrollo.



Ilustración 3.1 - Microsoft Hololens

Hololens cuenta con un diseño ergonómico y adaptable, su peso es de 579g, tiene instalado Windows 10 como sistema operativo, un chipset Intel de 32 bits con módulo de seguridad TPM 2.0, una batería que aporta entre 2 y 3 horas de autonomía en uso intensivo, 64GB de memoria interna y 2GB de memoria RAM [12].



Ilustración 3.2 - Placa base de Hololens

En cuanto a la parte óptica, dispone de lentes holográficas transparentes, dos motores de luz HD en formato panorámico 16:9 y un sistema de calibración automática de distancia pupilar.



Ilustración 3.3 - Parte óptica de Hololens

La resolución holográfica de Hololens es de 2.3 millones de puntos de luz y la densidad holográfica es de más de 2500 radianes (puntos de luz por radián).

Por cada ojo, la resolución máxima es de 720p, o lo que es lo mismo, de 1268x720, y la resolución mínima es de 360p (634x360) [13]. En cuanto al número de frames por segundo, Microsoft recomienda a los desarrolladores que se mantengan en un mínimo de 60 frames por segundo, pues este se reduce a 30 al utilizar la herramienta de captura de video.

Por otro lado, Hololens tiene incorporados varios sensores que le permiten interactuar con el usuario, Human Understanding, por diferentes

medios como sonido espacial, seguimiento de la mirada, entrada de gestos y soporte de voz.



Ilustración 3.4 - Sensores Hololens

Hololens cuenta con un sensor IMU (unidad de medición inercial) [14], cuatro cámaras de captura de entorno, una cámara de profundidad, una cámara de 2 megapíxeles para realizar fotos y video en HD, que permite grabar y fotografiar los hologramas que visualiza el usuario (captura de realidad mixta) [15]. También dispone de cuatro micrófonos y un sensor de luz ambiental, para poder regular la iluminación de los hologramas de forma automática.

Por último, Hololens tiene incluidos algunos elementos de entrada y salida (I/O) y de conectividad, como altavoces integrados, una conexión de audio Jack de 3.5mm, un botón de encendido y apagado, leds indicadores del estado de la batería, potenciómetros para el volumen y el brillo, una conexión micro USB 2.0 para carga y datos, conexión Wi-Fi 802.11ac y conexión Bluetooth 4.1 LE.

3.2. Análisis y configuración del software

Para el desarrollo de aplicaciones para Hololens, Microsoft recomienda instalar la siguiente lista de herramientas software [16]:

- IDE Visual Studio
- SDK de Windows 10
- Emulador de Hololens
- Vuforia
- Unity 3D
- MixedRealityToolkit-Unity

Además de dichas herramientas, que se describirán a continuación, se requiere tener instalada la última actualización de Windows 10 [17] para un mejor rendimiento del SDK de Windows 10, y habilitar en el equipo la virtualización de hardware, Hyper-V [18], ya que el emulador de Hololens está basado en Hyper-V y utiliza RemoteFx [19] para gráficos acelerados por hardware.

3.2.1. IDE Visual Studio

Todas las ediciones de Visual Studio 2017 son aptas para desarrollar aplicaciones para Hololens, incluso la tercera actualización de la versión de 2015, aunque Microsoft recomienda utilizar la versión 2017.

En este caso se usará Visual Studio Community 2017 [20], que es gratuita. Durante la instalación hay que seleccionar el módulo “Universal Windows Platform development” y desmarcar “Game Development with Unity” ya que se instalará una versión concreta de Unity más adelante.

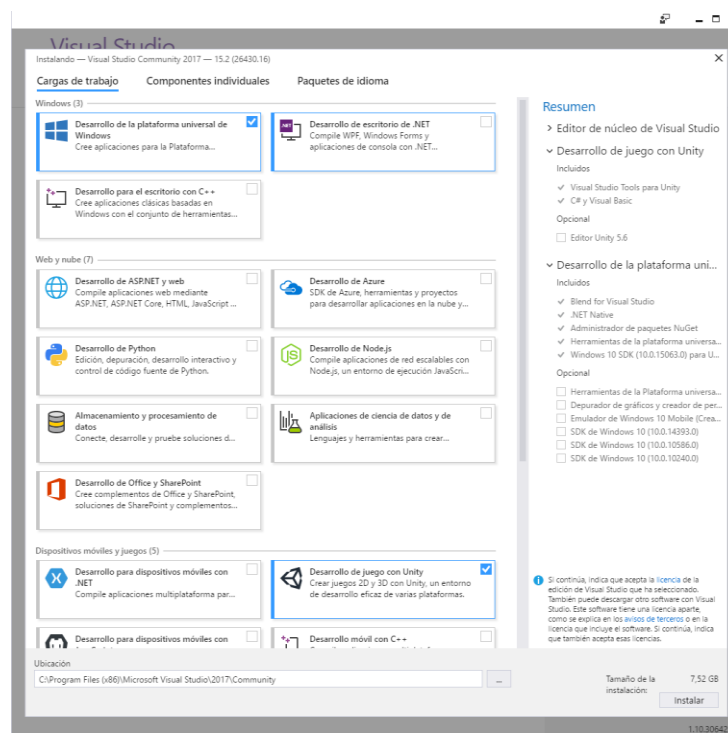


Ilustración 3.5 - Módulos Visual Studio 2017

3.2.1. SDK de Windows 10 y emulador Hololens

El SDK de Windows 10 puede instalarse de dos modos, desde la instalación de Visual Studio 2017, ubicado en la columna derecha “Resumen”, o bien, instalarse directamente desde la web de Microsoft [21].

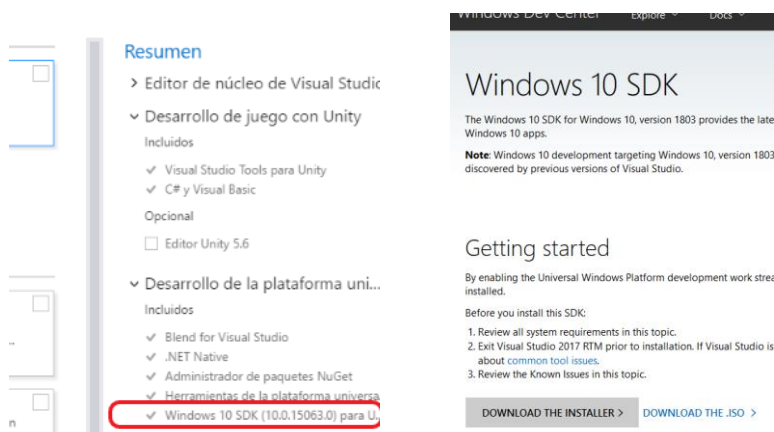


Ilustración 3.6 - SDK Windows 10

El emulador de Hololens será útil para depurar la aplicación que se va a desarrollar antes de probarla en las Hololens.

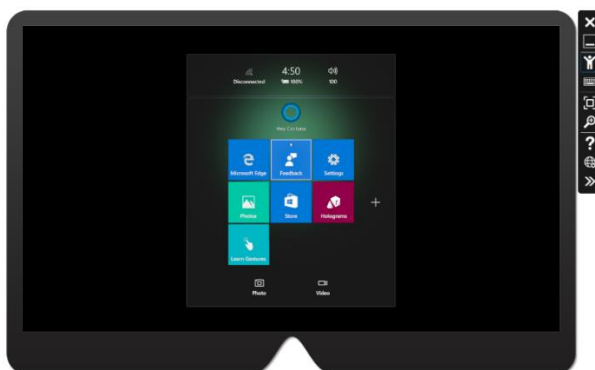


Ilustración 3.7 - Vista principal de Emulador Hololens

La instalación es sencilla, bastará con descargar, ejecutar el instalador proporcionado por Microsoft y seguir las instrucciones de instalación.

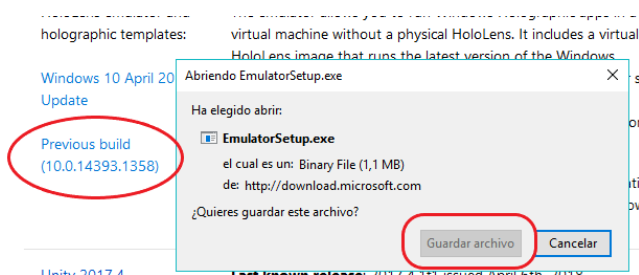


Ilustración 3.8 - Descarga instalador de emulador Hololens

Desde el emulador se puede interactuar con una máquina virtual de Hololens a través de teclado y ratón o utilizando un controlador o gamepad de Xbox, cuyo manejo es más sencillo.

En el menú lateral derecho del emulador, se puede visualizar diferentes opciones como minimizar y cerrar el emulador, hacer zoom y ajustar la ventana, etc... Aunque se centrará la atención en la última opción, "Tools", pues será más útil durante la depuración de la aplicación.



Ilustración 3.9 - Botón "Tools" de emulador Hololens

En este apartado se puede visualizar un panel donde especificar de forma manual algunos parámetros de la simulación como la habitación, la conexión a internet, etc...

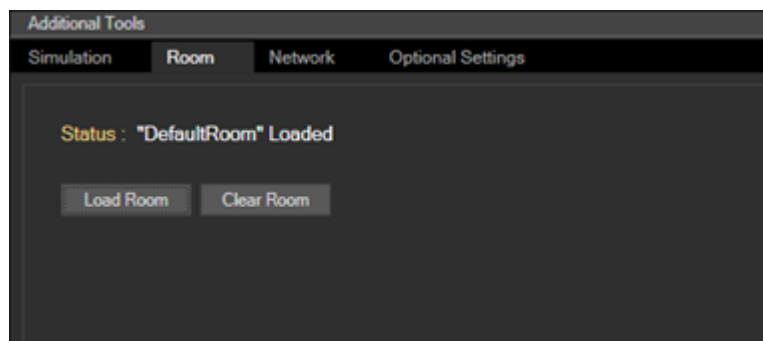


Ilustración 3.10 - Panel parámetros de simulación

3.2.2. Vuforia

Vuforia [22] es un SDK de realidad aumentada [23], utilizada principalmente para el reconocimiento de texto, imágenes y objetos a través de la información obtenida por una cámara.

En las recientes versiones de Unity 3D [24], Vuforia ya está incorporarlo al motor de juegos, por lo que podremos utilizarlo para nuestros proyectos.

Sin embargo, no se hará uso de Vuforia en este proyecto ya que el prototipo que se va a diseñar solo requiere reconocer el espacio y los gestos del usuario.

Para ello, se usará un “kit” de herramientas para realidad mixta [25] compatible con Unity diseñado por Microsoft, MixedRealityToolkit (MRTK) [26], donde ya están definidos los gestos utilizados en aplicaciones para Hololens y sistemas de identificación del espacio, además de otras herramientas y librerías que facilitarán el desarrollo de la aplicación.

3.2.3. *Unity 3D*

Se trata de un motor de juegos que tiene grandes facilidades que aceleran el desarrollo de prototipos, como es el presente caso, además Unity3D tiene soporte para el desarrollo de juegos y aplicaciones de realidad mixta y aumentada.

Tiene disponible un simulador holográfico similar al que se instaló anteriormente, con la diferencia de que solo puede manejarse con un controlador o gamepad de Xbox, aunque este emulador no estará disponible hasta que se aplique la configuración del proyecto para Hololens.

Dicha configuración, que se expondrá a continuación, es el resultado de la combinación de todas investigaciones realizadas en este proyecto, la mayoría para resolver problemas de incompatibilidades y optimización que han surgido durante el desarrollo de la aplicación.

Tras esta aclaración, se procede a explicar los pasos a seguir para aplicar la configuración Unity adecuada para Hololens.

El primer paso, es dirigirse a la ventana de construcción del proyecto, File > Build Settings, donde en primer lugar se cambiará la plataforma seleccionando “Universal Windows Platform” y pulsando sobre el botón “Switch Platform”, como se puede apreciar en la Ilustración 3.11.

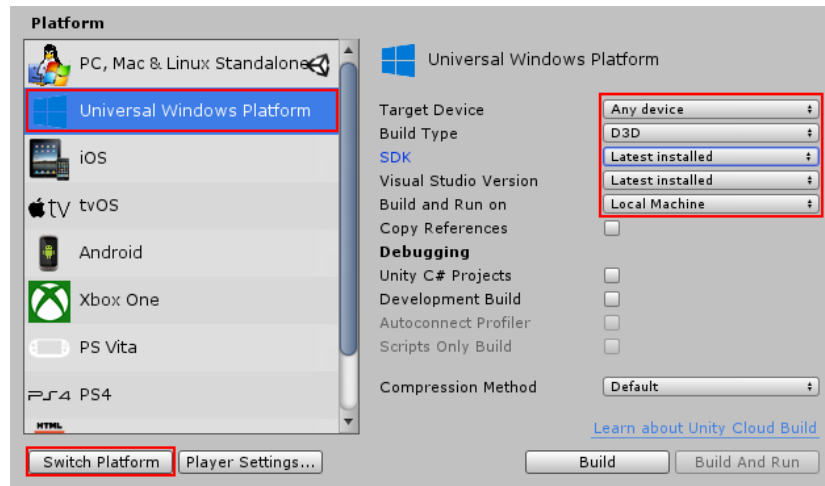


Ilustración 3.11 - Configuración Unity: Cambio de plataforma

Seguidamente, se indicará como dispositivo “Hololens”, en el tipo de construcción, seleccionaremos “D3D”.

En el campo SDK, se puede dejar la selección por defecto para que el sistema elija el más reciente o indicar una versión concreta si se tienen varias instaladas, igualmente se hará con la selección de la versión de Visual Studio. Y, por último, hay que marcar el campo “Copy References”, para que se importen las referencias de Unity necesarias en la solución que se construya.

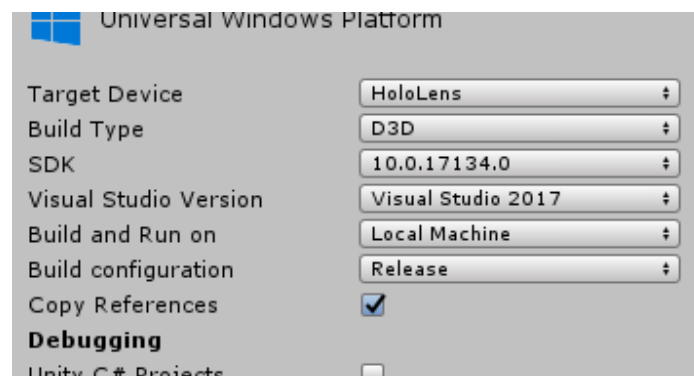


Ilustración 3.12 - Configuración Unity: Características del proyecto

A continuación, se accede a la configuración de usuario pulsando en el botón “Player Settings” y hay que dirigirse al apartado “XR Settings” del panel derecho.

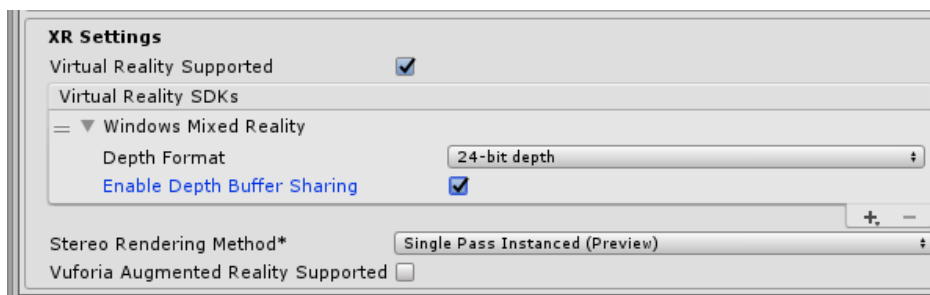


Ilustración 3.13 - Configuración Unity: XR Settings

Una vez ahí, se habilitará el soporte para realidad virtual, “Virtual Reality Supported”, seguidamente se mostrará la caja de dispositivos de realidad virtual, donde al pulsar en añadir se podrá seleccionar varios dispositivos. En este caso, seleccionaremos “Windows Mixed Reality”.

En este punto, se expandirán las opciones del dispositivo, se indicará el formato de profundidad a 24-bit y hay que habilitar el buffer de profundidad. Seguidamente, se cambia el método de renderización estéreo a “Single Pass Instantiated (Preview)”, que es el método más óptimo para desarrollar en Hololens.

A continuación, en el apartado “Other Settings” se debe asegurar que trabaja con .NET, observando los campos “Scripting”, tal y como se puede ver en la Ilustración 3.14.

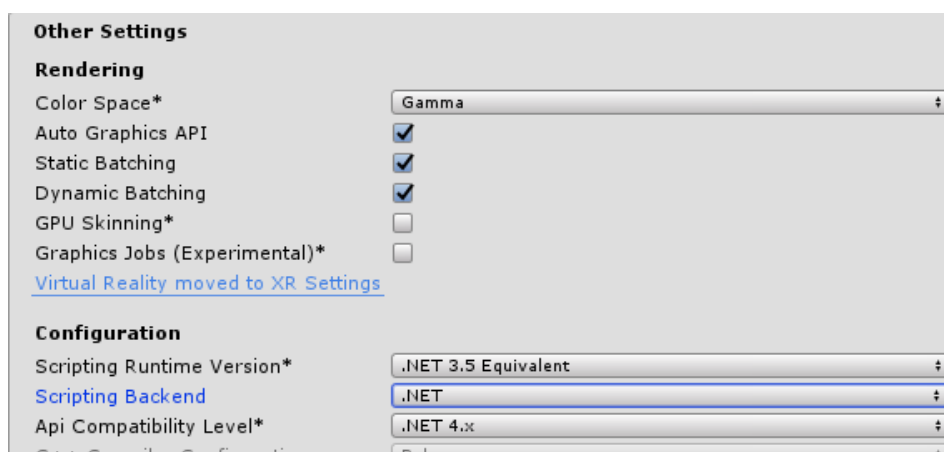


Ilustración 3.14 - Configuración Unity: Other Settings

Después, en el apartado “Publishing Settings” se marcarán aquellas capacidades necesarias para las aplicaciones holográficas, en este caso serán las siguientes:

- SpatialPerception
- WebCam
- Microphone
- InternetClient

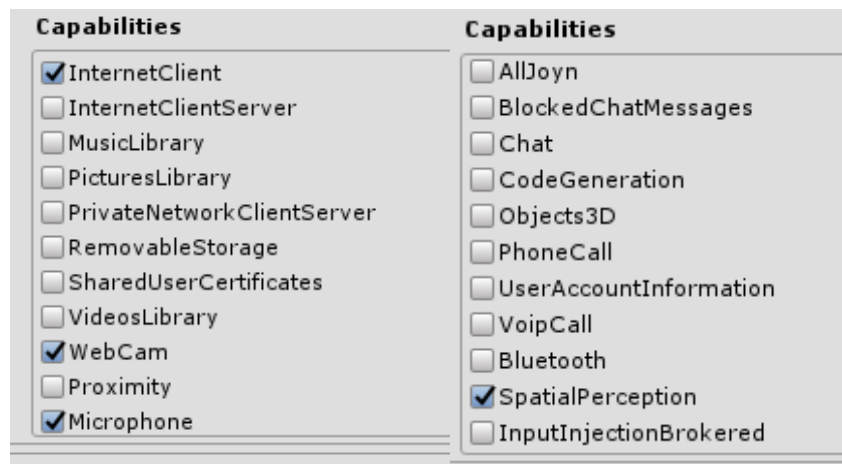


Ilustración 3.15 - Configuración Unity: Capabilities

Hay que recordar que Hololens utiliza una GPU de tipo móvil, por lo que se precisará reducir la calidad a la más rápida, para ello hay que acceder a “Edit” desde la barra superior de Unity, entrar en “Quality” y en la lista de Microsoft seleccionar la calidad más baja que será la más óptima para la aplicación.

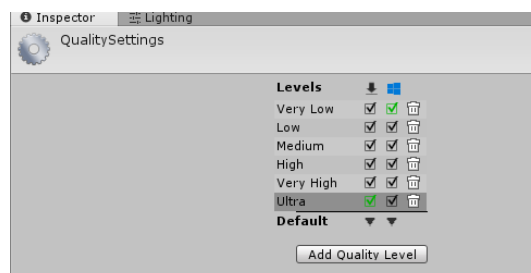


Ilustración 3.16 - Configuración Unity: Quality Settings

Por último, hay que activar la pestaña del emulador holográfico para poder realizar las primeras pruebas durante el desarrollo, ya que después se deberá probar la aplicación en el emulador de visual studio y finalmente en las Hololens.

Para ello, bastará con acceder a “Window” en la barra de herramientas y seleccionar “Holographic emulation”, Ilustración 3.17.

