Master en Desarrollo de Aplicaciones

sobre Dispositivos Móviles

(Ed. 2014-2015)

Proyecto Final

Aplicación Interactiva - PhotoBubblesVR

Nombre: Victor Roig Martinez

Introducción

Descripción del problema

El proyecto PhotoBubblesVR persigue la ***creación de una aplicación interactiva basada en Realidad Virtual (VR)*** y en el uso de la cámara y sensor de profundidad Intel RealSense o similar.

El objetivo es desarrollar una aplicación que corra sobre un Samsung Galaxy Note 4 en un kit Samsung Gear VR.

Puesto que se trata de un teléfono Samsung la aplicación será Android e incluirá diferentes módulos o componentes para:

- Representación de la escena

- Integración: Comunicación de datos entre cámara sensor y aplicación (bluetooth o similar)

- APIs para comunicación con aplicaciones externas etc.

La idea inicial es conseguir crear una experiencia de VR en la que el usuario pueda visualizar el conjunto de álbumes fotográficos personales almacenados en la Cámara de Android o a partir de un directorio especifico.

Los diferentes álbumes de fotos se presentaran en forma de esferas en las que se reflejara una foto representativa del álbum abarcando la esfera en 3D.

Para que resulte una experiencia interactiva y atractiva al usuario las esferas se moverán mediante patrones de animación que también se podrán personalizar para proveer un tipo de experiencia más o menos dinámica y/o aleatoria.

El usuario será capaz de interactuar con cada álbum o esfera mediante gestos que serán capturados por la cámara Intel RealSense.

Mediante diferentes gestos con las manos se permitirá al usuario seleccionar y mover esferas de posición en el espacio 3D.

Las esferas podrán abrirse de la misma forma como se explota un burbuja al tocarlas o hacer un gesto especifico con las manos. Y esto desencadenara la visualización en forma de carrusel o similar del grupo de fotos contenidos en el álbum o esfera.

Al mismo tiempo se pretende que el usuario pueda configurar la experiencia con música que también podrá venir de diferentes fuentes: almacenada en el teléfono o streamed desde otras aplicaciones (como playlists de spotify).

El tipo de música elegida será determinante ya que cambiara por completo la experiencia, por ejemplo: visualización acompañada de música relajante o estimulante.

Se trata de crear una experiencia (estimulante, relajante, distante, cercana...) en un ambiente agradable o personal en el que poder disfrutar de esos momentos en los que accedemos a nuestros recuerdos a través de la visualización de fotos.

Como parte del problema se incluye la evaluación de las diferentes alternativas (software, hardware, etc.) para implementar una solución más o menos acorde con lo que se pretende conseguir.

Objetivos

Dado lo reciente de la tecnología usada (VR y Samsung Gear) se pretende desarrollar por lo menos la experiencia de VR y dentro de lo posible (si no excede el límite de tiempo disponible) y se puede se añadirá la parte de presentación de fotos y la parte interactiva a base de gestos.

El objetivo principal como ya se mencionó anteriormente es crear una aplicación de VR, aunque la implementación difiera de la idea inicial. A tener en cuenta las limitaciones introducidas por las versiones del software de Oculus (todavía en beta) o el hecho de que el Samsung Gear VR no es ni siquiera un producto final sino una versión "solo para desarrolladores". Lo mismo se aplica al kit Intel Real Sense que también se adquirió en versión "solo para desarrolladores".

Motivación

Responde a una motivación personal. La de explorar el futuro de la tecnología, como interactuamos en una experiencia virtual y como jugamos con las sensaciones/emociones en este espacio de forma interactiva.

En los últimos 3-4 años, desde que empecé la primera parte del Master (y diploma CMU) he estado siguiendo activamente el desarrollo de las tecnologías móviles. Y muy en particular la de Android.

En 2013 asistí a la conferencia "Droidcon 2013" en Londres, en la que fui parte del equipo ganador del "Epson Moverio Challenge". El tema de nuestra aplicación fue un "Guía Virtual" para visitantes de la ciudad de Londres que usasen las gafas de realidad aumentada Epson Moverio BT-100.

La aplicación Android usaba los sensores de orientación/posición y otros para mostrar representaciones 3D con animaciones de edificios de interés público como el "London Eye", "St. Paul's cathedral", etc. e información adicional acerca de los mismos.

Este es el video de mi presentación de 2 minutos: https://www.youtube.com/watch?v=6SJa4j3EWCg

Ganar el concurso fue una experiencia inolvidable y una gran inspiración para entrar en el mundo de AR y también de VR.

Desde entonces he seguido de cerca cualquier acontecimiento relacionado con AR/VR y he asistido a presentaciones y eventos de empresas (Google, Vuforia, Meta, Intel), webinars y seguido blogs de tecnología y noticias relacionadas con el tema.

Dado que el Master se basa en el desarrollo de aplicaciones móviles era una ocasión ideal para aprender más sobre VR y sobre la tecnología móvil con la que se complementa. También resulta muy gratificante a nivel personal el hecho de conseguir desarrollar algo en este entorno.

Tecnologías utilizadas

En la fase inicial del proyecto se realizó una evaluación y prueba de concepto del hardware y software de desarrollo que serían usados para la implementación final teniendo en cuenta dos requisitos imprescindibles:

- Las aplicaciones se deberían poder: desarrollar para… y ejecutar sobre… Android.
- La plataforma debería proporcionar la calidad necesaria para garantizar una "3D immersive experience".

Elección del Hardware

Durante la fase inicial del proyecto empecé a evaluar diferentes tipos de hardware que podrían ser usados para mi aplicación de VR.
Inicialmente, investigue soluciones VR de bajo coste, como Google Cardboard u otras simplemente basadas en insertar el teléfono delante de las lentes:

Google Cardboard

Solución basada en Cardboard

Zeiss VR ONE

Me pareció que Cardboard por ejemplo tenía el inconveniente de no proveer sujeción. Y otros headsets que si tenían sujeción no proporcionaban ningún tipo de sensor interno por lo que asumí que programar en ellos cualquier cosa relacionada con posición/orientación o interacción podría necesitar más procesamiento del móvil, lo cual podría influir en el rendimiento y/o la complejidad. Así que los descarte y a continuación entre a investigar los headsets de gama alta.

A principios de 2015 ninguno de los anunciados en el mercado (como Morpheus o HTC Vive) estaban disponibles excepto el Oculus Rift DK2 que se encontraba ya en su segunda versión solo para desarrolladores:

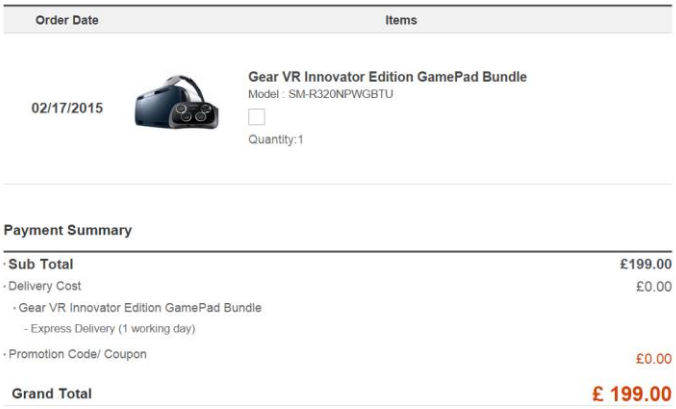
Oculus Rift DK2

Samsung que se había aliado recientemente con Oculus había lanzado el Samsung Gear VR Innovator Edition. Una versión para móviles que cumplía con mi requisito imprescindible de correr sobre Android. Además usaba el Galaxy Note 4 un móvil que ya había adquirido con anterioridad basándome en la posibilidad de usarlo durante el proyecto.

Samsung Gear VR (Headset seleccionado para el proyecto)

Habiendo verificado la buena calidad de imagen y de experiencia VR con un Oculus Rift DK2 (ya de mi propiedad) y conociendo que el dispositivo interno se basaba en el Samsung Galaxy Note 2, decidí apostar por este headset. Usaría entonces mi Galaxy Note 4 y adquiriría por separado el "Gear VR Innovator Edition GamePad Bundle" que

también incluye un gamepad que podría ser útil para las interacciones de la aplicación.