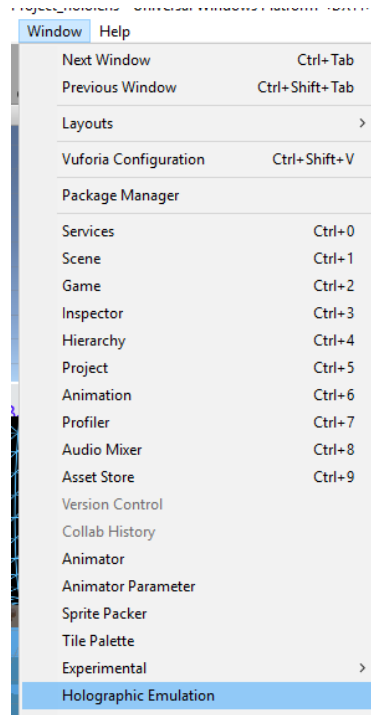


Ilustración 3.17 - Configuración Unity: Emulador Hololens

Este emulador, dispone de diferentes métodos de emulación, incluso pudiendo utilizar las Hololens en modo remoto. Sin embargo, este método deja de ser viable cuando se requiere poder reconocer gestos, por lo que es recomendable utilizar el método “Simulate in Editor”, Ilustración 3.18, dónde se puede elegir una habitación pre-registrada como escenario de pruebas, y simulas los gestos con un gamepad.

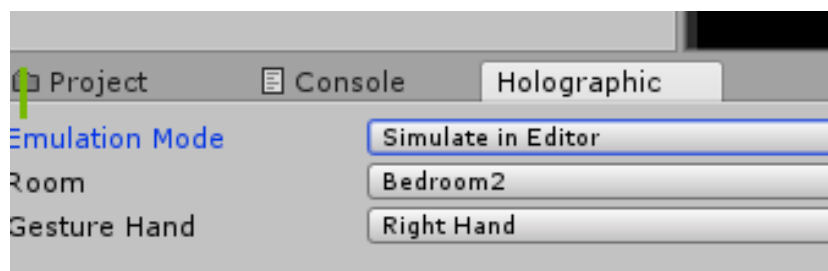


Ilustración 3.18 - Configuración Unity: Configuración de emulador Hololens

Ahora que ya está preparado Unity para Hololens se puede importar el paquete MRTK para comenzar con el desarrollo del proyecto.

3.2.4. MixedRealityToolkit-Unity

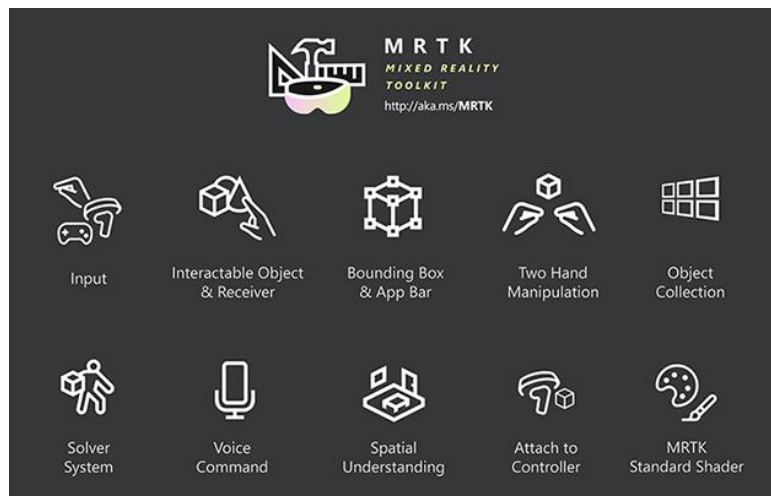


Ilustración 3.19 - Paquete MRTK

MRTK es un paquete de librerías desarrollado por Microsoft, para facilitar el desarrollo de aplicaciones para Hololens. En el presente caso, se utilizará una versión para Unity, que además de las librerías contiene prefabs [27] de elementos esenciales, como la cámara o el sistema de reconocimiento del entorno, lo que nos ahorrará mucho trabajo.

Para incorporarlo al proyecto, bastará con dirigirse a “Assets” en la barra de herramientas, “Import Package” y “Custom Package”, Ilustración 3.20. Seguidamente, se seleccionará el paquete de MRTK e importará todo su contenido al proyecto.

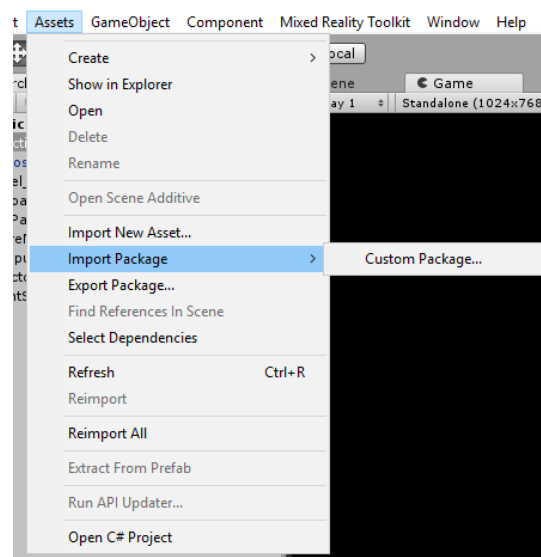


Ilustración 3.20 - MRTK - Importando paquete a Unity

4. Planificación del proyecto

4.1. Metodología de desarrollo

Para la gestión del presente proyecto se va a aplicar una metodología ágil, ya que se conoce el objetivo a alcanzar, pero se desconoce la solución que se debe tomar para alcanzarlo.

Se hará uso de una metodología ágil basada en SCRUM [28], por lo que se trabajará por paquetes de tareas repartidos en sprints o iteraciones que tendrán una duración de 2 semanas de trabajo.

Para comenzar con la planificación se requerirá la elaboración de las historias de usuario, HU, que son descripciones de funcionalidades que pueden estar disponibles en el proyecto [29], por parte del alumno.

El alumno deberá estimar las HU con Puntos de Historia, que sirven para valorar el esfuerzo que requieren las HU [30], en función de su tamaño, aplicando una estimación por analogía. Para ello se utilizarán valores de la siguiente serie, usualmente utilizada para estas estimaciones, (0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100) [31].

Se establecerá una priorización de las HU con la fórmula MoSCoW, que es una técnica de priorización [32]. Posteriormente, al inicio de cada sprint se realizará una evaluación del desarrollo del proyecto y, de ser necesario, una nueva priorización de las HU seleccionadas.

Al inicio del proyecto, se realizarán un par de iteraciones sin establecer una velocidad, que es la sumatoria de los puntos de historia de una iteración [33], de modo que para las siguientes iteraciones se tomará una velocidad media a partir de las anteriores, pudiendo definir el número restante de iteraciones.

El número total de sprints se definirá a partir del tiempo disponible para realizar el proyecto, descontando aquel que el alumno haya dedicado a tareas ajenas al desarrollo de la aplicación, es decir, a aquellas tareas propias del

TFM, como la investigación previa o la elaboración de la memoria, y que no se contemplan es esta planificación.

Dicho tiempo descontado es de **70** horas, 30 ya invertidas en investigación, preparación y documentación inicial, y 40 que se reservarán para la documentación final del proyecto. Por lo que se dispone de **230** horas para el desarrollo del prototipo.

A partir de este punto, contando con que cada sprint tendrá una duración de 2 semanas y teniendo en cuenta el tiempo que el alumno puede dedicar a la elaboración del proyecto, se obtiene que el desarrollo del prototipo se realizará en **10** iteraciones aproximadamente.

Para mayor comodidad, y disponer de margen de tiempo, se trabajará con **11** iteraciones. De este modo, se dispondrá de media iteración extra para el desarrollo, se puede estimar las fechas de cada iteración:

<i>Iteraciones</i>	Fecha de inicio
<i>1</i>	22/1/2018
<i>2</i>	5/2/2018
<i>3</i>	19/2/2018
<i>4</i>	5/3/2018
<i>5</i>	19/3/2018
<i>6</i>	9/4/2018
<i>7</i>	23/4/2018
<i>8</i>	7/5/2018
<i>9</i>	21/5/2018
<i>10</i>	4/6/2018
<i>11</i>	18/6/2018

Tabla 4-1 - Fechas de inicio de iteraciones

Una vez definido el total de sprints, y conociendo la duración de las tareas de evaluación y documentación del presente TFM, lo cual se estima que conllevará medio sprint, se puede definir una fecha de finalización.

Fecha de finalización estimada: 1 / 7 / 2018

Tras la finalización de cada sprint, se realizará el correspondiente informe del trabajo realizado, plasmándolo directamente en la memoria, junto con los correspondientes gráficos de la evolución de la iteración y la evolución general del proyecto hasta el momento.

4.2. Historias de Usuario

Cómo todo proyecto de ingeniería software, es necesario establecer los requisitos del proyecto a desarrollar para poder comenzar a trabajar.

En este caso, no se trata de desarrollar un software a medida para un cliente en concreto, por lo que el alumno deberá interpretar el papel del cliente para poder definir los requisitos y funcionalidades del proyecto.

Posteriormente, el alumno deberá valorar los requisitos y funcionalidades deseadas como cliente, e interpretarlas como historias de usuario, priorizándolas y puntuándolas.

Se comienza diseñando los casos de uso del proyecto, Anexo 7.3.1, para descubrir las necesidades del sistema y tener una visión más general del mismo. En este proyecto, se detectan dos grandes casos de uso: “Obtención de la nube de puntos”, Ilustración 4.1, y “Edición de la nube de puntos”, Ilustración 4.2.

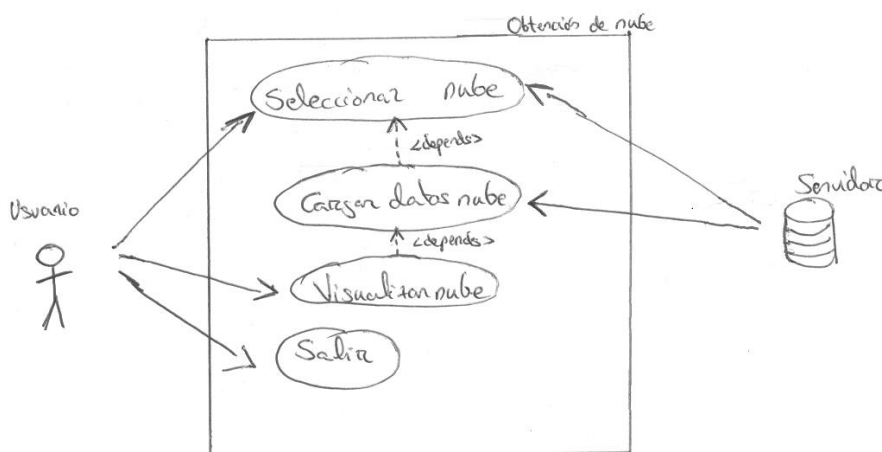


Ilustración 4.1 - Diagrama de casos de uso: Obtención de la nube de puntos

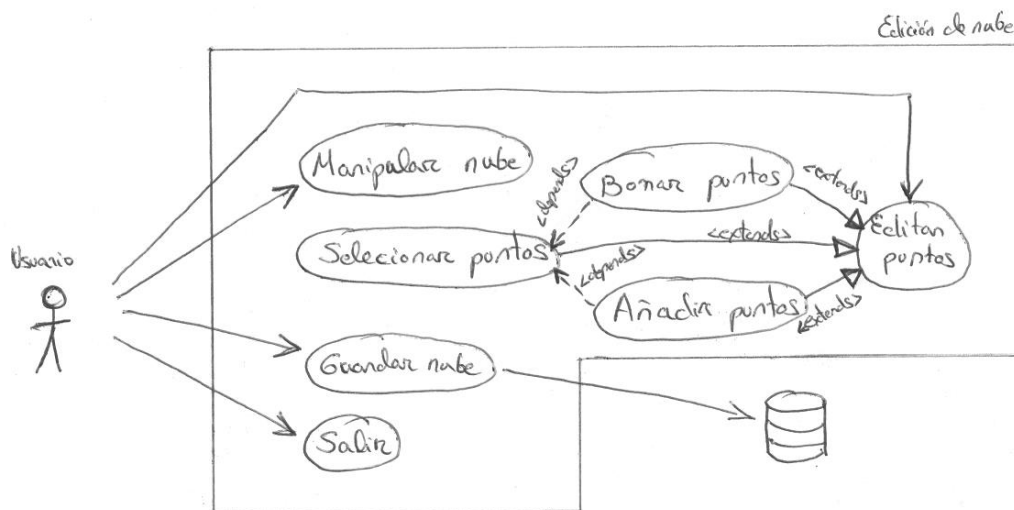


Ilustración 4.2 - Diagrama de casos de uso: Edición de la nube de puntos

Tras este proceso, se han elaborado las siguientes historias de usuario, cuyas fichas/tarjetas, con toda la información relevante, están disponibles en el anexo 7.1:

ID	Tipo	Título	Prioridad	Puntos H.U.
1	H.U.	Sistema de carga de PCL	Must	13
2	H.U.	Visualización de PCL	Must	20
3	Tema	Edición de PCL	-	-
3.1	H.U.	Manipulación de PCL	Must	20
3.2	H.U.	Selección de puntos por área	Must	40
3.3	H.U.	Manipulación de puntos seleccionados	Should	40
3.4	H.U.	Borrado de puntos de PCL	Must	13
3.5	H.U.	Añadido de puntos a PCL	Would	40
4	H.U.	Guardado de la nube de PCL	Must	13
5	Tema	Interfaz de usuario	-	-
5.1	H.U.	Menú de selección de PCL	Must	13
5.2	H.U.	Menú inicio de aplicación	Must	5
5.3	Epic	<i>Menú de Herramientas PCL</i>	Must	-
5.3.1	H.U.	Menú manipulación CPL	Must	40
5.3.2	H.U.	Menú herramientas puntos de PCL	Must	40
5.4	H.U.	Menú guardado PCL	Could	8

Tabla 4-2 - Historias de usuario iniciales

Cabe aclarar, que estás son las historias de usuario iniciales. Durante el desarrollo del proyecto, antes de comenzar una nueva iteración es probable que algunas de ellas sufran cambios, se dividan, se descarten o, incluso, sea necesario añadir nuevas.

Se explicará con más detalle la evolución del proyecto en el apartado 4.4 de esta memoria.

4.3. Estimación de recursos y costes

En este apartado del proyecto se va a proceder a la estimación de los recursos y costes del mismo.

En cuanto a la estimación de los recursos, el alumno debe prever los recursos que se van a consumir durante el desarrollo del proyecto, tanto digitales, materiales y de personal, para poder establecer un cálculo aproximado del coste del proyecto, teniendo presente la perspectiva de un entorno empresarial.

Para realizar dicha estimación, se comenzará en primer lugar por establecer los recursos digitales.

Se entiende por recursos digitales las herramientas software que van a utilizarse en este proyecto, véase la Tabla 4-3.

La mayoría del software utilizado es de licencia gratuita, por lo que no tiene coste, como Visual Studio Community 2017, Unity, MRTK y CloudCompare.

Sin embargo, el paquete Microsoft Office es un paquete de software de pago. Entre los diferentes packs para empresa que ofrece Microsoft en su tienda online [34], se encuentra el pack “Office 365 Empresa” que, en la tarifa de pago mensual, es el más rentable con un coste de 10.60€ por usuario/mes. Teniendo en cuenta la estimación temporal que se ha realizado anteriormente, se necesitará de esta herramienta durante 6 meses, por lo que el coste total de la misma será de 63.60€.

Igualmente ocurre con la herramienta 3DS Max, que se necesitara para reconocimiento de modelos 3D importados. En este caso, observando las tarifas disponibles en la tienda oficial [35], a priori puede parecer que la suscripción mensual es la opción más económica, 248.05€, pero si se tiene en

cuenta los 6 meses de desarrollo, realmente es más económico optar por la tarifa anual, 1984.40€, pudiendo utilizar la herramienta en otros proyectos en los siguientes meses tras la finalización del prototipo, contando en el actual presupuesto con la mitad del coste de la herramienta.

Teniendo en cuenta que la suscripción anual tiene un coste de 1984.40€, para el presente presupuesto tendremos en cuenta solo la mitad, puesto que se utilizará durante medio año, siendo el coste de 992.2€.

Por último, se incluirá el coste de la licencia anual de Adobe Photoshop que tiene un coste para empresa de 359.88€ [36], pero solo se tendrá en cuenta la mitad.

Software	Finalidad	Coste
Unity Personal Edition	Desarrollo del prototipo	0€
Visual Studio Community 2017	Implementación y compilación de código del prototipo	0€
<i>MRTK Unity</i>	Paquete de librerías de Hololens para Unity	0€
3DS Max	Edición de modelos y reconocimiento de formato .max en Unity	992.2€
CloudCompare	Comparar representación de nubes de puntos	0€
Microsoft Office	• Elaboración de informes SCRUM y memoria del proyecto • Elaboración de la planificación y seguimiento de la evolución del proyecto	63.60€
Photoshop	Creación y edición de texturas e iconos	179.94€
	Coste total	1235.74€

Tabla 4-3 - Estimación de recursos digitales

Igualmente, se realizará la estimación de los recursos materiales necesarios.

Algunos de estos recursos no son consumidos en su totalidad en el desarrollo de proyecto, por ello, se realizará un cálculo entre su coste y la vida útil del recurso.

Es el caso del equipo utilizado para la elaboración de este trabajo fin de master. El coste de este equipo es de 636.2€, y la vida útil estimada para el mismo es de 35040 horas, es decir 4 años. Teniendo en cuenta el número de horas que durará el proyecto, el coste de este recurso será de 6€.

Esto también se debe aplicar a los servicios de luz y ADSL que se van a requerir en este proyecto.

Para calcular el consumo eléctrico que se estima realizar durante el desarrollo del proyecto, se debe tener en cuenta no sólo el consumo del equipo de desarrollo sino también la iluminación de la zona de trabajo.

El consumo en vatios del equipo de desarrollo es de 250W y el consumo de la iluminación de la zona de trabajo es de 7W.

Teniendo en cuenta el valor del kW/h, 0.143502€, el costo del consumo del equipo y la iluminación en kW/h es de 0.03688€,

Realizando un simple cálculo entre las horas que durará el proyecto y los vatios consumidos, se obtiene la estimación del coste total en consumo eléctrico de este proyecto, 11.064€.

Para estimar el coste del servicio ADSL que se va a utilizar durante el desarrollo del trabajo fin de master, se va a realizar un cálculo por el cual se obtendrá el coste del servicio por hora y, por consiguiente, el coste total del servicio en las 300 horas de duración del proyecto.

Teniendo en cuenta que la tarifa contratada es de 45€ mensuales, se obtiene que el coste por hora de este servicio es de 0.0625€.

Con este dato, se estima que el coste total de este servicio durante el desarrollo del proyecto es de 18.75€.

Por último, se agregarán el resto de materiales como las Hololens de Microsoft que utilizaremos en la ejecución y desarrollo del proyecto [37], 3299€.

A continuación, se elabora una lista de los recursos materiales anteriormente citados, junto con el resto de recursos que se consumirán en su totalidad en el proyecto, véase la Tabla 4-4:

Material	Finalidad	Coste
Equipo de desarrollo	Elaboración del proyecto	6€
Consumo eléctrico	Alimentación del Equipo de desarrollo	11.064€
Servicio ADSL	Consulta de documentación, investigación, adquisición y actualización del software	18.75€
Microsoft Hololens	Dispositivo sobre el que se ejecutará la aplicación	3299€
Notas adhesivas	Planificación SCRUM	3€
Coste Total		3337.814€

Tabla 4-4 - Estimación de recursos materiales

Seguidamente, se procede a estimar el coste de los recursos de personal, que en este proyecto se trata del trabajo realizado por el alumno.

Por dicho motivo, el alumno debe sopesar el valor de su esfuerzo y trabajo por hora, de manera que se pueda cuantificar económicamente su labor en este proyecto.

Para ello, se ha realizado una serie de consultas en portales web de búsqueda de empleo como, InfoJobs [38] y SimplyHired [39], con la intención de que el alumno pueda calcular el sueldo medio de los diferentes roles que toma el alumno en este trabajo fin de master.

Tras definir los papeles a interpretar por el alumno en el proyecto y estudiar diferentes ofertas de empleo del sector TI en España, se obtienen los presupuestos para gastos de personal según el puesto y el salario por horas que el alumno ejerce en dicho puesto, véase la Tabla 4-5.

<i>Puesto/Rol</i>	Salario anual	Salario por horas	Horas de trabajo	Coste
<i>Jefe de Proyecto</i>	42000€	18.75€	70	1312.5€
<i>Programador Unity .Net</i>	33000€	14.73€	180	2651.4€
<i>Programador UI/UX</i>	21000€	9.38€	50	469
	Total			4432.9€
	Total con 30% de impuesto de contratación			5762.77€

Tabla 4-5 - Estimación de coste de personal

Tras realizar los cálculos necesarios, se obtiene que el coste total de los recursos de personal es de 5762.77€.

Los salarios con los que se ha elaborado la anterior tabla, son salarios en bruto anuales repartidos en 14 pagas. El salario por horas se ha obtenido como resultado de dividir el salario bruto anual en 14 pagas, en 20 días y 8 horas laborales.

Las horas dedicadas por cada rol son estimadas, aunque se conozca el tiempo que se dedicará a planificación, gestión y documentación del proyecto, en la parte de desarrollo, a priori, es complicado concretar con certeza el tiempo que se requerirá tanto en desarrollar las funcionalidades de la aplicación, como diseñar la experiencia de usuario en un proyecto dónde puede surgir muchos cambios.

Una vez ya se dispone de todas las estimaciones sobre los recursos a consumir durante el proyecto, se puede obtener el coste total estimado del proyecto, véase la Tabla 4-6.

<i>Recursos</i>	Coste
<i>Recursos Digitales</i>	1235.74 €
<i>Recursos Materiales</i>	3337.814 €
<i>Recursos de Personal</i>	5762.77
Total	10336.32€

Tabla 4-6 - Estimación de coste total de proyecto

Para establecer un presupuesto final, a este coste total se le aplicará un incremento del 20%, con la intención de cubrir los costes de posibles inconvenientes o infravaloraciones.

De este modo, el presupuesto para el proyecto es de **12403.58€**.

4.4. Evolución del proyecto

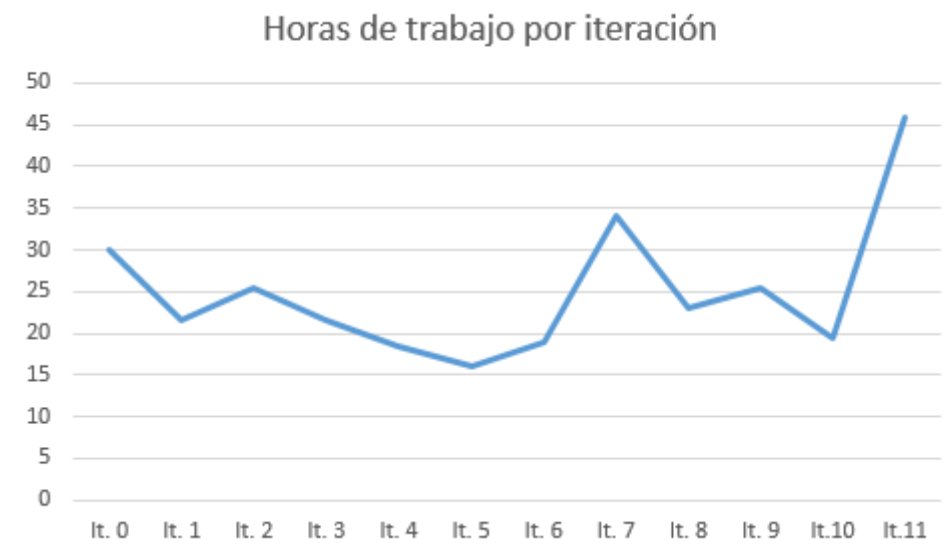
En este punto del presente documento se procederá a realizar un resumen de la evolución del proyecto.

Comenzamos este proyecto con un total de 13 historias de usuario y 305 puntos de historia, anexo 7.1, al final de esta primera versión de la aplicación se cuenta con un total de 18 historias de usuario y un total de 402 puntos de historia, Anexo 7.2.

De todas ellas, se han llegado a implementar 14 con un total de 282 puntos de historia realizados, véase dicho progreso en la Ilustración 4.3 y en la Ilustración 4.4.

Por lo que han quedado pendientes 4 historias de usuario, las historias 15, 16, 17 y 18, con un total de 120 puntos de historia pendientes de implementar, y que pueden realizarse en una futura segunda versión del proyecto.

También cabe aclarar, que todas las pruebas de aceptación definidas en las historias de usuario finales, disponibles en el Anexo 7.2, se han realizado con éxito. Los contratiempos encontrados durante dichas pruebas se ha detallado en el apartado 5.4 de la presente memoria.

**Ilustración 4.3 – Gráfico: Velocidad de proyecto****Ilustración 4.4 - Gráfico: Horas por iteración**

Observando la ilustración anterior, se puede apreciar el número de horas totales por iteración que el alumno ha invertido en la elaboración del presente proyecto, incluyendo la iteración inicial, iteración 0, y la totalidad de la iteración 11, donde mitad estaba destinada a finalizar la fase de producción y la otra parte restante a elaborar y completar la memoria final del proyecto.



Ilustración 4.5 – Dedicación por fases

En la Ilustración 4.5, se simplifica en valores porcentuales el tiempo dedicado a cada una de las fases esenciales de este trabajo final de máster, siendo la fase de producción a la que mayor tiempo se ha destinado.

Por último, comentar que, a pesar de haber tenido un gran control sobre el tiempo dedicado al proyecto para evitar superar las 300 horas de trabajo, finalmente se ha producido un exceso de 9 horas y media. De las cuales 2 se consumieron en la fase de desarrollo y el resto en la elaboración y corrección de la presente memoria.

5. Producción y desarrollo

Al estar trabajando con una metodología ágil, la fase de producción se desarrollará de manera evolutiva e incremental, permitiendo realizar y detectar posibles cambios, y/o nuevas necesidades del proyecto.

Por ello, se ha decidido exponer en esta memoria la implementación del prototipo siguiendo el orden de los pasos, en la evolución que se ha experimentado, durante el desarrollo del mismo.

5.1. Descarga, lectura y visualizado de la nube de puntos

Se inicia el proyecto buscando en la red proyectos de carga y visualizado de nubes de puntos, localizando un plugin, “Point Cloud Free Viewer” [40] en la Asset store de Unity [41] capaz de leer los puntos de una nube almacenada en local y visualizarlos en la escena.

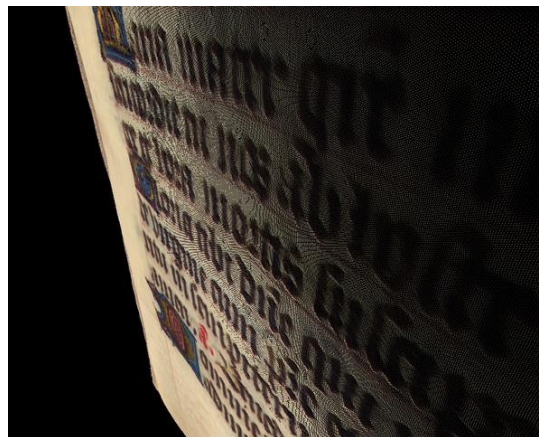


Ilustración 5.1 - Ejemplo de nube de puntos con Point Cloud Free Viewer

Tras adquirir el plugin, se procede a investigar su código fuente para entender el proceso de lectura y representación de los puntos de la nube.

Durante esta tarea, se aprecia que el proceso de lectura del script comienza con un filtro de extensión de la nube, admitiendo sólo el formato OFF, y continua con el almacenaje de los puntos leídos en una lista de objetos Vector3 [42], a cada componente del punto se le multiplica su valor por la escala, definida al inicio del script.

Los colores de cada punto son almacenados en una lista de objetos tipo Color [43], y se almacenan los valores RGB pasando el valor de cada componente del rango [0,255] al [0,1].

El script sigue con un reparto de grupo de puntos por malla, Mesh [44], creando grupos de 65000 puntos por malla.

Esto se debe a que las mallas en Unity tienen un límite de 65535 vértices [45], por defecto Unity aplica la configuración GPU de 16bits, ya que no todas las GPU's soportan el límite de 32bit, que permitiría almacenar hasta 4 millones de vértices, como es el caso de los dispositivos Android con GPU Mali-400.

De modo que se tiene que calcular cuantas mallas son necesarias para almacenar todos los puntos leídos, e ir creando mallas por cada 65000 puntos, utilizando una cifra más redondeada.

Cuando ya se han almacenado todos los datos en sus respectivas listas, estas se asocian a los vértices y colores de la malla, y esta a su vez es asignada a un GameObject [46], ya que es la clase que Unity utiliza para representar objetos tridimensionales en el espacio, que se ancla en un GameObject que actuará de objeto padre.

Al finalizar el proceso anterior, se dispone en la escena de la nube de puntos cargada en un GameObject-padre, compuesto por varios GameObject's-hijos, véase la Ilustración 5.2.

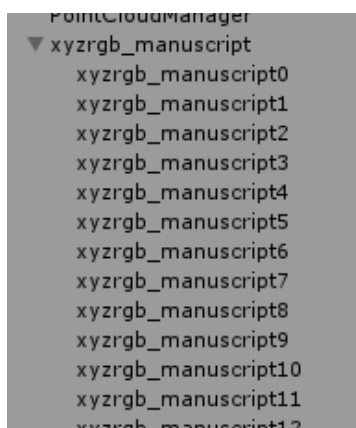


Ilustración 5.2 - Ejemplo de jerarquía de la nube de puntos con "Point Cloud Free Viewer"

Habiendo entendido todo lo anterior, se pretende tomarlo como referencia para definir un script propio y hacerlo compatible con el formato PLY, que es con el que se trabajará en este proyecto.

5.1.1. Elementos esenciales para Hololens

Antes comenzar a programar nada en la escena, se deben importar un conjunto de elementos imprescindibles para aplicaciones Hololens, Ilustración 5.3.

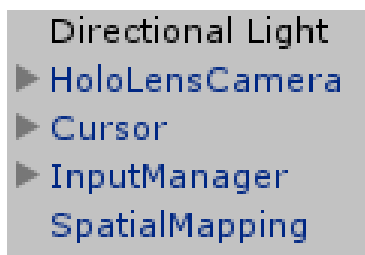


Ilustración 5.3 - Elementos imprescindibles para Hololens

“HoloLensCamera” es un prefab del paquete MRTK que contiene una cámara configurada con las características adecuadas para Hololens y que contiene a GazeControls que actúa como gestor de controladores como gamepad, teclado, etc...

El objeto “Cursor”, es un objeto que permitirá detectar dónde y en que objeto se encuentra el foco de atención para poder interactuar con él, y que se desplaza con el movimiento de la cabeza del usuario, manteniéndose siempre en el centro de la visión.

“InputManager” es el objeto que contiene los scripts encargados de gestionar todos los eventos de entrada de gestos, toques, etc...

El “SpatialMapping” es el encargado de reconocer el entorno, obteniendo un modelo de la habitación real en la que se está trabajando.

Este conjunto de elementos, sin indispensables para el correcto funcionamiento de cualquier aplicación para Hololens, por lo que es lo primero que se tiene que añadir un script a todos ellos para evitar que sean destruidos al inicio de cada escena.

Seguidamente, se define la escena como inicial, que nada más ejecutarse, invoca a la escena “main”, donde se ha desarrollado la primera fase del proyecto.

De este modo, siempre se dispone de los mismos elementos esenciales evitando problemas de pérdida de referencias entre escenas, aunque cuando haya que utilizar alguno de sus componentes se tendrá que utilizar la función Find [47] de la clase GameObject.

5.1.2. Lectura y visualizado de la nube de puntos

Tomando como nubes de prueba los modelos disponibles en la web “The Stanford 3D Scanning Repository” [48], y observando la cabecera de los ficheros se puede conocer la estructura de los datos que contiene una nube de puntos con formato PLY, Ilustración 5.4, se procede a montar un script de lectura similar al anteriormente estudiado.

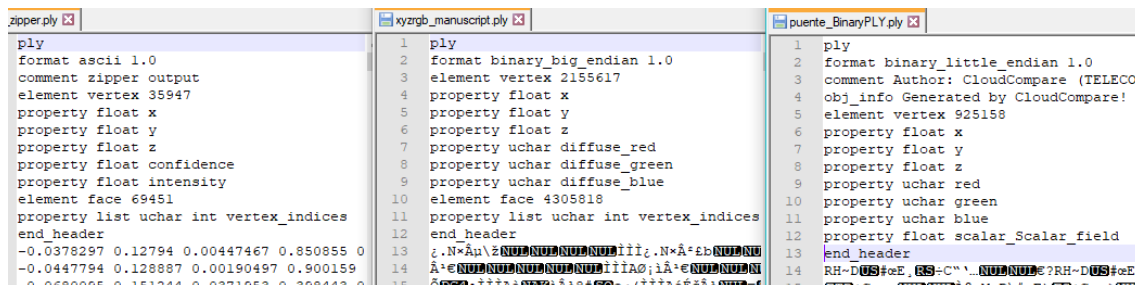


Ilustración 5.4 - Ejemplos de cabeceras de nubes de puntos PLY

Como se puede apreciar, la cabecera está delimitada por una línea inicial que indica tipo de fichero y una línea final, que indica donde termina la cabecera y empiezan los datos.

También se indica el total de vértices de la nube y el formato del contenido de los datos, pudiendo ser ASCII [49], binario en big endian o binario en little endian [50].

En cuanto a las propiedades de los puntos, se presta atención en los componentes de posición y color, que en algunos casos no están presentes, y el tipo de dato, como flotantes, uchar, etc...

Con todo ello presente se diseña en primer lugar la lectura de la cabecera del fichero de la nube de puntos, véase la Ilustración 5.5, donde se muestra dicho paso en pseudocódigo.

Se adelanta trabajo, identificando si se trata de un documento binario y de qué tipo, ya que para comenzar se trabajará utilizando un fichero en ASCII y posteriormente se incluirá la interpretación de los datos en binario.

También se decidió considerar el fichero no válido si el primer “element” leído no era el que indicaba el número de puntos de la nube, dato esencial para poder leer correctamente el fichero, para evitar conflictos con otros posibles “element” o la inexistencia de dicho dato en el fichero.

```

linea=leerLinea();
valido = true;
Si (linea == "ply")
{
    Si (linea != "end_header" && valido==true)
    {
        linea = leerLinea();
        linea.split();
        Si (linea[0] == "format" && linea[1] == "ascii")
        { isBinary=false;}
        Sino Si(linea[0] == "format" && linea[1] == "binary_big_endian")
        {
            isBinary=true; isLittleB=false;
        }
        Sino si(linea[0] == "format" && linea[1] == "binary_little_endian")
        {
            isBinary=true; isLittleB=true;
        }
        Sino
        {
            Si (linea[0] == "element")
            {
                Si (linea[1] == "vertex" && !primerElementLeido)
                { totalPuntos = linea[2];}
                Sino si (!primerElementLeido)
                {valido = false;}

                primerElementLeido=true;
            }
            Sino si(linea[0] == "property")
            {
                identificación de tipo de datos y su posición en cada linea
            }
        }
    }
}

```

Ilustración 5.5 – Pseudocódigo: Lectura de cabecera PLY

La parte de identificación de tipo de datos y posición en cada línea, es el trozo de código que se encargará de identificar el tipo de datos de cada componente del punto y su posición en la línea leída, para poder obtener su valor rápidamente al aplicarle a esta la función Split, la cual convierte la cadena en una lista de subcadenas cortadas por los espacios en blanco.

En el caso de los formatos binario esta parte es más compleja y se explicará más adelante, pues en este punto del proceso de desarrollo la atención está fijada en los formatos en ASCII.

De modo que el siguiente paso a dar, es lanzar la función de lectura de las líneas de datos de puntos. Esto se realizará en un hilo con una variable semáforo que indicará cuando ha finalizado la lectura, ya que, debido a la cantidad de datos y el coste que requiere el proceso de lectura, se producen congelación de frames [51] dando la sensación al usuario de que la aplicación se ha parado.

```
Si (isBinary)
{ LanzaHilo (leePuntosBinariy);}
Sino
{ LanzaHilo (leePuntos);}

Mientras (!lecturaTerminada)
{ actualiza frame en hilo principal}
```

Ilustración 5.6 - Seudocódigo: Hilos para lectura de puntos

En el módulo “leePuntos”, se continúa leyendo el fichero PLY utilizando la información, tipo y posición de los datos, obtenida en la lectura de la cabecera para interpretar la información de los puntos.

Dicha información es grabada en un objeto la clase PointCloud, clase propia cuya estructura está disponible en la Ilustración 5.7, y dichos objetos son almacenados en una lista para su posterior carga en mallas.

```

Class PointCloud
{
    int id;//índice de lectura desde el fichero cargado
    double x;
    double y;
    double z;
    bool color;
    Color miColor;
    int miMalla = -1; //índice de la malla
    int id_puntoMalla = -1; //índice dentro de la malla
}

```

Ilustración 5.7 - Seudocódigo: Estructura de la clase PointCloud

Durante la lectura de puntos, se irá calculando los máximos y mínimos de la nube con las coordenadas de cada punto, ya que se utilizará después para escalar la nube en función de su eje más amplio y para poder centrar las mallas al eje de coordenadas del objeto padre.

```

leePuntos()
{
    for (i=0 to i=TotalPuntos)
    {
        linea = leeLinea();
        coordenadas = linea[posicionCoordenadas];

        calculaMaximosyMinimos

        si (haycolor)
        { color = linea[posicionColor] / 255; //transforma en rango [0,1]}
        sino
        { color = magenta;}

        listaPuntos.add(new PointCloud(coordenadas, color));
    }
    lecturaTerminada=true;
}

```

Ilustración 5.8 - Seudocódigo: Lectura de datos de puntos en ASCII

Seguidamente, tras finalizar el proceso de lectura, se procede al reparto de puntos en mallas calculando cuantas se necesitarán e instanciando una nueva malla por cada grupo y asignándole el material que lleva asociado el shader [52] necesario para representar correctamente los colores de los puntos de la nube, véase Ilustración 5.9.

```

GameObject nube;
limite=65000
grupos= totalpuntos/limite;

for(i=0 to i=grupos)
{ instancia malla(i, limite);}
instancia malla(grupos-1, totalpuntos - ((grupos-1)*limite)); //resto de puntos

instancia malla(idMalla, numPuntos)
{
    gameObject grupo;
    grupo.material = materialPunto;

    Mesh malla = creaMalla(idMalla, numPuntos, limite);

    grupo.malla = malla;
    grupo.parent = nube;
}

creaMalla(id, npuntos, limite)
{
    mesh malla;
    misPuntos[];
    misIndices[];
    misColores[];

    for (i=0 to i=npuntos)
    {
        misPuntos[i] = CoordenadasPunto;
        misColores[i] = ColoresPunto;
        misIndices[i]=i;
    }

    malla.vertices = misPuntos;
    malla.colors = misColores;
    malla.setIndices(misIndices, MeshTopology.Points, 0);
    return malla;
}

```

Ilustración 5.9 - Seudocódigo: Reparto de puntos y generación de objeto nube

Para finalizar, se toman los valores mínimos y máximos de las coordenadas que se calcularon durante la lectura de puntos, para calcular el punto central de cada coordenada.

Para ello, solo hay que restar el valor mínimo al valor máximo, al resultado dividirlo entre 2 y, después, sumar el valor mínimo, todo ello por cada coordenada.

Una vez se dispones de valores centrales para cada coordenada, se aplica una traslación de cada hijo utilizando dichos valores, de modo que estos quedan centrados al eje de coordenadas del objeto padre.

Y, por último, aplicamos al padre la escala que se calculó anteriormente, de esta forma todos los hijos se escalan por igual sin producir distorsiones, Ilustración 5.10.

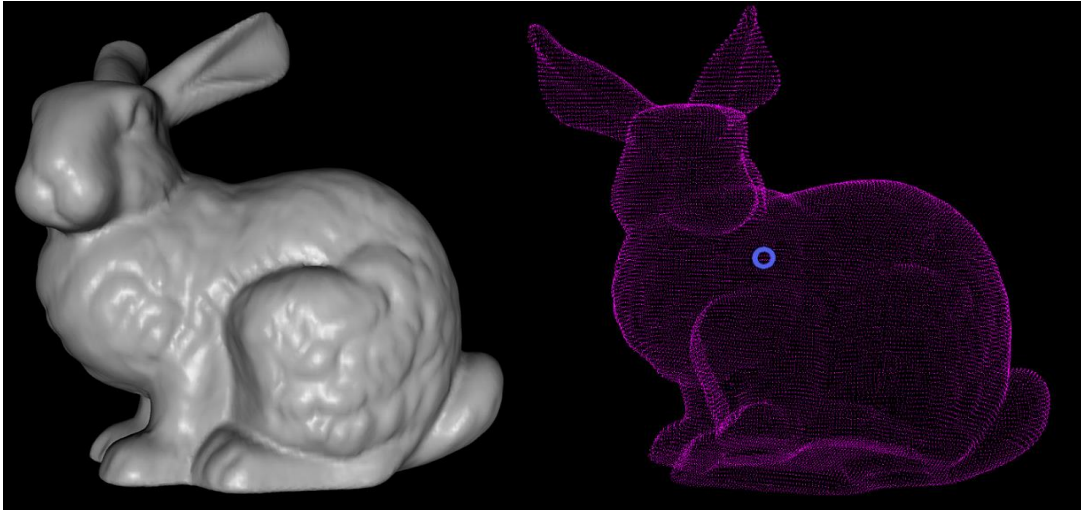


Ilustración 5.10 - Ejemplo de nube de puntos cargada

5.1.3. Descarga de la nube

Hasta el momento, se ha trabajado con una nube de punto almacenada localmente, pero acceder a ruta local no es sencillo en aplicaciones Universal Windows Platform, por lo que las nubes se obtendrán siempre desde un servidor online, al menos en este prototipo.