Order Date	Items
02/17/2015	Gear VR Innovator Edition GamePad Bundle Model : SM-R320NPWGBTU Quantity:1

Payment Summary

Sub Total	£199.00
Delivery Cost	£0.00
Gear VR Innovator Edition GamePad Bundle	
- Express Delivery (1 working day)	
Promotion Code/ Coupon	£0.00
Grand Total	**£ 199.00**

Elección del Software

Otra de las ventajas de escoger el Samsung Gear VR era que al estar basado en Oculus proporcionaba un software de desarrollo gratuito; el Oculus Mobile SDK que incluye diversos plugins bastante interesantes y que se podían combinar con otras herramientas.

La documentación del Oculus Mobile SDK en la primera versión que me descargue v0.5.0- Release March 31, 2015) hablaba de dos opciones:

1) ***Developing a Native Mobile VR Application*** que correspondía a desarrollo en código nativo Android en mi caso.

2) ***Developing a Unity Mobile VR Application*** que correspondía a desarrollar una aplicación en la popular Game engine "Unity 3D" usando los correspondientes plugins proporcionados por el SDK.

En mi caso la opción 2 era definitivamente la más apropiada, ya que conocía un poco de Unity y durante los meses de Julio a Septiembre estaría asistiendo a un curso del mismo por lo que podría implementar más funcionalidad a medida que aprendiese más Unity.

Teniendo en cuenta el hecho de que cualquier desarrollo nativo es normalmente más complejo y sabiendo que Unity ofrecía cosas como la Asset Store y una implementación bastante simple basada en prefabs o componentes, Unity era a todas luces la opción favorita.

Sensores de Interacción (Opcional)

Como objetivo opcional me había marcado la posibilidad de incluir interacción humana en la aplicación de VR. Así que tuve que evaluar diferentes tipos de interfaces humanas aun sabiendo que algunas de ellas no estaban todavía en el mercado y eran solo para desarrolladores.

La idea era poder interaccionar con los objetos de la escena, por ello decidí evaluar dos cámaras basadas en sensores infrarrojos y como backup el GamePad proporcionado por Samsung:

1) Intel RealSense camera

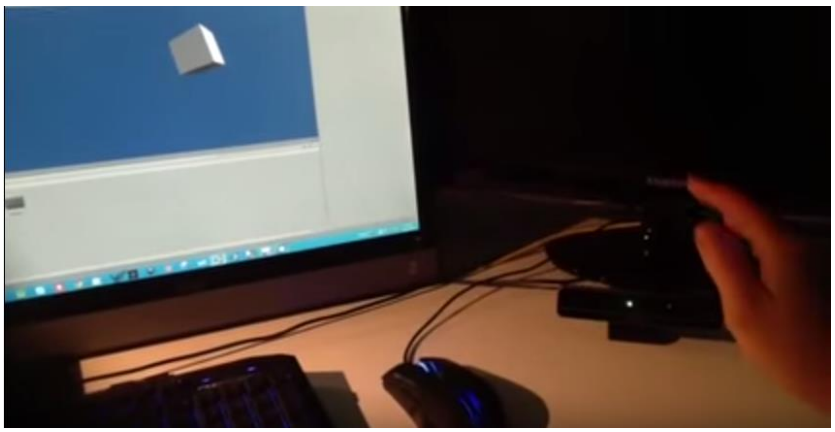
2) Leap Motion

3) Samsung GamePad

Las cámaras de IR ofrecían gran potencial para crear un tipo de interacción novedosa y para darle a la aplicación un toque de originalidad, pero las pruebas de concepto que realice no se adaptaron al hardware y software de que

disponía.

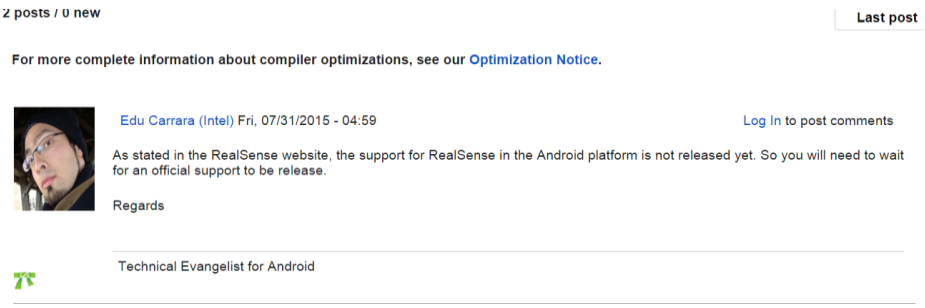
Después de descargar e instalar la opción 1) Intel Real Sense conseguí hacer funcionar un ejemplo sencillo en mi PC.