Para ello, se necesita disponer de un servidor donde alojar las nubes de puntos, que después se descargaran en la aplicación a través de la librería UnityWebRequest [53] en un MemoryStream [54] que sustituirá a la ruta local en el lector.

Al igual que ocurriría con la lectura de puntos, el proceso se realiza en una corrutina, Coroutine [55], ya la descarga de nubes muy pesadas puede provocar congelación de frames.

5.1.4. Uso de entorno y gesto "Tap"

El siguiente paso es investigar el modo de centrar la nube en el espacio de trabajo, es decir, cuando el usuario cargue la nube esta aparezca en el centro de la habitación en la que se encuentra.

De otro modo, la posición de la nube es relativa a la posición de las Hololens, que están definidas como eje de coordenadas del escenario virtual, pudiendo darse el caso de que el usuario este demasiado cerca de una pared y la nube sea colocada fuera del alcance de este, Ilustración 5.11.



Ilustración 5.11 - Comparativa de mala (izq.) y buena (der.) ubicación de modelos

Para poder obtener el centro, se hará uso de un prefab disponible en el paquete MRTK llamado “SpatialUnderstanding” [56], que junto con el “SpatialMapping”, es capaz de identificar paredes, techo y suelo además de otras superficies de forma óptima y limpia.

“SpatialUnderstanding” además permite activar y desactivar el sistema de escaneado del entorno, pudiendo liberar recursos del sistema y proporciona datos que pueden ser útiles para establecer límites.

En el presente caso, se establecerá un mínimo de superficie detectada para poder pasar al siguiente paso, de esta forma se asegura disponer de una cantidad fiable de datos sobre el entorno para poder operar con ellos.

Una vez ha finalizado el proceso de escaneado, se toma el objeto “SpatialUnderstanding”, se recorren sus hijos en busca de los puntos centrales de cada uno proporcionados por sus componentes “Collider” [57], y se calculan los valores máximos y mínimos, con los que se obtendrá el punto central del entorno, Ilustración 5.12.

```
Max = valor minimo de sistema;  
Min = valor máximo de sistema;  
puntoMedio = nulo;  
CalculaCentro()  
{  
    hijos = SpatialUndertanding.total_hijos;  
    for(i=0 to i=hijos)  
    {  
        h = hijo.gameObject;  
        centroHijo = h.colider.bounds.center;  
  
        compara max y min con centroHijo  
        y almacena si cumplen condiciones  
    }  
    puntoMedio = calcula punto medio entre puntos Max y Min;  
}
```

Ilustración 5.12 - Seudocódigo: Calcula centro del entorno

A continuación, se diseña la interfaz de todo el sistema en un storyboard, para tener presente dicho diseño cuando se desarrolle cada interfaz de usuario, véase Anexo 7.3.2.

Seguidamente se implementa una pequeña interfaz de inicio, basada en el storyboard diseñado previamente, con modelos 3D como transición entre el paso del escaneado y el siguiente, para que el usuario perciba que termina un proceso y empieza otro.

Dicha interfaz consta de un texto 3D y un cubo dispuesto a modo de botón. El texto informa de si se ha alcanzado o no el área mínima detectada y cuando esta se alcance, se muestra el botón el cual al interactuar con él, da paso a la siguiente función del proyecto, Ilustración 5.13

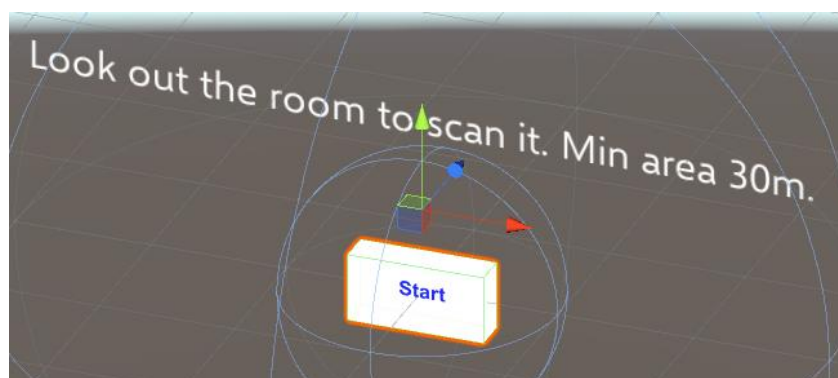


Ilustración 5.13 - Interfaz de inicio

La interacción con el botón se realizará a través del gesto “Tap” cuyo evento, “OnInputClicked”, se captura utilizando la interface “IInputClickHandler”.

Al desencadenarse dicho evento, realizando el gesto “Tap” en el botón, se para el proceso de escaneado del entorno, calculando el punto central como se explicó anteriormente, se deja de dibujar las líneas del “SpatialUnderstanding” y se cede paso al siguiente procedimiento.

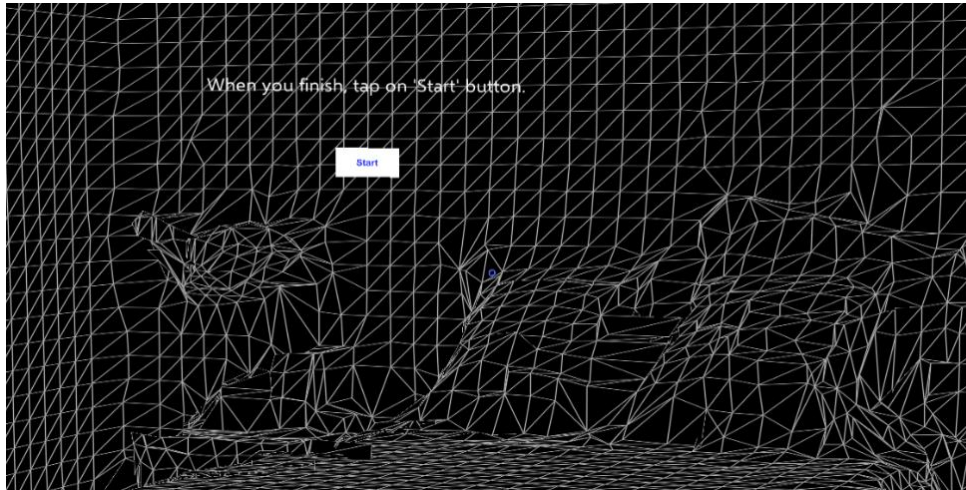


Ilustración 5.14 - Ejemplo de escaneado de entorno e interfaz de inicio

5.1.1. Selección de nubes

Para el siguiente paso, se ha requerido diseñar e implementar una interfaz de usuario que permitirá a este seleccionar que nube quiere visualizar.

El diseño planeado para esta interfaz consiste en un Combo Box, un listado de elementos, cuyos elementos sean seleccionables a modo de botón, y otro par de botones en la parte inferior para lanzar la carga de la nube seleccionada o salir de la aplicación.

Tras elaborar el elemento combobox siguiendo un sencillo tutorial [58], se genera un panel con un par de botones, para iniciar la lectura o cerrar la aplicación, al que añadiremos dicho combobox.

Para obtener la lista de nubes subidas al servidores, se utilizara un sencillo script PHP que devolverá una cadena con todos los ficheros disponibles en el directorio indicado, Ilustración 5.15.

```
<?php
$directorio = opendir("clouds");
while ($archivo = readdir($directorio))
{
    if (is_dir($archivo) == false)
    {
        echo $archivo . PHP_EOL;
    }
}
?>
```

Ilustración 5.15 – Script PHP que devuelve lista archivos PLY

Con dicho script en el servidor, basta con realizar una petición con un UnityWebRequest a la ruta del archivo php para obtener la lista de nubes. Seguidamente, se genera un botón por cada elemento de dicha lista, dentro del combobox. Véase el resultado final en la Ilustración 5.16.

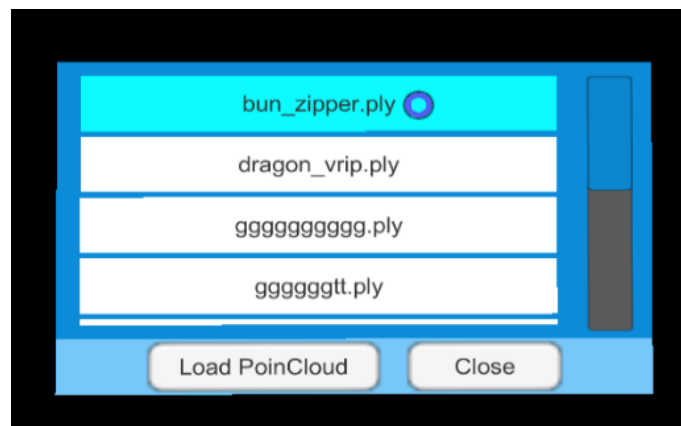


Ilustración 5.16 - Menú selección de nube

5.1.2. Compatibilidad con PLY binario

En este punto del proyecto se decide separar en escenas la parte principal y la parte de lectura y visualización, de modo que el proyecto quede conformado por tres escenas: la escena inicial donde se inicializan los elementos esenciales de Hololens, la escena principal, donde se realiza el reconocimiento del espacio y la selección de la nube a cargar, y la escena de la aplicación, donde estarán todas las funciones de trabajo sobre la nube incluyendo la lectura de la misma.

De este modo, el punto central y el nombre de la nube seleccionada son enviados a la escena de la aplicación, donde se descarga y lee para después mostrarla en la ubicación del punto central recibido.

En el caso de nubes con grandes cantidades de puntos, durante proceso de descarga y lectura de la nube, se ve la necesidad de indicarle al usuario información sobre lo que internamente está ocurriendo, pues si no es así este puede pensar que la aplicación se ha detenido.

Por ello, se implementa una interfaz de usuario, basada en el storyboard diseñado previamente, donde el usuario podría no solo conocer el estado del proceso de descarga y lectura, sino tener también la oportunidad de cambiar de nube, cerrar la aplicación, visualizar la nube ya preparada o reintentar el proceso de descarga y lectura en caso de producirse algún error.

En el diseño se piensa en utilizar un modelo 3D de reloj de arena animado, el cual girará sobre sí mismo mientras dure el proceso de descarga y lectura, y parará cuando este finalice, mostrando además los tres botones, interactivos con el gesto “Tap”, que estarán ocultos el resto del tiempo.

A este símbolo se le añade un mensaje de una palabra para indicar que proceso se está ejecutando en ese momento, Ilustración 5.17.

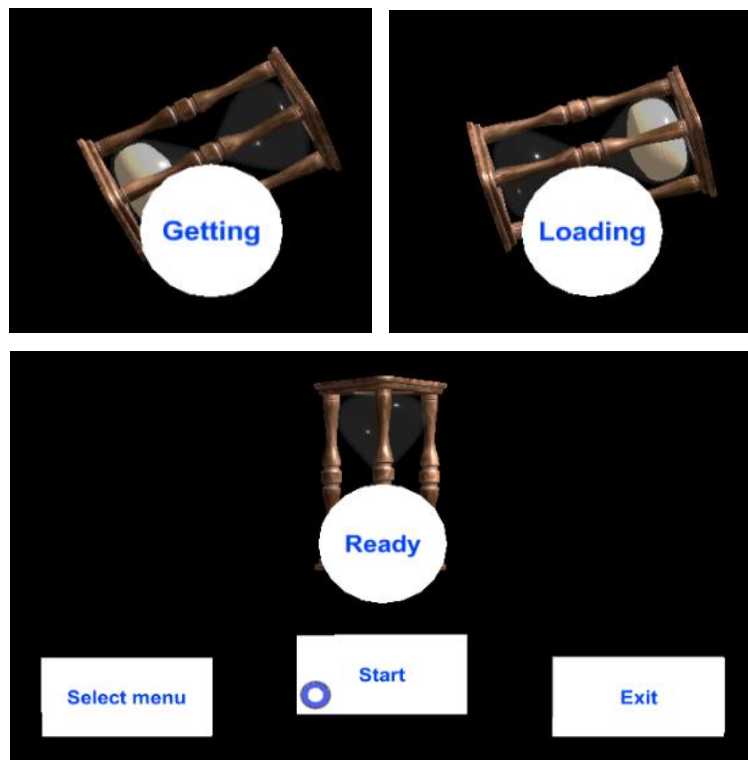


Ilustración 5.17 - Ejemplos de menú de carga de nube

Con esto ya implementado, se decide actualizar el sistema de lectura de puntos para hacerlo compatible con los ficheros PLY binarios.

Se comienza añadiendo en la lectura de la cabecera del fichero PLY, en la parte del código donde se interpreta los tipos de datos de los campos “property”, un identificador para cada tipo de dato, de modo que se pueda calcular el tamaño en bytes de las líneas que contienen datos de los puntos, Tabla 5-1.

Tipo de dato	Bytes	identificador
char	1	0
uchar	1	1
short	2	2
ushort	2	3
int	4	4
uint	4	5
float	4	6
double	8	7

Tabla 5-1 - Bytes e identificadores de tipos de datos

De modo que se almacena la posición de cada componente del punto y el identificador y posición de su correspondiente tipo de dato. Con esto, y conociendo el total de bytes que representaría la información de cada punto, se puede proceder a leer el fichero en formato binario.

Se debe tener en cuenta que deben descartarse el total de caracteres que contiene la cabecera, para inicializar la posición del lector en el primer dato binario del documento.

Una vez el lector esté preparado, se leerán el total de puntos aplicando un incremento del tamaño en bytes de cada punto, permitiéndole pasar de un punto a otro.

Por cada punto se realizará una traducción de bytes según el identificador del tipo de datos y posición, que se almacenó en la cabecera, tanto para las componentes de posición como las componentes de color.

El resto del proceso es idéntico al sistema de lectura de puntos en formato ASCII, véase un ejemplo de nube PLY binaria en la Ilustración 5.18.



Ilustración 5.18 - Ejemplo de nube de puntos en PLY binario

5.2. Diseño y desarrollo de herramientas

En este punto del desarrollo, se van a implementar las funcionalidades que el usuario puede llevar a cabo sobre una nube de puntos ya cargada.

Para ello, se diseña un menú 3D de herramientas, compuesto por tres grupos de herramientas: Herramientas de sistema, herramientas de nube y herramientas de puntos.

Como se puede apreciar en storyboard del anexo 7.3.2, cada grupo de herramientas tiene un icono 3D que se diseñará y animará utilizando las herramientas y modelos simples de Unity.

Cada uno de los iconos 3d de este menú, tienen un conjunto de botones que solo será visible cuando se interactúe con ellos con el gesto “Tap”, pero ese no es el único gesto con el que se puede interactuar con estos iconos.

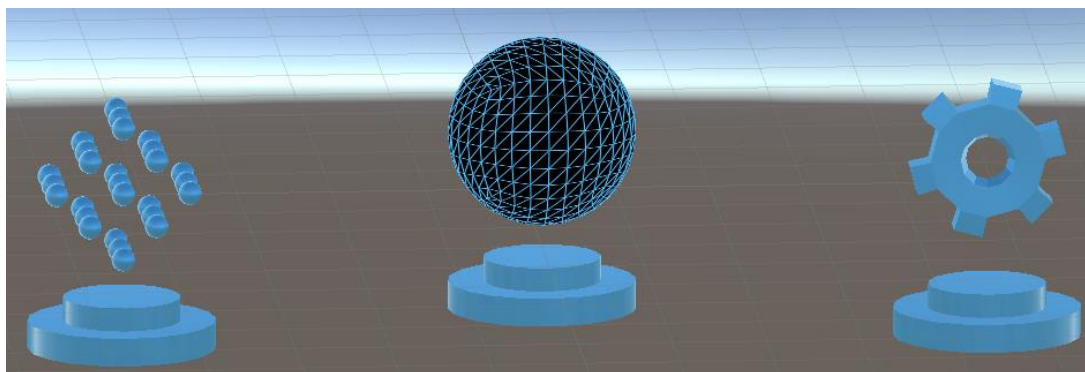


Ilustración 5.19 - Menú de herramientas

En esta ocasión, es posible que el usuario necesite poder desplazar, rotar y escalar estos iconos para adaptarlos mejor a su espacio de trabajo, por lo que se va a establecer el gesto “Double-Tap” como gesto que activa el estado de manipulación de los iconos.

Tanto el gesto “Tap” como el “Double-Tap” desatan el evento “OnInputClicked”, pero la particularidad es que el evento no es capaz de diferencia entre ambos gestos.

Por lo que se define un sistema que permite identificar cada gesto, evitando confusiones véase el apartado 5.4.7, de modo que cuando se produzca un “Tap” se desplegará el menú y se ocultarán el resto de iconos, y cuando se detecte un “Double-Tap” se activará y desactivará el modo edición de ese icono 3D.

El modo edición consiste en activar la edición en el script de MRTK llamado “BoundingBoxRig”, previamente asociado al gameObject del icono 3D. A este script hay que asociarle, desde la pestaña del inspector, el prefab de MRTK del “BoundingBoxBasic”, pero no el prefab del AppBar ya que no será de utilidad en este proyecto. De modo, que se editara el script para controlar la excepción de que el prefab de Appbar es nulo, y genere solo un BoundingBox.

Con ello, el usuario podrá rotar y escalar el icono 3D, pero para poder desplazarlo se deberá asociar a dicho icono el script “HandDraggable”, permitiendo entonces que el objeto sea movable.

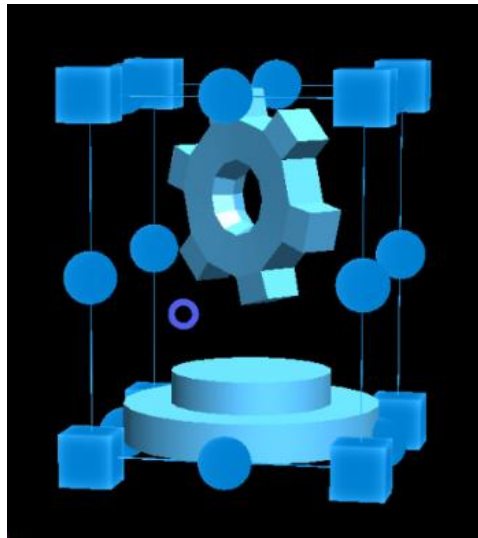


Ilustración 5.20 - Ejemplo de uso de BoundingBoxRig

5.2.1. Herramientas de sistema

El diseño del icono de las herramientas de sistema se ha diseñado con un par de cilindros como base y un modelo de engranaje sencillo [59], ya que es este el símbolo más utilizado en las aplicaciones para representar las funciones relativas al sistema.

A continuación, atendiendo a lo diseñado en el storyboard del Anexo 7.3.2, se crean los botones 3D que se desplegarán al realizar un toque en el icono y permitirán al usuario utilizar las funciones respectivas al sistema, Ilustración 5.21.

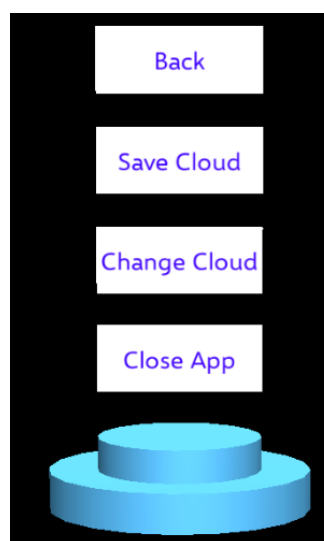


Ilustración 5.21 - Menú Herramientas de sistema desplegado

Algunas de estas funcionalidades son de implementación simple, como por ejemplo “Close App” que es una simple llamada a “Application.Quit()”, la cual cierra la aplicación.

“Back” es una función común en todas las herramientas definidas en el menú de herramientas, que consiste en ocultar los botones desplegados y volver a dejar visible el modelo 3D del icono correspondiente y habilitar el resto de iconos del menú de herramientas.

“Change Cloud” permite al usuario seleccionar otra nube de puntos para visualizar, para ello simplemente realiza un cambio de escena pasando de la actual a la escena principal, donde se halla el menú de selección de nubes de puntos.

Por último, la función “Save Cloud” es la más compleja de las presentes en las herramientas de sistema.

El proceso comienza desplegando la interfaz para la subida de la nube de puntos visualizada en la escena, permitiendo escribir al usuario un nuevo nombre para la nube a través del campo de texto y el prefab del teclado de MRTK [60], Ilustración 5.22

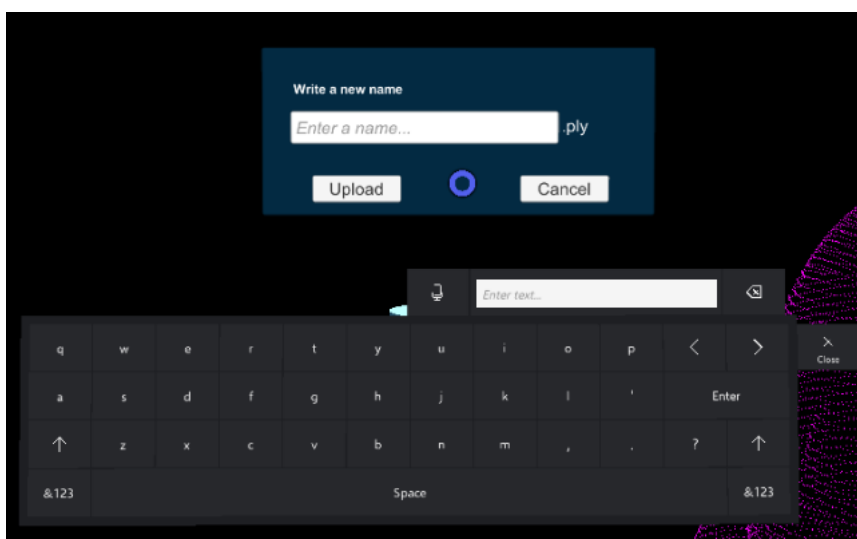


Ilustración 5.22 - Panel de guardado de nube con teclado desplegado

Una vez el usuario indica el nombre el nuevo nombre de la nube, pulsará sobre el botón “Upload” para iniciar el proceso interno de guardado.

Este proceso prepara una cabecera para el nuevo fichero PLY en formato binario, para que el fichero resultante sea lo más liviano posible, indicando el tipo de dato de los componentes de los puntos, que en el caso de las coordenadas será “float” y en el caso de los colores “uchar”.

Una vez se tiene definida la cabecera, se genera un listado con todos los puntos del octree [61] y se recorren uno por uno, filtrando si están marcados como borrados o no, siendo descartados de ser almacenados en el fichero en caso afirmativo.

También se dispone, tanto en la cabecera como en el recorrido de puntos, un filtro para detectar si la nube tiene o no color, por lo que sólo se escriben datos referentes al color en caso afirmativo.

Cabe indicar que, como se está generando un fichero con información binaria, los datos de los puntos recorridos deben pasar por una función que realiza la traducción de sus valores a formato binario antes de ser almacenados en el archivo.

Al final del proceso, se realiza la subida a la nube utilizando las librerías “WWWForm” y “UnityWebRequest”, utilizando una URL del servidor a un archivo PHP que tomará el fichero y lo añadirá al directorio de nubes en el servidor [62], Ilustración 5.23.

```
<?php
$directorio = "./clouds";
if ($_FILES["file"]["error"] == UPLOAD_ERR_OK) {
    $tmp_name = $_FILES["file"]["tmp_name"];
    $name = $_FILES["file"]["name"];
    move_uploaded_file($tmp_name, "$directorio/$name");
}
?>
```

Ilustración 5.23 - Script PHP que recoge el fichero PLY y lo almacena en el servidor

Mientras todo lo anterior se produce de forma interna, el usuario es informado del estado del proceso en el panel de subida, en el cual se actualizarán los mensajes según dicho estado.

5.2.2. Herramientas de nube

El diseño del icono de las herramientas de nube se ha diseñado con un par de cilindros como base y una esfera a la que se le ha aplicado una copia del material “Wireframe”, utilizado en el “SpatialUnderstanding”, con una modificación de color para que concordara con el utilizado en el icono.

En esta ocasión sólo se dispondría de las funciones “Back” y “Manipulate”, esta última permitirá al usuario manipular la nube completa, es decir, poder desplazarla, rotarla y escalarla fácilmente, Ilustración 5.24.

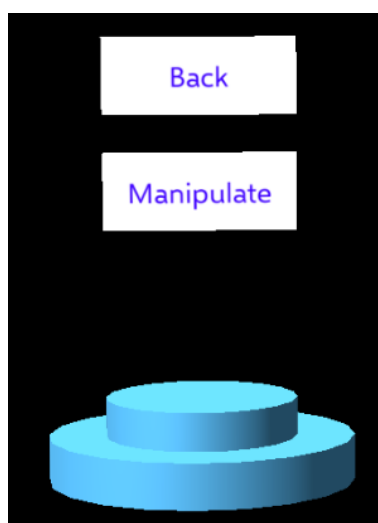


Ilustración 5.24 - Menú de Herramientas de nube desplegado

Dicha manipulación se hace efectiva a utilizando un objeto pivote, cuyo diseño se ha realizado con la intención de que sirva al usuario como medio orientativo, pues sus “aletas” representan cada eje del espacio, siendo el rojo el eje X, el azul el eje Z y el verde el eje Y, Ilustración 5.25.

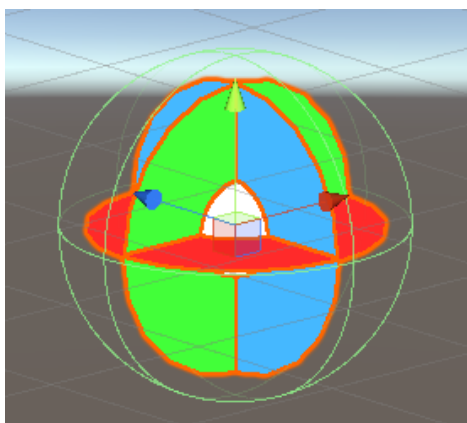


Ilustración 5.25 - Pivote de manipulación de nube

Este objeto, a diferencia de los elementos intractuables vistos hasta ahora, es interactuable con dos manos y no requiere de un “BoundingBoxRig”.

Al iniciar “Manipulate”, el pivote se coloca sobre el origen de coordenadas de la nube, Ilustración 5.26. Y cuando el usuario realiza el gesto de “Hold” con las dos manos, se activa el gesto de manipulación bimanual sobre el pivote.

De este modo, el usuario realmente está manipulando el pivote, pero durante la manipulación bimanual, todas las transformaciones que sufra serán proyectadas de forma proporcional a la nube, evitando que el usuario realice gestos muy amplios, pues ello provoca la cancelación del gesto debido a que las manos se alejan del rango de seguimiento de los sensores.



Ilustración 5.26 - Ejemplo de nube de puntos en modo manipulación

Finalmente, se dispone de otro botón “Done”, solo visible mientras esté activa la función “Manipulate”, y cuando se pulsa, este desactiva el modo manipulación, ocultando de nuevo el pivote.

5.2.3. Herramientas de puntos

El diseño del icono de las herramientas de puntos se ha diseñado con un par de cilindros como base y un conjunto de pequeñas esferas dispuestas como un cubo.

En este menú, además de “Back”, se dispone de herramientas como “Select points” y “Remove points”, Ilustración 5.27, cuyas funcionalidades están estrechamente ligadas. Por ejemplo, si no se ha seleccionado primero los puntos, la herramienta “Remove points”, no hará nada.

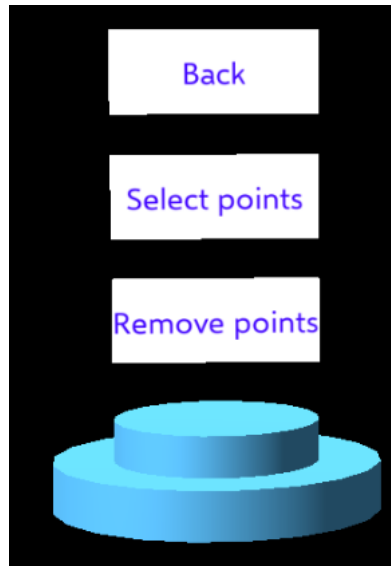


Ilustración 5.27 - Menú Herramientas de puntos desplegado

“Select points” funciona a través de una esfera, que servirá a modo de brocha, utilizando su área y punto central para la búsqueda de puntos.

Teniendo en cuenta todas las transformaciones que ha sufrido la nube hasta el momento de la selección, tanto la nube como las mallas, y que en Unity las posiciones de los objetos hijos son relativas a la posición del padre, se debe aplicar dichas transformaciones a la esfera, para que la búsqueda sea efectiva. En resumen, hay que hallar la transformación entre sistemas de coordenadas de las mallas y el del octree.

Dicha búsqueda se realizó sobre un octree con los puntos de la nube, pues indicando un punto y un radio, la librería que se utilizó en este proyecto podía devolver el conjunto de puntos que encontraba dentro de dicha área.

Una de las cosas que se debió hacer para poder hacer uso del octree, fue actualizar la clase “PointCCloud”, para que incluyera un método de comparación “Equals()” y un par de variables booleanas “is_selected” y “is_deleted”, de cara a la función de selección de puntos.

Con todo lo anterior claro, el proceso de selección comienza con un gesto “Hold” sobre la esfera de selección, la cual cambia de color, indicando que se ha activado la búsqueda interna de puntos en el octree.

Seguidamente, con el gesto “Manipulation” se consigue desplazar la esfera selectora con el movimiento y dirección de la mano. Durante este gesto, todos aquellos puntos de la nube que pasen por el área de la esfera quedaran seleccionados, “is_selected = true”, y cambiaran a color verde intenso.

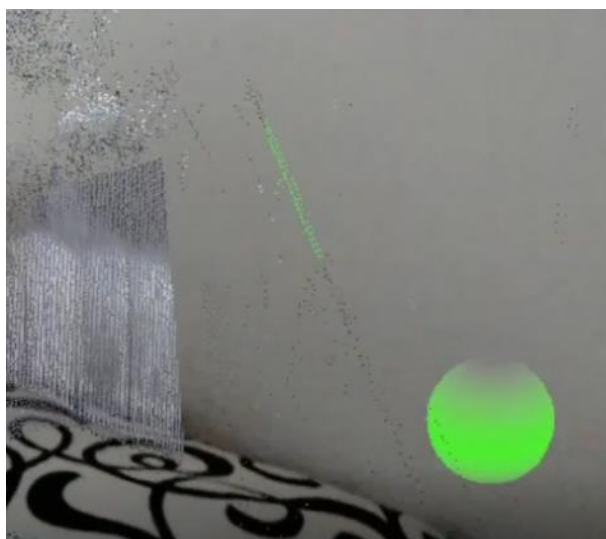


Ilustración 5.28 - Ejemplo de puntos seleccionados

Una vez el usuario finalice la selección de puntos, podrá o bien deshacer la selección, haciendo que los puntos marcados vuelvan a “is_selected=false”, o bien los elimine. En tal caso, los puntos marcados pasan de color verde a color transparente y la variable “is_deleted” recibe un “true”.

5.3. Sonidos

Para finalizar con el proyecto, se decide completar la experiencia de usuario añadiendo sonidos a la aplicación.

Se selecciona una melodía ambiental para utilizarla como sonido de fondo, “Cooperation road” disponible en la biblioteca gratuita de Youtube [63], el cual se reproducirá en bucle durante todo el tiempo que se esté ejecutando la aplicación.

El resto de sonidos se utilizará para botones al pulsarlos y avisos sonoros que indiquen aparición de paneles, botones, errores, etc, [64].

Estos sonidos son de carácter espacial, es decir, se reproducen en una ubicación concreta, dando al usuario la localización del objeto asociado o el evento sobre el que se quiere llamar la atención.

Para incluir los sonidos en las escenas, simplemente se necesita añadir la componente “Audio Source” [65] y la pista de audio q reproducirá.

5.4. Problemas enfrentados

Durante el desarrollo de este trabajo final de master se han presentado diversos problemas, fruto de los continuos avances de la tecnología elegida y de la escasa documentación disponible. Algunos de estos contratiempos se han resuelto exitosamente a base de constancia e investigación, pero otros han quedado pendientes de futuras versiones y/o actualizaciones de las herramientas de desarrollo.

5.4.1. Incompatibilidades y actualizaciones del software de desarrollo

Los primeros problemas que surgieron fue durante la fase inicial del desarrollo, donde la versión de Unity instalada inicialmente, la 2017.1.1f, mostraba errores de incompatibilidad con la cámara AR de Vuforia cuando se pretendía ejecutar el proyecto. Estos errores se solucionaron al actualizar Unity la versión 2017.2.0f3.

Al iniciar la instalación de dicha versión, se comprobó que el plugin de Vuforia con el que se iba a trabajar ya no era necesario, pues la nueva versión de Unity ya la tiene incorporada en su plataforma.

Al mismo tiempo, Microsoft actualizó la información del sitio web para desarrolladores de Hololens, incluyendo en la sección de herramientas de desarrollo el paquete de herramientas llamado “HoloToolKit” [66] para facilitar el desarrollo de aplicaciones para Hololens con Unity, que hasta el momento había estado en fase experimental.

Con esta nueva noticia, y con una versión estable de HoloToolkit, se decide descartar Vuforia como herramienta de desarrollo AR, ya que el nuevo paquete de librerías cubría todas las necesidades AR que tenía el proyecto a desarrollar.

Tras incorporar la librería HoloToolkit al proyecto, surge un nuevo problema de incompatibilidad, en esta ocasión con la versión de Windows 10, siendo necesario actualizar el sistema operativo del equipo a la nueva versión de Windows 10, “Windows 10 fall creators” [67].

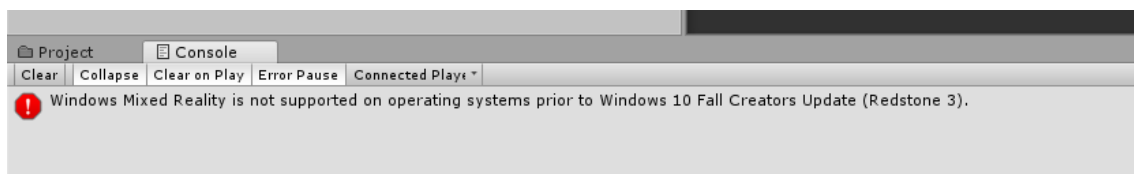


Ilustración 5.29 - Error necesario Windows 10 Fall Creators

5.4.1. Intérprete innecesario entre MonoDevelop y Visual Studio

Al intentar subir la primera versión estable de este proyecto a un repositorio en GitHub, para llevar un control de versiones, se pudo ver que el proyecto era increíblemente pesado, superando los 3.5Gb teniendo en cuenta que lo único que contenía este eran un par de scripts y el MRKT, cuyo peso descomprimido no superaba 200Mb.

Indagando un poco en las carpetas del proyecto, la mayor parte del peso estaba alojado en la carpeta donde se almacenaba la solución para compilar en Visual Studio.

Tras investigar dentro de la solución, se comprobó que había un conjunto de librerías que no tenía nada que ver con el proyecto. Estas librerías eran un intérprete del editor por defecto de Unity, MonoDevelop [68].

La solución a este problema pasaba por eliminar dicho editor del equipo, de modo que lo mejor era utilizar una versión beta de Unity, 2017.3 beta, que entonces era la única que permitía elegir si se quería disponer de MonoDevelop o no.

Tras el cambio de versión, se realizó una prueba construyendo una nueva solución y compilándola en Visual Studio. Esta funcionaba correctamente y su peso no superaba los 900 Mb. De modo que se llega a la conclusión de que el intérprete era totalmente innecesario.

5.4.2. Trabajando con varias escenas

Trabajar con más de una escena en Unity significa liberar recursos y tener poca carga en memoria. Sin embargo, las referencias de elementos esenciales para Hololens, por ejemplo, la cámara, se pierden y comienzan a generarse problemas como, fallos en la detección de gestos.

La única solución viable a este problema, es instanciar todos los elementos esenciales que pierdan la referencia en una escena inicial, y asociarles un script en el que se indique que no se destruyan en el evento “OnLoad()”, lanzado en el cambio de escena.

5.4.3. Trabas entre librerías Unity y librerías UWP (internet e hilos)

En las librerías de Unity existen hasta tres métodos diferentes para establecer una conexión a la red y descargar contenido, pero sólo uno es compatible con Universal Windows Platform, UnityWebRequest. Esto es algo que no se percibió hasta que se realizó la segunda depuración, es decir, la depuración en visual studio de la solución generada por Unity.

Otra librería con la que se ha tenido problemas ha sido “Thread”, que no es compatible con UWP, por lo que para poder trabajar con hilos se ha tenido que filtrar por sistema, para que en el caso de esta en UWP, utilizar la librería “Task” en lugar de “Thread”, Ilustración 5.30.

```

#if UNITY_EDITOR
using System.Threading;
#endif
#if WINDOWS_UWP
using System.Threading.Tasks;
#endif
.
.
.
#if WINDOWS_UWP
Task t = new Task(async()=>{LeePuntos();});
t.Start();
#endif

#if UNITY_EDITOR
Thread t = new Thread(LeePuntos);
t.Start();
#endif
.
.

```

Ilustración 5.30 - Ejemplo de incompatibilidad entre librerías

5.4.4. *Depuración y emuladores, Holographic Emulator y Hololens Emulator mienten*

Cuando se construye una aplicación en Unity, este realiza una compilación del proyecto y genera un ejecutable. Sin embargo, en el caso de aplicaciones para Windows Mixed Reality, Unity no genera un ejecutable directamente, sino que en su lugar se prepara una solución para Visual Studio, dónde se tiene que volver a compilar antes poder crear un paquete para la Windows Store, que es el formato que tienen las aplicaciones Universal Windows Platform [69].

Es durante este proceso cuando se hará uso de los emuladores para depurar antes de compilar. Y es donde se ha detectado que, aunque el proyecto parezca funcionar en el Holographic Emulator de Unity, puede no hacerlo tras construir la solución y ejecutarla en el Hololens Emulator, como ha sido el caso descrito en el punto 5.4.3.

Incluso, si la aplicación funciona correctamente en ambos emuladores, al crear el paquete para Windows Store e instalarlo en las Hololens, percatarse de que algo no va como se esperaba, por ejemplo, el caso comentado del punto 5.4.6.

De modo, que no se puede confiar al 100% en que todo lo que funcione en los emuladores funcionará correctamente también en las Hololens, haciendo que la depuración y el encontrar causas de errores sean muy complicado.

5.4.5. Paso de paquete de librerías: De HoloToolkit a MixedRealityToolkit

Al poco tiempo de iniciar el desarrollo, terminando las funcionalidad de visualizado de puntos, Microsoft incluyó una gran cambio en paquete de librerías HoloToolkit, incluyendo librerías compatibles con el desarrollo de aplicaciones para, sus nuevas gafas de realidad mixta [70], cambiando también el nombre del paquete a “MRTK” (MixedRealityToolkit).

Este nuevo paquete [71], también incluía mejoras de las librerías para Hololens y nuevas funcionalidades para Hololens como por ejemplo las librerías para manipular objetos utilizando de BoundingBox.

De modo que para poder hacer uso de esta versión se tuvo que volver a actualizar Unity a una versión más reciente. Además de actualizar el SDK de Windows, ya que para entonces se lanzó una nueva actualización forzosa de Windows 10 y con ella una nueva versión del SDK para Hololens.

5.4.6. Los Custom Shader no son estereoscópicos ni se incluyen en la solución de forma automática

Tras realizar la doble depuración y probar la aplicación en ambos emuladores, se instala la aplicación en las gafas y es cuando se percibe que las nubes de puntos sólo se renderizan en un ojo, causando agotamiento visual.

Investigando un poco por las redes, se descubre que la causa de ello es que el shader encargado de interpretar los colores de los puntos no está optimizado para ejecutarse sobre Hololens, de modo que se incluyó el código necesario para hacerlo compatible [72].

Además, en la configuración de gráficos del proyecto, “Graphics Settings”, se tienen que añadir todos los custom shader que tenga el proyecto para Hololens, si no este no será importado, Ilustración 5.31.

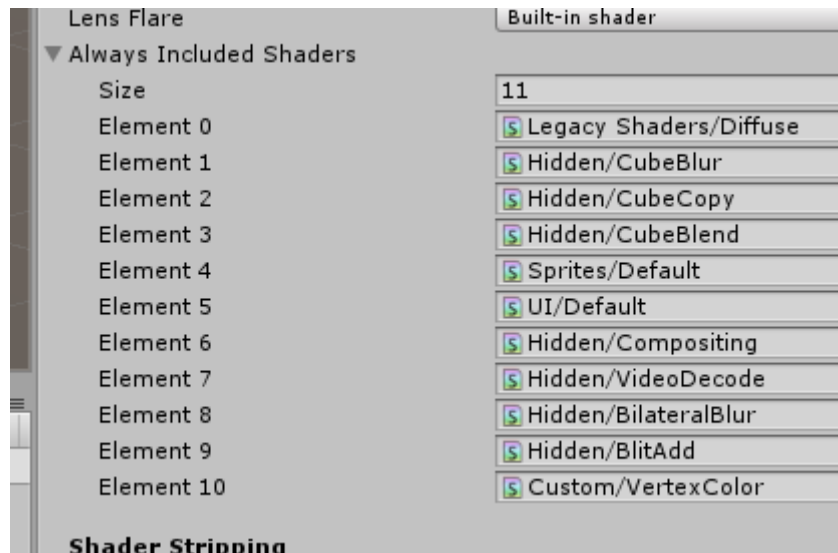


Ilustración 5.31 - Listado de shaders del proyecto

5.4.7. Contratiempos con el gesto Double tap

El gesto de “Double Tap” ha sido el gesto más problemático, ya que para poder utilizarlo hay que editar el código proporcionado por MRTK para habilitarlo, pues por defecto no lo está.

Y una vez que es reconocido por el gestor de gestos, el evento “OnInputClicked”, capturado desde la interface “IInputClickHandler”, no era capaz de distinguir entre un “Simple Tap” y un “Double Tap”, por lo que siempre los acpturaba.