Mi prueba - Intel Real Sense con Unity 4.6 (video): https://www.youtube.com/watch?v=3n0fYUr-Pg0

Aunque el ejemplo con Unity funcionaba muy bien, me encontré con un problema de versiones de software. El ejemplo de Intel solo funcionaba con Unity 4.6, mientras que para mi proyecto había empezado ya a usar la última versión: Unity 5.

Además, apareció otra limitación y es que aunque la cámara tiene conexión USB estándar no se puede conectar directamente a Android. Véase este extracto de un foro de Intel:



2 posts / 0 new

Last post

For more complete information about compiler optimizations, see our Optimization Notice.

Edu Carrara (Intel) Fri, 07/31/2015 - 04:59

Log In to post comments

As stated in the RealSense website, the support for RealSense in the Android platform is not released yet. So you will need to wait for an official support to be release.

Regards

Technical Evangelist for Android

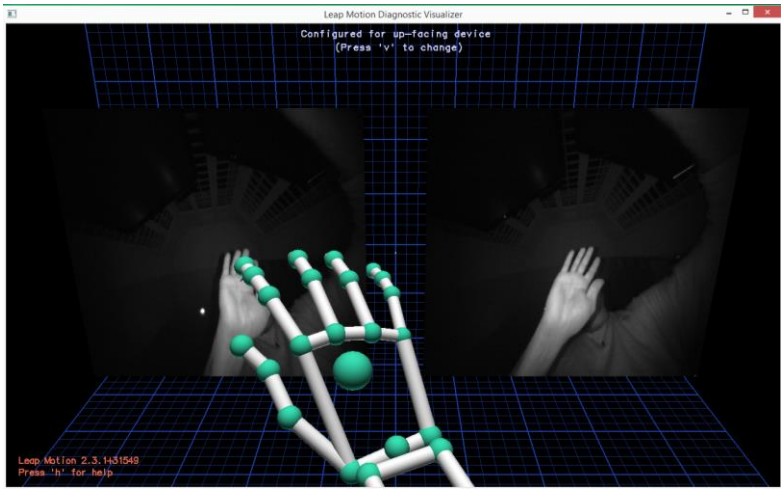
Y de todas formas si el teléfono estuviese “enganchado” (docked) al headset, el puerto USB no estaría disponible para la cámara Intel tampoco.

Esta era una limitación importante ya exigía implementar la interacción por separado en un PC y comunicar la posición de la mano / tipo de gesto a la aplicación móvil por Bluetooth o API. Una solución demasiado costosa en términos de desarrollo y complejidad.

La segunda opción Leap Motion prometía, porque antes de adquirir el hardware había visto que este sensor ofrecía un kit que permitía incorporar el sensor a un headset.

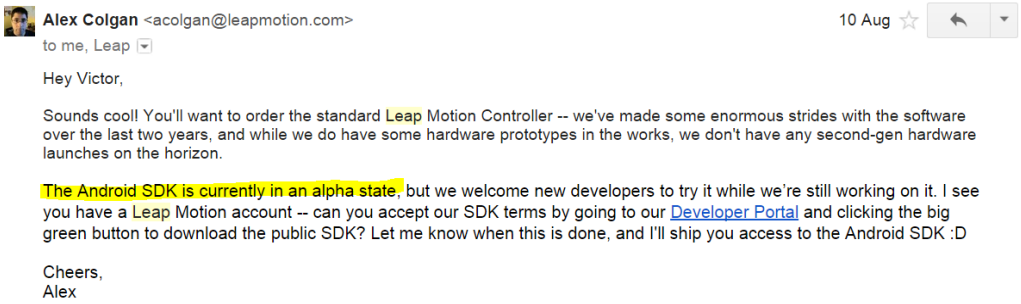
Sensor Leap Motion Leap Motion montado en Oculus Rift DK2

Pude probar el sensor con éxito en mi PC como se muestra en el siguiente screenshot:



Probando el sensor Leap Motion en mi desktop

Pero por desgracia también tuve que descartar el uso de Leap Motion porque según ellos: la versión final para del SDK de Android no estaba todavía disponible. Solamente una versión Alpha que al final no puede conseguir descargarme de su "developer portal":


Alex Colgan <acolgan@leapmotion.com> 10 Aug

to me, Leap

Hey Victor,

Sounds cool! You'll want to order the standard Leap Motion Controller -- we've made some enormous strides with the software over the last two years, and while we do have some hardware prototypes in the works, we don't have any second-gen hardware launches on the horizon.

The Android SDK is currently in an alpha state, but we welcome new developers to try it while we're still working on it. I see you have a Leap Motion account -- can you accept our SDK terms by going to our Developer Portal and clicking the big green button to download the public SDK? Let me know when this is done, and I'll ship you access to the Android SDK :D

Cheers,
Alex

A falta de opciones para implementar interacción por gestos tuve que recurrir a la opción de backup: Opción 3) uso del Samsung GamePad.

Por suerte este resultó mucho más fácil. Dado que es también fabricado por Samsung, la integración entre teléfono móvil, headset y GamePad fue muy simple y funcionó a la perfección.

Además dada la popularidad de Unity pude comprobar que existían numerosos ejemplos online sobre cómo implementar la funcionalidad de los botones dentro de la aplicación (botón de disparo, etc.).

Con mi "environment setup" completado, empecé a desarrollar la aplicación.

Dado que no tenía la componente gestual tuve que redefinir el concepto de la experiencia, así que en lugar de perseguir la idea de visualizar imágenes pensé en hacer algo todavía más interactivo: un juego basado en esferas que se destruyen cuando el usuario lanza proyectiles en forma de cubo en el espacio 3D.

La aplicación pasó entonces a llamarse: ***Spheres36VR***.

Arquitectura de la aplicación

La arquitectura de la aplicación ***Spheres36VR*** se basa en la generación de un apk Android que se ejecuta en el teléfono que a su vez se inserta en el headset de VR.

El hardware incluye:

1) Samsung Galaxy Note 4
   (con Android Lollipop 5.0 y después actualizado a Lollipop 5.1)
2) Samsung Gear VR Innovator Edition

El software de desarrollo se ejecutó sobre Windows 8.1 Pro e incluye:

1) Android SDK
2) Oculus Mobile SDK v0.6.0.1
3) Unity 3D v.5.1.2f1 (64-bits) (Personal Edition)

El Android SDK, el Oculus Mobile SDK y Unity 3D se utilizan para compilar y generar el apk que se ejecutara sobre el Smartphone Android.

La lógica de la aplicación se ha desarrollado enteramente en Unity 3D (Personal Edition). Esta edición ofrece acceso gratuito al software para proyectos de desarrollo.

1. Descripción de la aplicación

1.1. Funcionalidad

La aplicación es un juego interactivo en el que el jugador se mueve por la escena usando el GamePad y el “touch pad” del visor.

Al abrir la app desde Android aparece un mensaje que indica que el teléfono debe insertarse en el visor.

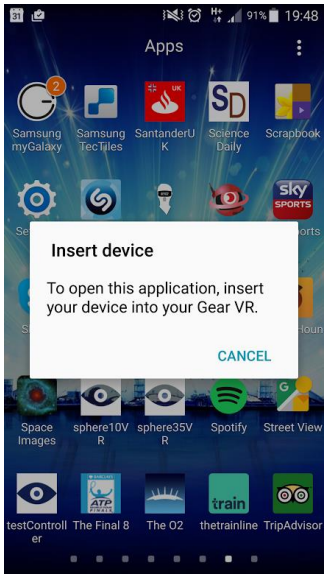


Ilustración 1.- Insertar teléfono en visor

Al insertar el dispositivo en el puerto USB, se reproduce un sonido y el visor detecta cuando este se encuentra ajustado en la cabeza (mediante un sensor en la parte interior derecha). Entonces se inicia la aplicación y aparece una “splashscreen” con un cubo 3D y un texto que lee: “made with unity”.