La solución a esto fue hallada en las redes [73], donde recomendaban utilizar un temporizador para poder identificar si se producía un “Double Tap” o no.

5.4.8. Buscando a UiRayCastCamera

No es hasta que se incorpora el prefab del teclado cuando se es consciente de que MRTK ha introducción una cámara raycast para poder interactuar con Canvas [74] y elementos UI.

Hasta ese momento , la solución había sido dispones BoxColiders [75] a los botones y paneles para poder interactuar con ellos como si se tratara de un objeto 3D.

De modo que lo único que se debe hacer para asociar dicha cámara a los elementos canvas que se tienen declarados por medio de un script, que realiza un Find y la asigna al componente “camera” del canvas.

```
UiCam = GameObject.Find("UiRaycastCamera");  
this.GetComponent<Canvas>().worldCamera = UiCam.GetComponent<Camera>();
```

Ilustración 5.32 - Código para encontrar cámara uiRaycast

6. Conclusiones

Este proyecto fue tomado desde el comienzo como un reto, un trabajo que pondría a prueba, no sólo los conocimientos que he adquirido en el Máster de Ingeniería Informática, sino también mi capacidad de investigación y autoaprendizaje en el área de nuevas tecnologías como la realidad mixta.

Me siento muy satisfecha y orgullosa del trabajado que he realizado en este proyecto, porque he sido capaz de utilizar una tecnología en pleno auge del sector informático y de la que apenas hay documentación, toda una aventura a ciegas.

La realidad mixta tiene grandes expectativas de futuro, por ello Microsoft sigue apostando por ella y adelantándose a sus posibles competidores, anunciando una nueva versión de Hololens, Hololens 2 [76], que se espera para el primer cuarto del próximo año 2019.

Esta nueva versión promete un mayor ángulo de visión, un marco más liviano y cómodo, mayor calidad de pantallas, un chip IA y un precio significativamente menor.

Para concluir con esta memoria, se mencionará una lista de las posibles mejoras a incluir, además de las historias de usuario que han quedado pendientes de desarrollar, en las próximas versiones del prototipo:

- Incluir en la aplicación un pequeño proceso de aprendizaje, un tutorial, para que el usuario aprenda a utilizar tanto los gestos como las opciones disponibles en el programa.
- Incluir compatibilidad con otros formatos de nubes de puntos, como por ejemplo OBJ, LAS, STL, etc
- Mejora de rendimiento de los procesos de selección, deselección y borrado de puntos.
- Investigar compatibilidad con otros dispositivos de realidad mixta, como los Windows Mixed Reality Headset de Microsoft.

- Añadir sistema de reunión, es decir, que diferentes usuarios puedan manipular y editar en tiempo real y a la vez una misma nube de puntos.
- Incluir en el punto anterior, que en dicho sistema se puedan conectar usuarios con dispositivos diferentes, pero compatibles con la aplicación, como otros headsets u ordenadores.

7. Anexos

7.1. Tarjetas de historias de usuario iniciales

1	Sistema de carga de PCL	
Cómo usuario quiero poder cargar una nube de puntos en la aplicación		
Estimación: 13		Tema: General
Prioridad: 20		
Pruebas de aceptación		
- Indicar nombre de nube existente en el directorio y comprobar los resultados. - Indicar nombres y extensiones inexistentes y comprobar los mensajes de error.		

Tabla 7-1 – Historia de usuario inicial ID: 1

2	Visualización de la PCL	
Como usuario quiero poder visualizar una nube de puntos escalada/superpuesta sobre el entorno en el que me encuentre.		
Estimación: 20		Tema: General
Prioridad: Must		
Pruebas de aceptación		
• Comprobar que la nube cargada se representa sin deformidades, es decir, los puntos están correctamente ubicados. • Comprobar que la nube está posicionada en una zona accesible y escalada al tamaño adecuado. • Comprobar que se representan correctamente los colores de los puntos		

Tabla 7-2 – Historia de usuario inicial ID: 2

3.1	Manipulación de PCL	
Como usuario quiero poder manipular (mover, rotar y escalar) la nube de puntos.		
Estimación: 20		Tema: Edición de PCL
Prioridad: Must		
Pruebas de aceptación		
Comprobar que la nube se desplaza correctamente durante la		

manipulación.

- Comprobar que la nube puede rotarse sobre los ejes X, Y, Z.
- Comprobar que la nube puede escalarse por igual en todos los ejes, es decir que un eje no se escale más o menos que el resto.

Tabla 7-3 – Historia de usuario inicial ID: 3.1

3.2	Selección de puntos por área	
Como usuario quiero poder seleccionar un grupo de puntos de la nube de puntos utilizando un objeto como selector, quedando todos los puntos incluidos en su área marcados por un color identificativo. También quiero que los puntos marcados puedan desmarcarse.		
Estimación: 40		Tema: Edición PCL
Prioridad: Must		
Pruebas de aceptación		
• Comprobar que se puede desplazar el objeto selector y ese sigue la dirección y sentido de la mano • Comprobar que, efectivamente, aquellos puntos que pasan por el objeto selector quedan marcados. • Confirmar que los puntos marcados pueden desmarcarse		

Tabla 7-4 - Historia de usuario inicial ID: 3.2

3.3	Manipulación de los puntos seleccionados	
Como usuario quiero poder mover, rotar o escalar un conjunto de puntos seleccionado.		
Estimación: 40		Tema: Edición PCL
Prioridad: Should		
Pruebas de aceptación		
• Aplicar transformaciones a un grupo de puntos seleccionado previamente		

Tabla 7-5 – Historia de usuario inicial ID: 3.3

3.4	Borrado de puntos de PCL
Como usuario quiero poder eliminar los puntos que haya seleccionado en la nube de puntos.	

Estimación: 13	Tema: Edición PCL
Prioridad: Must	
Pruebas de aceptación	
• Seleccionar un grupo de puntos y activar la función de borrado, verificando si realmente desaparecen de la nube	

Tabla 7-6 – Historia de usuario inicial ID: 3.4

3.5	Añadido de puntos a PCL
Como usuario quiero poder añadir puntos en zonas no los haya.	
Estimación: 40	Tema: Edición PCL
Prioridad: Would	
Pruebas de aceptación	
• Seleccionar o trazar un área sin puntos • Generar puntos dentro del área marcada y comprobar el resultado	

Tabla 7-7 – Historia de usuario inicial ID: 3.5

4	Guardado de la nube
Como usuario quiero poder salvar/guardar la nube de puntos editada, pudiendo indicar un nuevo nombre para la nube.	
Estimación: 13	Tema: General
Prioridad: Must	
Pruebas de aceptación	
• Subir nube habiendo indicado un nuevo nombre • Intentar subir la nube cuando del servidor no está activado, comprobar mensajes de error • Intentar subir una nube sin indicar un nombre nuevo	

Tabla 7-8 – Historia de usuario inicial ID: 4

5.1	Menú de selección de PCL
Como usuario quiero poder ver las nubes disponibles para poder escoger una para cargarla en el sistema.	
Estimación: 13	Tema: Interfaz de usuario
Prioridad: Must	

Pruebas de aceptación
• Revisar que se carga la lista completa de nubes • Verificar el funcionamiento de todos los botones con el gesto “Tap” • Seleccionar una nube y verificar que se carga la nube seleccionada y no otra

Tabla 7-9 – Historia de usuario inicial ID: 5.1

5.2	Menú inicio de aplicación	
Como usuario quiero conocer el estado del proceso de carga y ,una vez finalizado, poder decidir si cambiar de nube, cerrar la aplicación o visualizar la nube cargada.		
Estimación: 5		Tema: Interfaz de usuario
Prioridad: Must		
Pruebas de aceptación		
• Confirmar el funcionamiento de los botones con el gesto “Tap” • Validar durante el proceso de carga, los mensajes de información de estado mostrados		

Tabla 7-10 – Historia de usuario inicial ID: 5.2

5.3.1	Menú manipulación PCL
Como usuario quiero tener un menú, que pueda recolocar a mi gusto, donde tener agrupadas todas las herramientas para la manipulación de la nube de puntos.	
Estimación: 40	Tema: Interfaz de usuario
Prioridad: Must	
	Epic: Menú de Herramientas PCL
Pruebas de aceptación	
• Interactuar con el gesto “Tap” con los botones de cada herramienta • Con el gesto “Double Tap”, realizar traslaciones, rotaciones y escalas al menú	

Tabla 7-11 – Historia de usuario inicial ID: 5.3.1

5.3.2	Menú herramientas puntos de PCL
Como usuario quiero tener un menú, que pueda recolocar a mi gusto, donde	

tener agrupadas todas las herramientas relacionadas con los puntos de la nube de puntos.	
Estimación: 40	Tema: Interfaz de usuario Epic: Menú de Herramientas PCL
Prioridad: Must	
Pruebas de aceptación	
• Interactuar con el gesto “Tap” con los botones de cada herramienta • Con el gesto “Double Tap”, realizar traslaciones, rotaciones y escalas al menú	

Tabla 7-12 – Historia de usuario inicial ID: 5.3.2

5.4	Menú de guardado PCL	
Como usuario quiero tener un panel o formulario donde poder indicar el nombre de la nueva nube y dispone de un teclado virtual para poder hacerlo.		
Estimación: 8		Tema: Interfaz de usuario
Prioridad: Could		
Pruebas de aceptación		
• Verificar que hacer “Tap” sobre el campo de texto aparece en escena el teclado • Comprobar qué al escribir en el teclado, el texto se va añadiendo en el campo del formulario de subida • Verificar que los botones del menú realizan las acciones que tienen programadas		

Tabla 7-13 – Historia de usuario inicial ID: 5.2

7.2. Tarjetas de historias de usuario finales

1	Descarga de la PCL	
Cómo usuario quiero poder descargar una nube de puntos de un determinado servidor.		
Estimación: 13		Tema: General
Prioridad: Must		
Pruebas de aceptación		
• Comprobar que se listan todas las PCL disponibles en el directorio del servidor. • Comprobar que al seleccionar una PCL, se descargue esta y no otra cualquiera. • Descargar un fichero/modelo repetido y comprobar el mensaje de error.		

Tabla 7-14 – Historia de usuario final ID: 1

2	Visualización de la PCL	
Como usuario quiero poder visualizar una nube de puntos escalada/superpuesta sobre el entorno en el que me encuentre.		
Estimación: 20		Tema: General
Prioridad: Must		
Pruebas de aceptación		
• Comprobar que la nube cargada se representa sin deformidades, es decir, los puntos están correctamente ubicados. • Comprobar que la nube está posicionada en una zona accesible y escalada al tamaño adecuado. • Comprobar que se representan correctamente los colores de los puntos		

Tabla 7-15 - Historia de usuario final ID: 2

3	Menú de selección de PCL	
Como usuario quiero poder ver las nubes disponibles para poder escoger una para cargarla en el sistema.		
Estimación: 13		Tema: Interfaz de usuario
Prioridad: Must		
Pruebas de aceptación		

- Revisar que se carga la lista completa de nubes
- Verificar el funcionamiento de todos los botones con el gesto “Tap”
- Seleccionar una nube y verificar que se carga la nube seleccionada y no otra

Tabla 7-16 - Historia de usuario final ID: 3

4	Menú inicio de aplicación	
Como usuario quiero conocer el estado del proceso de carga y ,una vez finalizado, poder decidir si cambiar de nube, cerrar la aplicación o visualizar la nube cargada.		
Estimación: 5		Tema: Interfaz de usuario
Prioridad: Must		
Pruebas de aceptación		
• Confirmar el funcionamiento de los botones con el gesto “Tap” • Validar durante el proceso de carga, los mensajes de información de estado mostrados		

Tabla 7-17– Historia de usuario final ID: 4

5	Centrado de elementos en el entorno	
Como usuario quiero estar seguro de que todo lo que pueda manipular esté a mi alcance, centrado en la habitación donde me encuentre.		
Estimación: 20		Tema: General
Prioridad: Must		
Pruebas de aceptación		
- Realizar un escaneado del entorno - Confirmar que los elementos se colocan en el centro calculado		

Tabla 7-18 – Historia de usuario final ID: 5

6	Soporte para ficheros PLY en binario	
Como usuario quiero poder cargar nubes de puntos en formato binario, ya que son más livianas y su descarga es más rápida		
Estimación: 5		Tema: General
Prioridad: Must		

Pruebas de aceptación
• Seleccionar una nube PLY en formato binario y comprobar que carga correctamente

Tabla 7-19– Historia de usuario final ID: 6

7	Menú herramientas de sistema
Como usuario quiero tener un menú, que pueda recolocar a mi gusto, donde tener agrupadas todas las herramientas del sistema, como cerrar la aplicación, cambiar de nube, etc..	
Estimación: 20	Tema: Interfaz de usuario
Prioridad: Must	Epic: Menú de Herramientas PCL
Pruebas de aceptación	
• Interactuar con el gesto “Tap” con los botones de cada herramienta • Con el gesto “Double Tap”, realizar traslaciones, rotaciones y escalas al menú	

Tabla 7-20 – Historia de usuario final ID: 7

8	Menú manipulación PCL
Como usuario quiero tener un menú, que pueda recolocar a mi gusto, donde tener agrupadas todas las herramientas para la manipulación de la nube de puntos.	
Estimación: 40	Tema: Interfaz de usuario
Prioridad: Must	Epic: Menú de Herramientas PCL
Pruebas de aceptación	
• Interactuar con el gesto “Tap” con los botones de cada herramienta • Con el gesto “Double Tap”, realizar traslaciones, rotaciones y escalas al menú	

Tabla 7-21 – Historia de usuario final ID: 8

9	Selección de puntos por área	
Como usuario quiero poder seleccionar un grupo de puntos de la nube de puntos utilizando un objeto como selector, quedando todos los puntos incluidos en su área marcados por un color identificativo. También quiero que los puntos marcados puedan demarcarse.		
Estimación: 40		Tema: Edición PCL
Prioridad: Must		
Pruebas de aceptación		
• Comprobar que se puede desplazar el objeto selector • Comprobar que, efectivamente, aquellos puntos que pasan por el objeto selector quedan marcados. • Confirmar que los puntos marcados pueden desmarcarse		

Tabla 7-22– Historia de usuario final ID: 9

10	Borrado de puntos de PCL	
Como usuario quiero poder eliminar los puntos que haya seleccionado en la nube de puntos.		
Estimación: 13		Tema: Edición PCL
Prioridad: Must		
Pruebas de aceptación		
• Seleccionar un grupo de puntos y activar la función de borrado, verificando si realmente desaparecen de la nube		

Tabla 7-23– Historia de usuario final ID: 10

11	Menú herramientas puntos de PCL	
Como usuario quiero tener un menú, que pueda recolocar a mi gusto, donde tener agrupadas todas las herramientas relacionadas con los puntos de la nube de puntos.		
Estimación: 40		Tema: Interfaz de usuario
Prioridad: Must		
		Epic: Menú de Herramientas PCL
Pruebas de aceptación		
• Interactuar con el gesto “Tap” con los botones de cada herramienta • Con el gesto “Double Tap”, realizar traslaciones, rotaciones y escalas al menú		

Tabla 7-24– Historia de usuario final ID: 11

12	Guardado de la nube
Como usuario quiero poder salvar/guardar la nube de puntos editada, pudiendo indicar un nuevo nombre para la nube.	
Estimación: 13	Tema: General
Prioridad: Must	
Pruebas de aceptación	
• Subir nube habiendo indicado un nuevo nombre • Intentar subir la nube cuando del servidor no está activado, comprobar mensajes de error • Intentar subir una nube sin indicar un nombre nuevo	

Tabla 7-25– Historia de usuario final ID: 12

13	Menú de guardado PCL
Como usuario quiero tener un panel o formulario donde poder indicar el nombre de la nueva nube y dispone de un teclado virtual para poder hacerlo.	
Estimación: 8	Tema: Interfaz de usuario
Prioridad: Could	
Pruebas de aceptación	
• Verificar que hacer “Tap” sobre el campo de texto aparece en escena el teclado • Comprobar qué al escribir en el teclado, el texto se va añadiendo en el campo del formulario de subida • Verificar que los botones del menú realizan las acciones que tienen programadas	

Tabla 7-26– Historia de usuario final ID: 13

14	Sonidos en la aplicación
Como usuario que la aplicación tenga una melodía de fondo, sonidos de aviso y sonidos en los botones de los menús.	
Estimación: 5	Tema: Interfaz de usuario
Prioridad: Must	
Pruebas de aceptación	
• Seleccionar o trazar un área sin puntos	

- Generar puntos dentro del área marcada y comprobar el resultado

Tabla 7-27– Historia de usuario final ID: 14

15	Añadido de puntos a PCL	
Como usuario quiero poder añadir puntos en zonas no los haya.		
Estimación: 40		Tema: Edición PCL
Prioridad: Would		
Pruebas de aceptación		
• Seleccionar o trazar un área sin puntos • Generar puntos dentro del área marcada y comprobar el resultado		

Tabla 7-28– Historia de usuario final ID: 15

16	Manipulación de los puntos seleccionados	
Como usuario quiero poder mover, rotar o escalar un conjunto de puntos seleccionado.		
Estimación: 40		Tema: Edición PCL
Prioridad: Should		
Pruebas de aceptación		
• Aplicar transformaciones a un grupo de puntos seleccionado previamente		

Tabla 7-29– Historia de usuario final ID: 16

17	Alterar el tamaño de los puntos de la PCL	
Como usuario quiero poder modificar el tamaño de los puntos, para hacerlos más o menos visibles.		
Estimación: 20		Tema: Edición PCL
Prioridad: Could		
Pruebas de aceptación		
• Aplicar subida de tamaño a los puntos • Aplicar reducción de tamaño a los puntos		

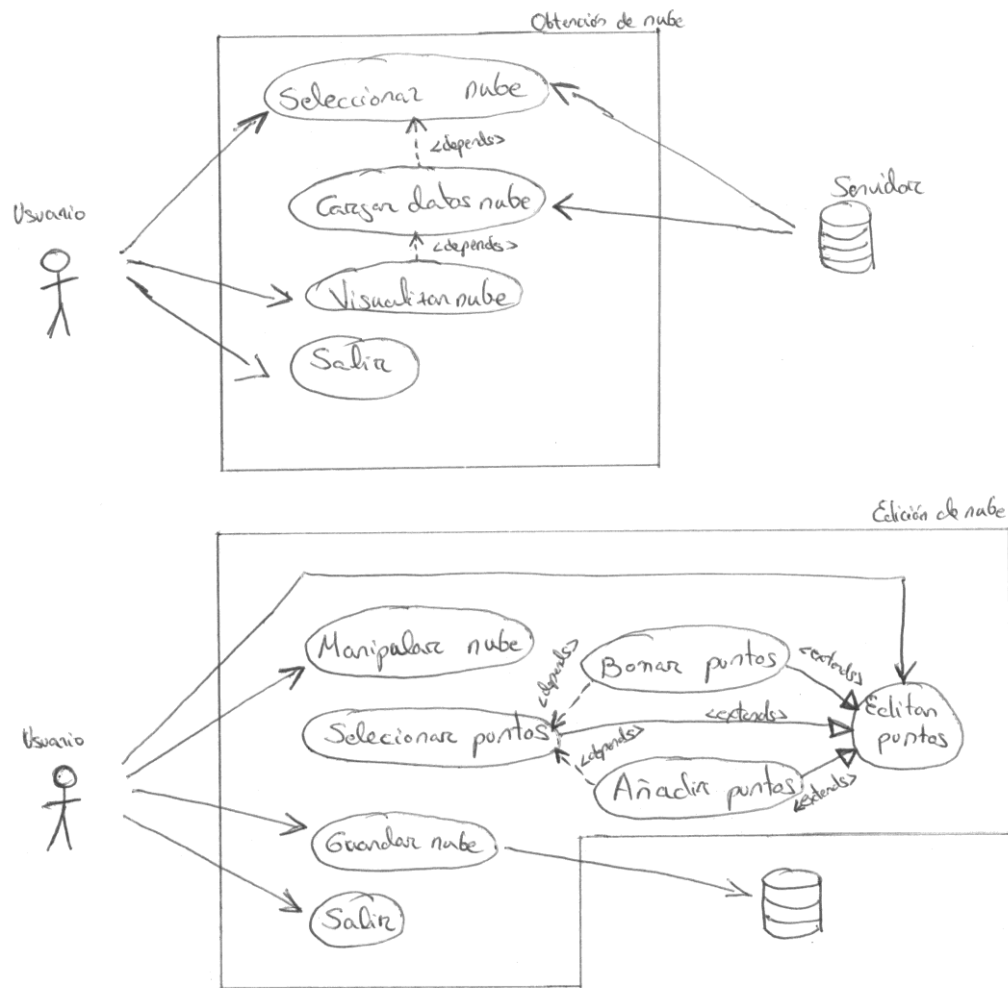
Tabla 7-30– Historia de usuario final ID: 17