A continuación se carga la aplicación en memoria y se ejecuta.

La primera cosa que aparece es un plano con un cielo azul (con música de fondo) y un mensaje “READY PLAYER ONE” seguido de una cuenta atrás “3 2 1”. A continuación se van generando en la escena objetos de tipo esfera 3D que se posicionan en diferentes lugares aleatoriamente.

El jugador puede entonces moverse por la escena mientras se muestra un contador de tiempo que indica el tiempo restante. El objetivo es conseguir destruir el máximo número de esferas posibles.

La única forma de destruir una esfera es usando el botón disparo, que produce el lanzamiento de un cubo de color amarillo (con un sonido asociado). Si el cubo lanzado choca contra una esfera, se elimina la esfera de la escena.

Al destruirse la esfera el programa inicia un sistema de partículas que representa fragmentos de la esfera alejándose del centro de la misma creando así un efecto de explosión.

El número de esferas a generar está limitado por el tiempo. Cuando el tiempo se agota, el sistema detiene la generación de esferas y muestra el mensaje “You ran out of time”. Esto indica al usuario que la partida ha terminado y muestra también el número de esferas destruidas del total generadas durante la partida.

El mensaje se queda en pantalla hasta que el usuario pulse el botón de disparo. Esto iniciará automáticamente una nueva partida, eliminando de la escena todas las esferas restantes de la partida anterior e inicializando contadores y variables de estado para un nuevo juego.

1.2. Estructura del programa

El programa en Unity está estructurado siguiendo los estándares vistos en el curso de Unity del Master.

Está constituido por una única escena en la que existen los siguientes GameObjects:

GameObjects de la Escena Principal	Descripción
txtGameHeader	3DText usado para mostrar mensaje de inicio y final de partida.
Txtdebug	3DText usado para mostrar el resultado final de la partida.
Directional Light	Luz direccional usada para dar más brillo a la escena.
ThrowingHand	Objeto padre usado para la generación de objetos proyectiles.
Crosshair3D	Cursor 3D.
Plane	Superficie sobre la que se mueve el jugador.
GameObjectStart	Objeto vacío al que se asocial el script startup (“main loop”).
OVRPlayerController	Controlador de jugador. Objeto padre que se

	incorpora en la aplicación al importar el "UnityIntegration.unityPackage" del Oculus Mobile SDK. Contiene la OVRCameraRig o cámara principal.

Por otra parte, los assets se agrupan en diferentes carpetas. En la siguiente tabla enumero los más importantes:

Carpeta	Elementos	Descripción
Assets\Scenes	scene_spheres	Escena principal del juego.
Assets\Prefabs	Sphere1	Tipo de esfera 1 usado como plantilla para la generación aleatoria de esferas.
	Sphere2	Idem pero tipo 2.
	Sphere3	Idem pero tipo 3.
	ThrowingCube	Tipo de cubo usado como plantilla de objeto proyectil usado para la generación de proyectiles (jugador pulsa botón disparo)
	explosion	Sistema de partículas usado para mostrar fragmentos de objetos despedidos desde el centro de una esfera al destruirse esta.
	MyCrosshair3D\ Crosshair3D	Cursor tipo cruz que se usa para indicar al jugador si hay algún objeto en la línea de visión. Esta cruz es un ejemplo del Oculus Mobile SDK y es bastante útil ya que utiliza un raycast que al colisionar con un objeto hace un pequeño efecto de zoom de la flecha que da a entender al jugador que se está mirando ("gaze") a un objeto. Es útil porque da sensación de profundidad al mirar a los objetos de la escena y también indica para donde saldrá disparado el proyectil.
Assets\Textures	Crosshair	Textura gráfica del usada por el Crosshair3D.
Assets\BubblePop_Effect	BubblePop	Sistema de Partículas mostrado al destruirse una esfera.
Assets\Scripts	GameManager	Clase estática con variables para controlar el estado del juego.
	Startup	Bucle principal del juego encargado de actualizar la escena a cada frame. También encargado de inicializar la partida y generar las esferas.
	ThrowObject3D	Script asociado al GameObject ThrowingHand que a su vez está asignado a la OVRCameraRig o cámara principal. Este script comprueba en el evento LateUpdate() si se ha apretado el botón de disparo. Y en ese caso, ejecuta el sonido de disparo (propiedad shootSound) y instancia un objeto proyectil en la posición del centerEyeAnchor de la

		OVRCameraRig (la posición de la cabeza del jugador en la escena). Después añade una fuerza relativa al objeto para darle así una velocidad de lanzamiento y que el objeto (cubo) salga así disparado.
	DestroyCubes	Script asociado al objeto cubo que al detectar una colisión instancia el sistema de partículas explosión en la posición donde se produjo esta colisión. También destruye el objeto e incrementa el contador de esferas destruidas.
	Crosshair3D	Proporcionado por Oculus. Implementa el cursor 3D basado en la vista del jugador (línea de visión).
Assets\Audio	weapon_player	Sonido de corta duración usado para indicar disparo de proyectil.
Assets\Music	Bees	Tema principal o background music del juego.
Assets\Materials	…	Diferentes materiales usados por prefabs y otros objetos.
Assets\OVR	Prefabs, Scripts, etc.	Objetos importados por el "UnityIntegration.unityPackage" del Oculus Mobile SDK.
Assets\Plugins	Android	Carpeta que contiene el AndroidManifest.xml file donde podemos activar permisos Android etc.
Assets\Plugins\assets	Oculussig_<<disp_id>>	Carpeta que contiene el fichero "Oculus signature file". Contiene el identificador del dispositivo Android sobre el que se ejecutara la aplicación. Solo puede usarse un dispositivo en la versión gratuita de Unity.
Assets\Resources	…	Imágenes que podrían usarse como textura para las esferas. Esta funcionalidad no está implementada.

1.3. Lógica del programa

La lógica del programa se inicia a partir del GameObjectStart, que como ya se indico es un objeto vacío que tiene asociado el script "startup.cs" con el bucle principal de la aplicación.

Este script contiene la función "Start()" común a los scripts de Unity que se usa para invocar la función de generación de esferas llamada "CreateSpheres()".

Dentro de esta función local se comprueba el estado del juego para inicializar variables y empezar/continuar generando esferas.

El estado del juego se controla desde la función "Update()". Esta función que se ejecuta a cada frame es la que comprueba si la partida sigue en marcha y actualiza el contador de tiempo y el estado gestionados a partir del script estático GameManager.

El lanzamiento de proyectiles y las colisiones se controlan desde los otros scripts: ThrowObject3D y DestroyCubes respectivamente.

1.4. Importación de Objetos

En el siguiente video incluyo una grabación (sin sonido) de como he importado el “UnityIntegration.unityPackage” del Oculus Mobile SDK dentro de mi aplicación en Unity 3D.

https://www.youtube.com/watch?v=Gjt7hBK1CoI

1.5. Generación del “Oculus Signature File”

Necesitamos generar un “signature file” para poder ejecutar aplicaciones en el dispositivo. Las aplicaciones que acceden la funcionalidad de VR del Oculus Mobile SDK se deben firmar con una firma única para que puedan acceder a las funciones de bajo nivel mediante las APIs proporcionadas por el SDK.

Esta no es la forma habitual de desplegar aplicaciones en la Oculus Store, normalmente solo necesitaríamos APKs adecuadamente firmados, pero se requiere para el despliegue en dispositivos de desarrollo durante el testing.

El “Oculus Developer Site” contiene una herramienta de generación de “signature file” y puede accederse en esta dirección:

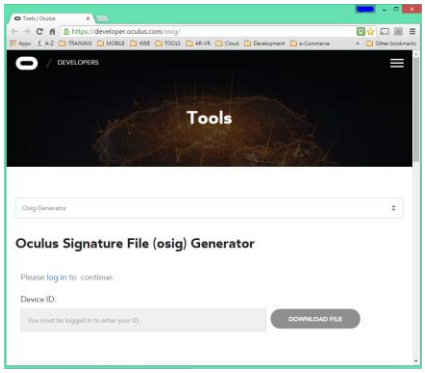


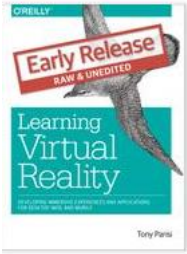
Ilustración 2.- Herramienta de generación del signature file

La página contiene instrucciones sobre cómo obtener el Device ID que se usa para generar el “Signature file”.