18	Deshacer borrado de los puntos	
Como usuario quiero poder deshacer el borrado de puntos.		
Estimación: 20		Tema: Edición PCL
Prioridad: Would		
Pruebas de aceptación		
Borrar un grupo de puntos y deshacer el borrado		

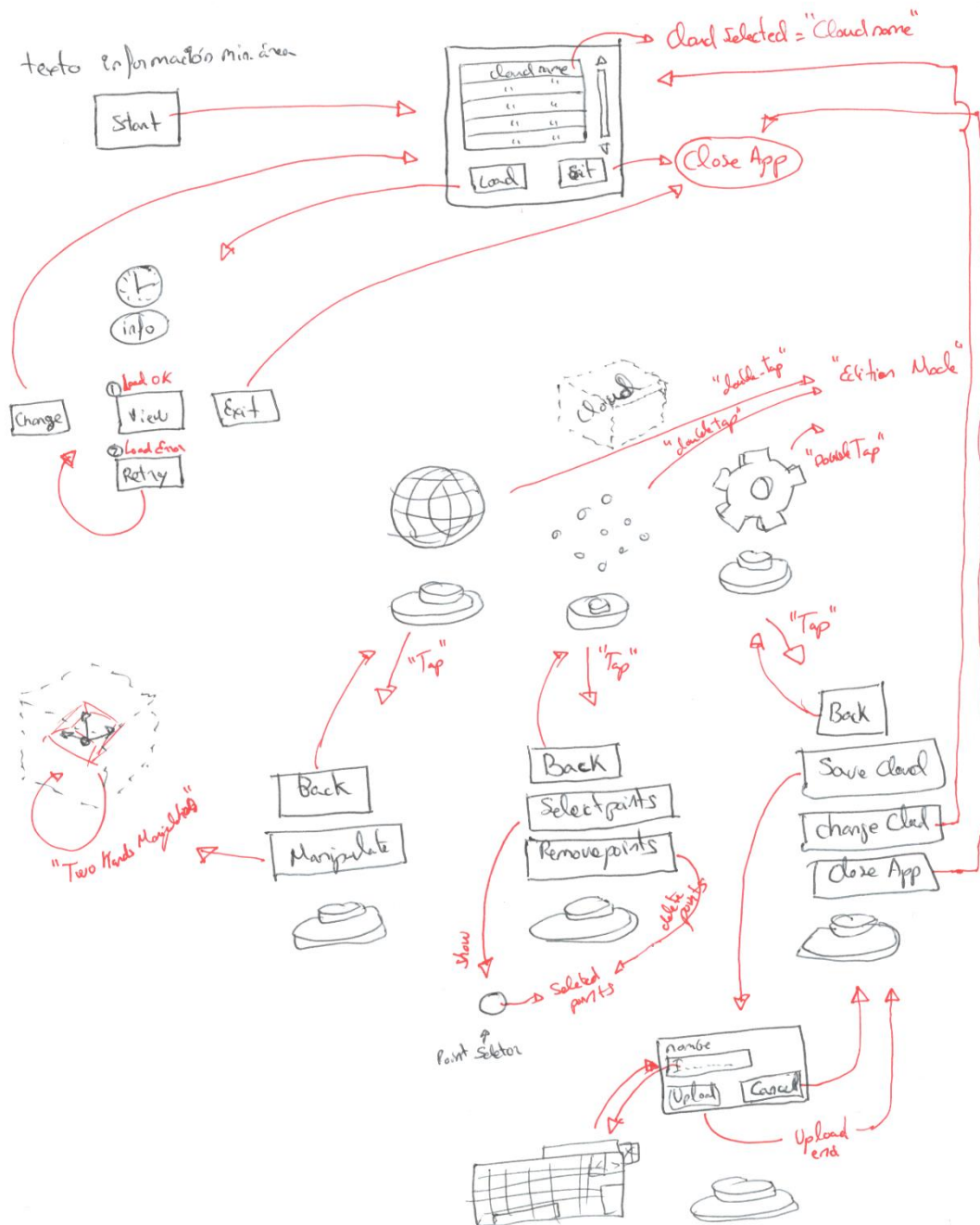
Tabla 7-31– Historia de usuario final ID: 18

7.3. Diagramas y grafos

7.3.1. Diagramas de caso de uso



7.3.2. Storyboard de toda la interfaz de usuario de la aplicación



7.4. Manual de usuario

En este punto se va a describir un breve manual de usuario, en el que se explicará cómo utilizar las funcionalidades de la aplicación.

Lo primero que debe conocer el usuario es el gesto básico "Tap", Ilustración 7.1, que consiste en tener un dedo erguido y seguidamente, sin doblarlo, bajarlo hasta que quede en línea con el nudillo.

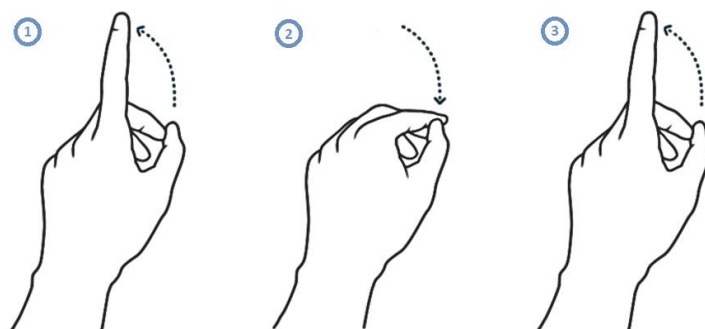


Ilustración 7.1 - Ejemplo de gesto básico "Tap"

7.4.1. Escaneado de entorno

Al iniciar la aplicación, se requerirá realizar un escaneado del entorno, para que el sistema pueda identificar la habitación, teniendo el usuario que moverse un poco por la habitación.

Durante el proceso, se podrá apreciar las zonas identificadas por el sistema, pues quedan cubiertas por una malla de triángulos.

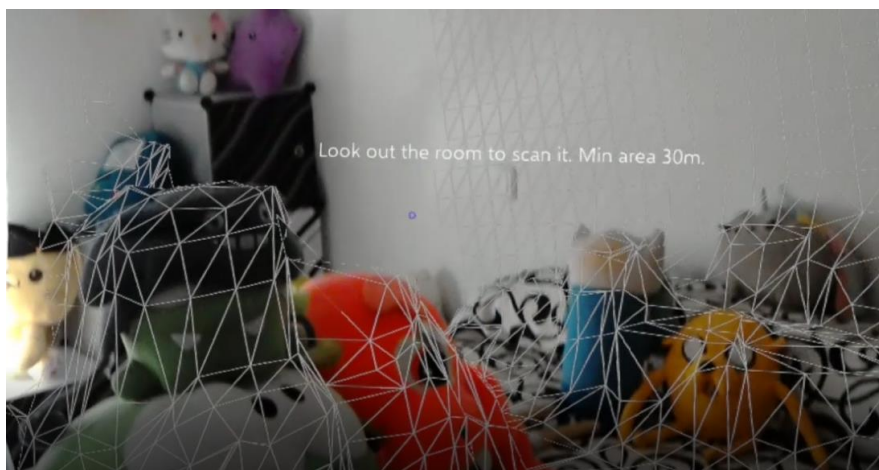


Ilustración 7.2 - Ejemplo de escaneado del entorno

Cuando se haya alcanzado el mínimo de superficie escaneada, el mensaje informativo cambiará y se activará el botón de inicio.

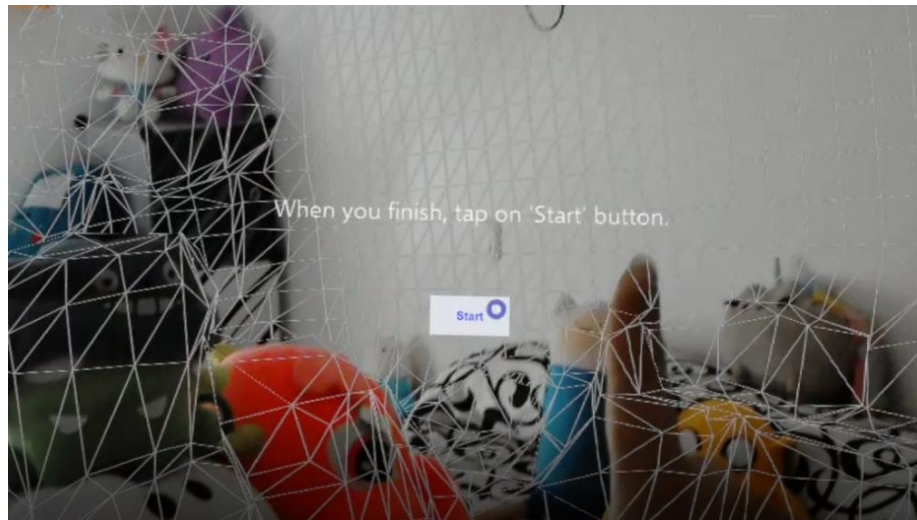


Ilustración 7.3 - Ejemplo de mínimo de superficie escaneada alcanzado

7.4.2. Elección de una nube de puntos

Una vez se pulse, con el gesto “Tap”, el botón de inicio, se mostrará al usuario el panel de selección de nubes ubicado en el centro de la zona escaneada, Ilustración 7.4.

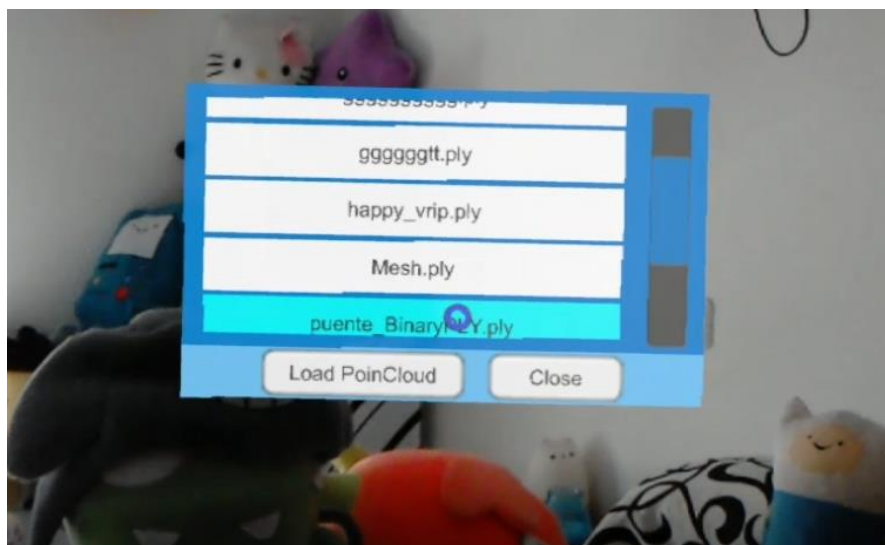


Ilustración 7.4 - Ejemplo de menú de selección de nubes

Moviendo la cabeza, se moverá también el puntero haciendo que los botones cambien de color al pasar sobre ellos.

Para desplazar la barra lateral, hay que hacer el gesto “Hold”, que es similar al “Tap” pero manteniendo el dedo bajado, Ilustración 7.5, y a la vez desplazar la mano hacia arriba o abajo, siempre sin perder el foco del puntero.

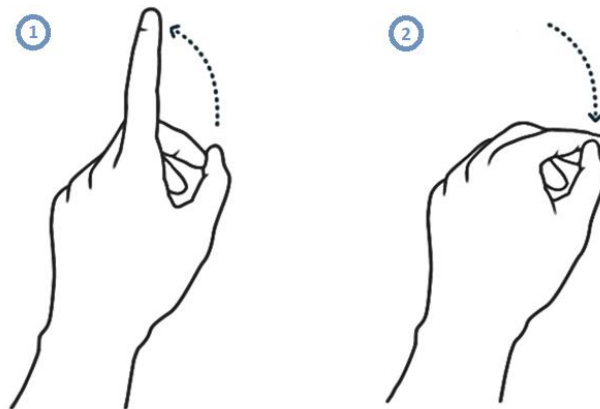


Ilustración 7.5 - Ejemplo de gesto básico "Hold"

Una vez ya se tenga decidido que nube se va a seleccionar, se posiciona el puntero sobre su botón y se pulsa con “Tap”, quedando el botón marcado con un color azul oscuro, Ilustración 7.6.

Seguidamente, se pulsará el botón “Load PointCloud” para iniciar la descarga o el botón “Close” para salir de la aplicación.

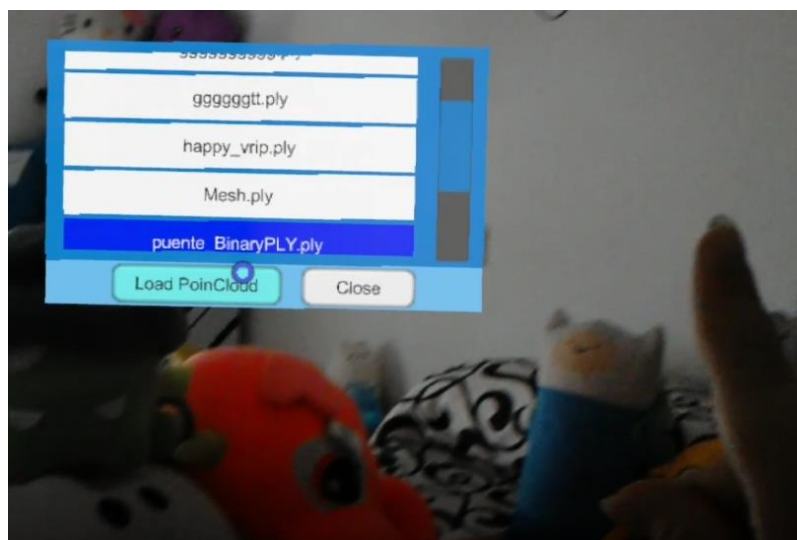


Ilustración 7.6 - Ejemplo selección de una nube

Durante la descarga y lectura de la nube se podrá ver un reloj de arena, girando sobre sí mismo, acompañado de un mensaje de texto que indicará el estado de del proceso interno.

Cuando dicho proceso finalice, y si no se ha producido ningún error, se mostrará el menú principal, Ilustración 7.7, compuesto por tres botones. Estos botones permiten al usuario, volver al menú anterior para seleccionar otra nube, salir de la aplicación o iniciar la visualización de la nube.

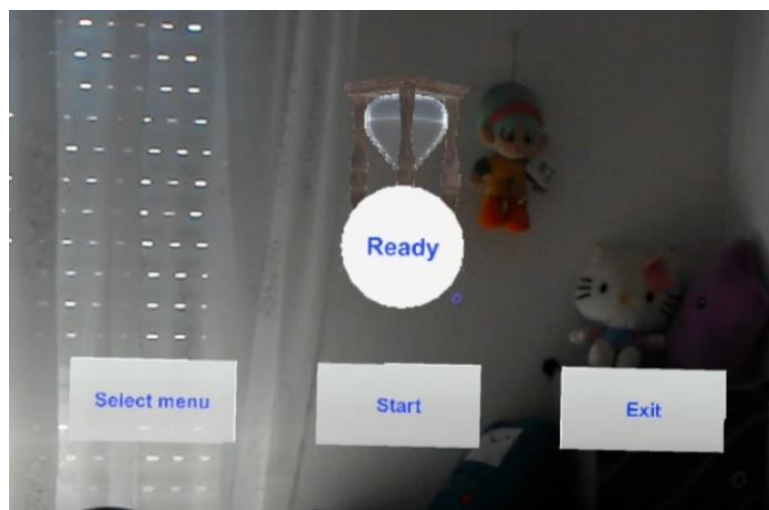


Ilustración 7.7 - Ejemplo de menú principal

Si pulsamos “Start”, se inicializará la visualización de la zona de trabajo donde podrá ver a la nube seleccionada y el menú de herramientas, Ilustración 7.8.



Ilustración 7.8 - Ejemplo de nube cargada y menú de herramientas

7.4.3. Área de trabajo y Herramientas

El menú de herramientas, por defecto, está ubicado junto a la nube en zona de trabajo, pero esto se puede cambiar.

Cada uno de los menús puede ser desplazado, rotado y escalado, para ello solo se tiene que realizar el gesto “Double-Tap”, que es igual que el gesto “Tap”, pero se realiza dos veces seguidas.

Esto habilita la edición del icono 3D del menú, generando una caja envolvente con pivotes, Ilustración 7.9. En ese momento el usuario puede trasladarlo haciendo “Hold” en su centro, o rotarlo y/o escalarlo aplicando “Hold” sobre los pivotes de la caja.

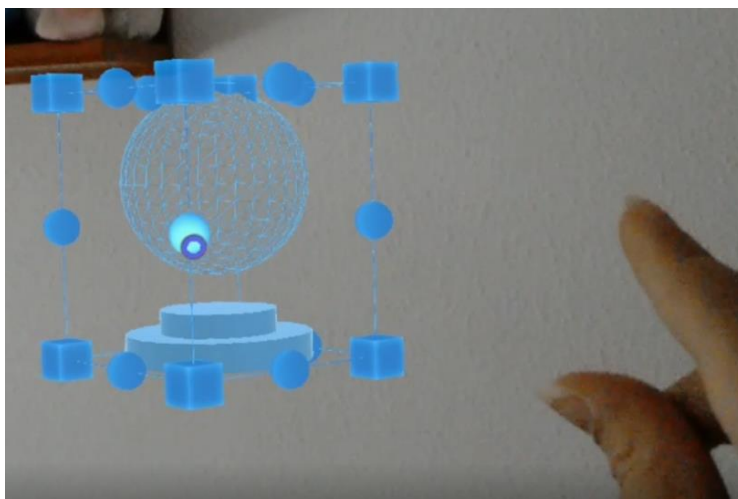


Ilustración 7.9 - Ejemplo de edición de icono 3D del menú de Herramientas

Si lo que se quiere es acceder a las funcionalidades de cada menú, bastará con hacer un “Tap”, y se desplegará el conjunto de botones disponible.

A continuación, se explicará cómo funciona cada menú del menú de herramientas.

7.4.3.1. Herramientas de sistema

En este menú se dispone de aquellas funciones propias del sistema, como cerrar la aplicación, cambiar de nube y guardar la nube, todas accesibles desde botones 3D, incluida la opción de volver al menú de herramientas.

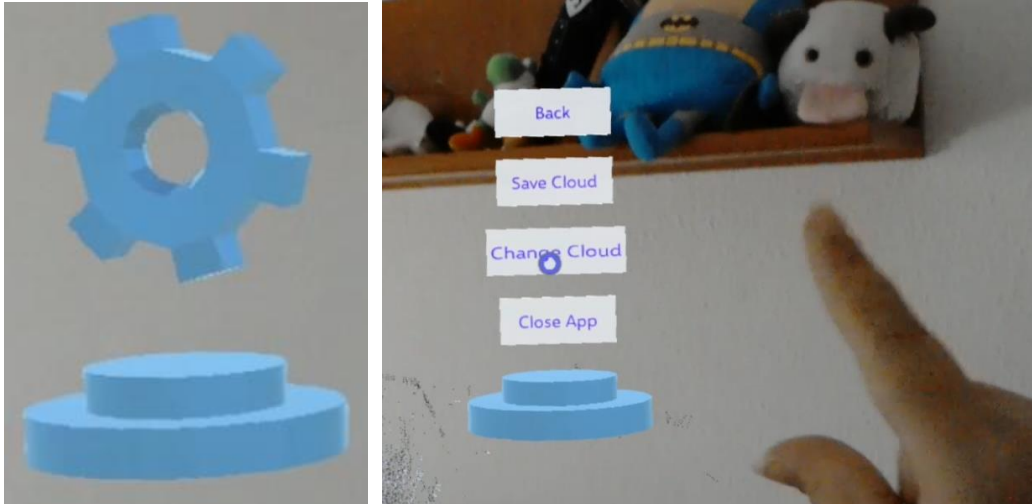


Ilustración 7.10 - Ejemplo de menú de herramientas de sistema y botones desplegados

La función “Save Cloud” despliega un panel donde se puede escribir el nombre de la nube a través de un teclado 3D, que surge cuando se hace “Tap” en el cuadro de texto.

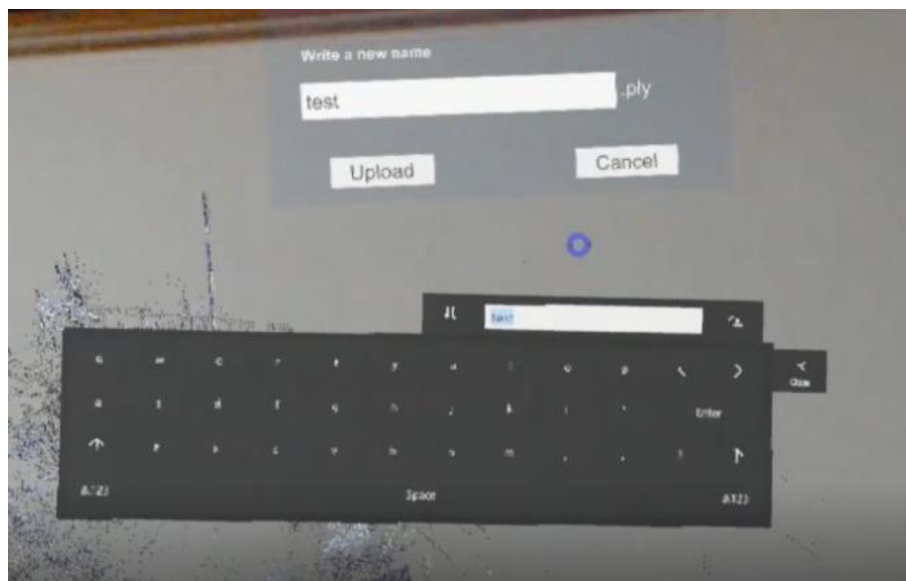


Ilustración 7.11 - Ejemplo Panel guardar nube y teclado 3D

Cuando se tenga listo la nube e indicado el nuevo nombre, sólo habrá que pulsar en upload para que se genere un nuevo fichero PLY y se suba al servidor.

7.4.3.2. Herramientas de nube

En este menú se dispone de aquellas funciones referentes a la nube, como la manipulación de la nube, todas accesibles desde botones 3D, incluida la opción de volver al menú de herramientas.

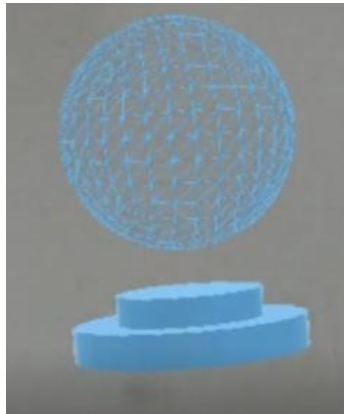


Ilustración 7.12 - Ejemplo de icono de Herramientas de nube

Si se hace “Tap” sobre el botón “Manipulate”, se mostrará un pivote en el centro de la nube, que es manipulable haciendo uso de las dos manos, Ilustración 7.13.

Cuando el pivote es trasladado, rotado y escalado, hace que estos cambios se apliquen a la nube de puntos. Y para finalizar la manipulación, se hará “Tap” sobre el botón “Done”.



Ilustración 7.13 - Ejemplo de manipulación de nube con dos manos

7.4.3.3. Herramientas de puntos

En este menú se dispone de aquellas funciones relacionadas con los puntos de la nube, como seleccionar puntos y borrar puntos, todas accesibles desde botones 3D, incluida la opción de volver al menú de herramientas.

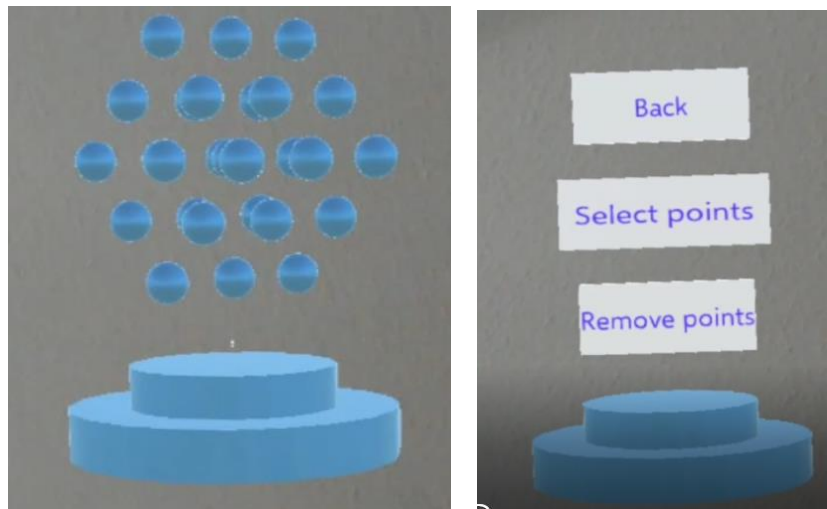


Ilustración 7.14 - Ejemplo de menú de herramientas de puntos y botones desplegados

Si se va a realizar una selección de puntos, se hará “Tap” en el botón “Select points” dejando visible la esfera selectora y cambiando el nombre al botón por “Unselect points”.

Para iniciar la selección de puntos, se debe enfocar a la esfera selectora y realizar el gesto “Hold”, haciendo que la esfera cambie de color, Ilustración 7.15.

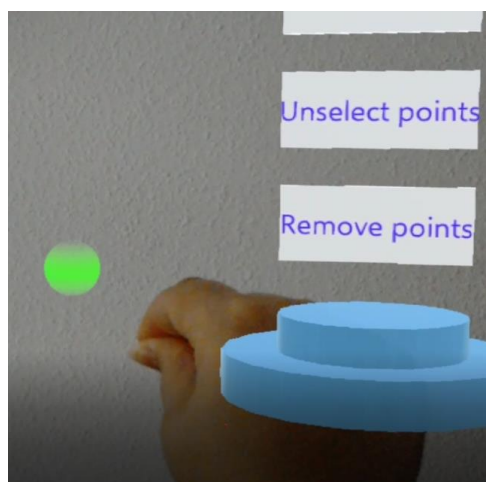


Ilustración 7.15 - Ejemplo de inicio de la selección de puntos

Manteniendo el gesto, se realizará un desplazamiento de la mano hacia los puntos que se deseen marcar, observando como la esfera va siguiendo la dirección de la mano y cambiando el color de los puntos que pasan por ella.

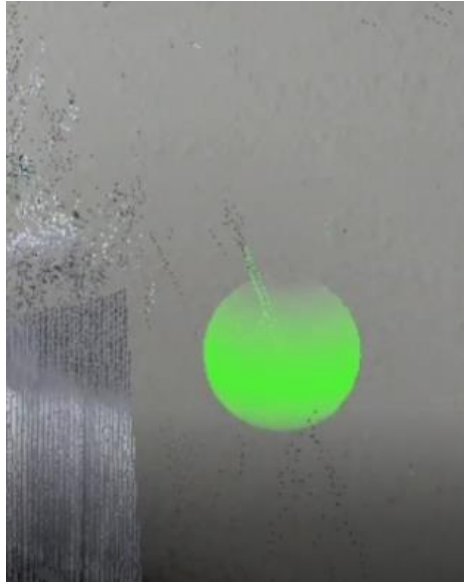


Ilustración 7.16 - Ejemplo de puntos marcados

Una vez se tengan todos los puntos deseados seleccionados, se podrá bien cancelar la selección haciendo “Tap” en “Unselect points”, o bien borrar dichos puntos haciendo “Tap” en “Remove points”.

Bibliografía

- [1] DomoDesk, «DomoDesk,» [En línea]. Available: <http://www.domodesk.com/221-a-fondo-que-es-iot-el-internet-de-las-cosas.html>. [Último acceso: 13 Marzo 2018].
- [2] Niantic, «Pokemon GO,» [En línea]. Available: <https://pokemongolive.com/es/>. [Último acceso: 1 Febrero 2018].
- [3] VRWERX, «Paranormal Activity: The Lost Soul,» [En línea]. Available: <https://www.pavrgame.com/>. [Último acceso: 1 Febrero 2018].
- [4] i. y. c. Ministerio de economía, «Industria conectada 4.0,» [En línea]. Available: <http://www.industriaconectada40.gob.es/Paginas/index.aspx#industria-4>. [Último acceso: 09 Enero 2018].
- [5] T. Palladino, «Next Reality - Renault Trucks & Hololens,» [En línea]. Available: <https://hololens.reality.news/news/renault-trucks-tests-hololens-visualize-quality-control-engine-assembly-operations-0181282/>. [Último acceso: 09 Enero 2018].
- [6] Autodesk, «Trabajo con nubes de puntos,» [En línea]. Available: <https://knowledge.autodesk.com/es/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2017/ESP/AutoCAD-Core/files/GUID-C0C610D0-9784-4E87-A857-F17F1F7FEEBE-htm.html>. [Último acceso: 13 Marzo 2018].
- [7] K. Joyce, «AR, VR, MR, RR, XR: A Glossary to the Acronyms of the Future,» vrFocus, [En línea]. Available: <https://www.vrfocus.com/2017/05/ar-vr-mr-rr-xr-a-glossary-to-the-acronyms-of-the-future/>. [Último acceso: 13 Marzo 2018].
- [8] Microsoft, «Microsoft Hololens,» [En línea]. Available: <https://www.microsoft.com/es-es/hololens>. [Último acceso: 22 Noviembre 2017].
- [9] Adobe, «Tienda Adobe,» [En línea]. Available: https://www.adobe.com/es/creativecloud/plans.html?plan=edu%3Fstore_code%3Des%3Fsdid%3DJRSIX&mv=search&s_kwid=AL!3085!3!226910053363!e!!g!!photoshop&ef_id=WnRlpgAAAizhSFQm%3A20180317184807%3As. [Último acceso: 17 Marzo 2018].
- [10] Autodesk, «3DS MAX Education,» [En línea]. Available: <https://www.autodesk.com/education/free-software/3ds-max>. [Último acceso: 17 Marzo 2018].
- [11] Microsoft, «Tienda Hololens,» [En línea]. Available: <https://www.microsoft.com/es-es/hololens/buy>. [Último acceso: 17 Marzo 2018].
- [12] Microsoft, «Hardware Hololens,» [En línea]. Available: <https://developer.microsoft.com/es-es/windows/mixed->

- reality/hololens_hardware_details. [Último acceso: 17 Marzo 2018].
- [13] M. Hachman, «pcworld,» 2 Marzo 2016. [En línea]. Available: <https://www.pcworld.com/article/3039822/consumer-electronics/we-found-7-critical-hololens-details-that-microsoft-hid-inside-its-developer-docs.html#>. [Último acceso: 16 Abril 2018].
- [14] Vistronica, «Vistronica,» [En línea]. Available: <https://www.vistronica.com/imu/>. [Último acceso: 16 Abril 2018].
- [15] T. Warren, «Inside Microsoft's Hololens,» theverge.com, 6 Abril 2016. [En línea]. Available: <https://www.theverge.com/2016/4/6/11376442/microsoft-hololens-holograms-parts-teardown-photos-hands-on>. [Último acceso: 16 Abril 2018].
- [16] Microsoft, «Install the tools,» [En línea]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools>. [Último acceso: 05 Junio 2018].
- [17] Microsoft, «Windows 10,» [En línea]. Available: <https://www.microsoft.com/en-us/software-download/windows10>. [Último acceso: 05 Junio 2018].
- [18] Microsoft, «Introducción a Hyper-V en Windows 10,» [En línea]. Available: <https://docs.microsoft.com/es-es/virtualization/hyper-v-on-windows/about/>. [Último acceso: 05 Junio 2018].
- [19] Microsoft, «Set up and configure RemoteFX vGPU for Remote Desktop Services,» [En línea]. Available: <https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-remotefx-vgpu>. [Último acceso: 05 Junio 2018].
- [20] Microsoft, «Visual Studio Community 2017,» [En línea]. Available: <https://docs.microsoft.com/es-mx/visualstudio/releases/vs2017-relnotes>. [Último acceso: 05 Junio 2018].
- [21] Microsoft, «Windows 10 SDK,» [En línea]. Available: <https://developer.microsoft.com/en-US/windows/downloads/windows-10-sdk>. [Último acceso: 05 Junio 2018].
- [22] Vuforia, «Vuforia,» [En línea]. Available: <https://www.vuforia.com/>. [Último acceso: 18 Junio 2018].
- [23] Innovae, «Realidad Aumentada,» [En línea]. Available: <http://realidadaumentada.info/tecnologia/>. [Último acceso: 18 Junio 2018].
- [24] Unity, «Unity 3D,» [En línea]. Available: <https://unity3d.com/es>. [Último acceso: 07 Junio 2018].
- [25] Editeca, «Realidad Mixta,» [En línea]. Available: <https://editeca.com/realidad-mixta/>.

[Último acceso: 18 Junio 2018].

- [26] Microsoft, «MixedRealityToolkit-Unity,» [En línea]. Available: <https://github.com/Microsoft/MixedRealityToolkit-Unity>. [Último acceso: 07 Junio 2018].
- [27] Unity, «Unity Documentation: Prefabs,» [En línea]. Available: <https://docs.unity3d.com/Manual/Prefabs.html>. [Último acceso: 25 Junio 2018].
- [28] P. ágiles, «SCRUM,» [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 25 Junio 2018].
- [29] scrummanager, «Historias de usuario - Scrum,» [En línea]. Available: http://www.scrummanager.net/bok/index.php/Historia_de_usuario. [Último acceso: 25 Junio 2018].
- [30] a. Garzas, «Puntos de Historia de Usuario,» [En línea]. Available: <http://www.javiergarzas.com/2015/06/puntos-historia-para-estimar-y-no-horas.html>. [Último acceso: 25 Junio 2018].
- [31] J. Garzas, «Escala de puntos de historia,» [En línea]. Available: <http://www.javiergarzas.com/2014/11/escala-de-puntos-historia.html>. [Último acceso: 25 Junio 2018].
- [32] A. Business, «MoSCoW,» [En línea]. Available: <https://www.agilebusiness.org/content/moscow-prioritisation-0>. [Último acceso: 25 Junio 2018].
- [33] J. Garzas, «Velocidad,» [En línea]. Available: <http://www.javiergarzas.com/2014/03/la-velocidad-en-un-proyecto-agil-o-scrum.html>. [Último acceso: 25 Junio 2018].
- [34] Microsoft, «Tienda Microsoft Office 365,» [En línea]. Available: <https://products.office.com/es-ES/compare-all-microsoft-office-products?tab=2>. [Último acceso: 26 Junio 2018].
- [35] Autodesk, «Tienda Autodesk - Suscripción 3DS Max,» [En línea]. Available: <https://www.autodesk.es/products/3ds-max/subscribe?referrer=%2Fproducts%2F3ds-max%2Fsubscribe&plc=3DSMAX&term=1-YEAR&support=ADVANCED&quantity=1>. [Último acceso: 26 Junio 2018].
- [36] Adobe, «Tienda Adobe,» [En línea]. Available: <https://www.adobe.com/es/creativecloud/plans.html?promoid=695P7SPW&mv=other>. [Último acceso: 27 Junio 2018].
- [37] Microsoft, «Tienda Hololens,» [En línea]. Available: <https://www.microsoft.com/es-es/p/microsoft-hololens-development->

- edition/8xf18pqz17ts?activetab=pivot%3aoverviewtab. [Último acceso: 26 Junio 2018].
- [38] InfoJobs, «InfoJobs,» [En línea]. Available: <https://www.infojobs.net/>. [Último acceso: 26 Junio 2018].
- [39] SimplyHired, «SimplyHired,» [En línea]. Available: <https://www.simplyhired.es/>. [Último acceso: 26 Junio 2018].
- [40] G. Llorach, «Asset Store - Point Cloud Free Viewer,» [En línea]. Available: <https://assetstore.unity.com/packages/tools/utilities/point-cloud-free-viewer-19811>. [Último acceso: 30 Junio 2018].
- [41] Unity, «Asset Store,» [En línea]. Available: <https://www.assetstore.unity3d.com/>. [Último acceso: 30 Junio 2018].
- [42] Unity, «Vector3,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/Vector3.html>. [Último acceso: 30 Junio 2018].
- [43] Unity, «Color,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/Color.html>. [Último acceso: 30 Junio 2018].
- [44] Unity, «Mesh,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/Mesh.html>. [Último acceso: 30 Junio 2018].
- [45] Unity, «Mesh limit vertices,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/Mesh-indexFormat.html>. [Último acceso: 30 Junio 2018].
- [46] Unity, «GameObject,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/GameObject.html>. [Último acceso: 30 Junio 2018].
- [47] Unity, «GameObject.Find(),» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/GameObject.Find.html>. [Último acceso: 30 Junio 2018].
- [48] U. d. Stanford, «The Stanford 3D Scanning Repository,» [En línea]. Available: <http://graphics.stanford.edu/data/3Dscanrep/>. [Último acceso: 30 Junio 2018].
- [49] «ASCII,» [En línea]. Available: <https://ascii.cl/es/>. [Último acceso: 30 Junio 2018].
- [50] Jose, «Arume: Formatos Little endian y Big endian,» [En línea]. Available: <http://www.arumeinformatica.es/blog/los-formatos-big-endian-y-little-endian/>. [Último acceso: 30 Junio 2018].
- [51] Sistemas, «Definición de Frame,» [En línea]. Available: <https://sistemas.com/frame.php>.

[Último acceso: 30 Junio 2018].

- [52] Unity, «Materiales, Shaders & Texturas,» [En línea]. Available: <https://docs.unity3d.com/es/current/Manual/Shaders.html>. [Último acceso: 30 Junio 2018].
- [53] Unity, «UnityWebRequest,» [En línea]. Available: <https://docs.unity3d.com/Manual/UnityWebRequest.html>. [Último acceso: 30 Junio 2018].
- [54] «MemoryStream,» [En línea]. Available: [https://msdn.microsoft.com/es-es/library/system.io.memorystream\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.io.memorystream(v=vs.110).aspx). [Último acceso: 30 Junio 2018].
- [55] Unity, «Coroutine,» [En línea]. Available: <https://docs.unity3d.com/Manual/Coroutines.html>. [Último acceso: 30 Junio 2018].
- [56] C. Vetter, «HoloLens Tutorial – Spatial Understanding,» [En línea]. Available: <https://www.cameronvetter.com/2016/12/21/hololens-tutorial-spatial-understanding/>. [Último acceso: 30 Julio 2018].
- [57] Unity, «Colider,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/Collider.html>. [Último acceso: 30 Junio 2018].
- [58] c00pala, «Unity UI - Scroll Menu Pt 2: Vertical List - A scrolling, vertical list of buttons,» [En línea]. Available: <https://www.youtube.com/watch?v=Zl6DwJtjIBA>. [Último acceso: 20 Febrero 2018].
- [59] «TurboSquid - Gear Model,» [En línea]. Available: <https://www.turbosquid.com/FullPreview/Index.cfm/ID/1261714>. [Último acceso: 2 Julio 2018].
- [60] sakai, «Hololens Keyboard,» [En línea]. Available: <https://blog.nextscape.net/archives/Date/2017/12/hololens-keyboard-01>. [Último acceso: 1 Julio 2018].
- [61] Niton, «UnityOctree,» [En línea]. Available: <https://github.com/Niton/UnityOctree>. [Último acceso: 2 Julio 2018].
- [62] Unity, «Sending a form to an HTTP server (POST) - Using WWWForm,» [En línea]. Available: <https://docs.unity3d.com/Manual/UnityWebRequest-SendingForm.html>. [Último acceso: 2 Julio 2018].
- [63] «Youtube,» [En línea]. Available: <https://www.youtube.com/>. [Último acceso: 3 Julio 2018].
- [64] «Freesound,» [En línea]. Available: <https://freesound.org/>. [Último acceso: 2 Julio 2018].

- [65] Unity, «Audio Source,» [En línea]. Available: <https://docs.unity3d.com/ScriptReference/AudioSource.html>. [Último acceso: 2 Julio 2018].
- [66] Microsoft, «HoloToolkit,» [En línea]. Available: <https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/v1.2017.2.0>. [Último acceso: 24 Enero 2018].
- [67] Microsoft, «Windows 10 update,» [En línea]. Available: <https://support.microsoft.com/es-es/help/4028685/windows-10-get-the-update>. [Último acceso: 30 Junio 2018].
- [68] «MonoDevelop,» [En línea]. Available: <https://www.monodevelop.com/>. [Último acceso: 30 Junio 2018].
- [69] L. Klint, «SitePoint,» [En línea]. Available: <https://www.sitepoint.com/building-your-first-hololens-app-creating-assets/>. [Último acceso: 26 Enero 2018].
- [70] Microsoft, «Windows Mixed Reality headsets,» [En línea]. Available: <https://www.microsoft.com/en-us/store/collections/vrandmixedrealityheadsets>. [Último acceso: 30 Junio 2018].
- [71] Microsoft, «MixedRealityToolkit for Unity 2017.2.1.4 version,» [En línea]. Available: <https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/2017.2.1.4>. [Último acceso: 25 Abril 2018].
- [72] Untiy, «Optimización Shader para Hololens,» [En línea]. Available: <https://docs.unity3d.com/2017.2/Documentation/Manual/SinglePassStereoRenderingHoloLens.html>.
- [73] GitHub, «Double Tap recognition,» [En línea]. Available: <https://github.com/Microsoft/MixedRealityToolkit-Unity/issues/654>. [Último acceso: 30 Mayo 2018].
- [74] Unity, «Canvas,» [En línea]. Available: <https://docs.unity3d.com/es/current/Manual/UICanvas.html>. [Último acceso: 3 Julio 2018].
- [75] Unity, «BoxColider,» [En línea]. Available: <https://docs.unity3d.com/Manual/class-BoxCollider.html>. [Último acceso: 3 Julio 2018].
- [76] D. Hardawar, «Hololens 2 para 2019,» [En línea]. Available: <https://www.engadget.com/2018/06/13/microsoft-hololens-2-surface/?guccounter=1>. [Último acceso: 3 Julio 2018].