A partir de una ventana de terminal ejecutaremos el comando “adb devices”. Este imprimirá por pantalla el identificador único del dispositivo. Copiando este ID de dispositivo en la página de Oculus (osig) y pulsando en ***Download File*** obtendremos el fichero “Signature file” para nuestro dispositivo”.

Este fichero debe mantenerse en un lugar seguro ya que se necesitara para todos los Gear VR despliegues que hagamos independientemente de la herramienta de desarrollo que utilicemos.

Nota: Este apartado es la traducción de un fragmento del libro...



Learning Virtual Reality

by Tony Parisi

Published by O'Reilly Media, Inc., 2015

2. Despliegue en Android

Unity empaqueta la lógica en un apk estándar usando la opción de menú File – Build. Las opciones de despliegue se incluyen en File – Build Settings y en este caso se utilizaron parámetros especiales para que la aplicación se ejecutase en el VR headset. Estos difieren un poco de los que se usarían si la aplicación se ejecutara en un desktop.

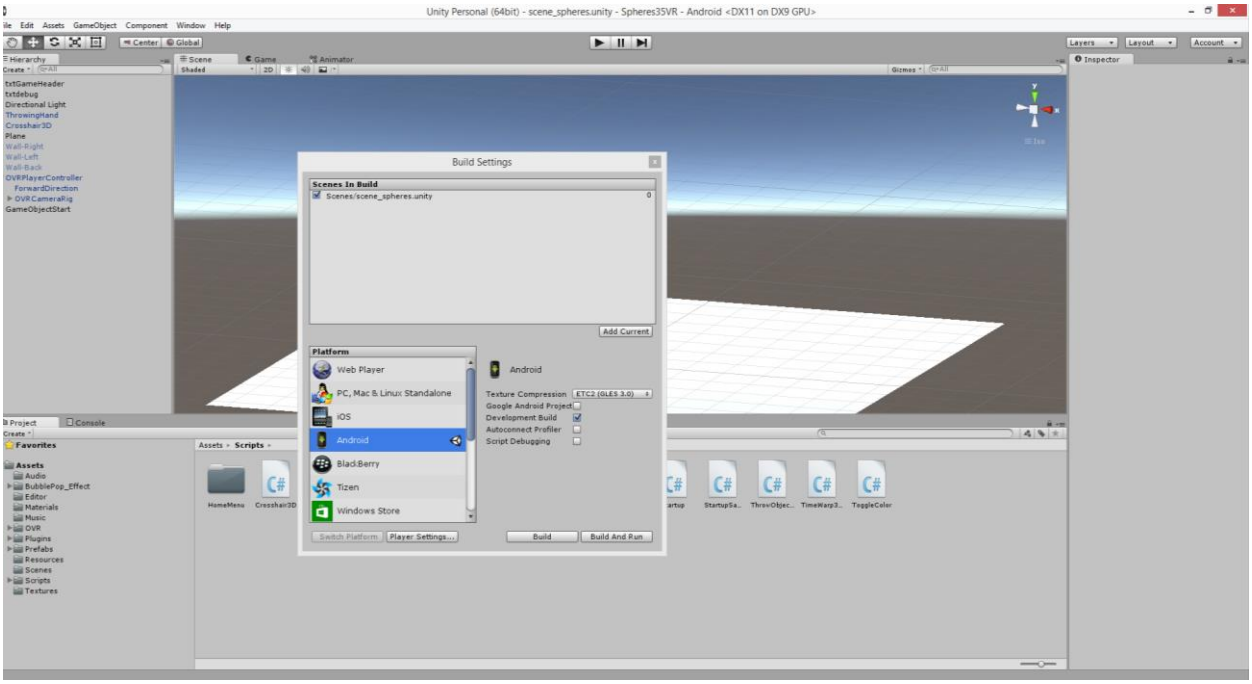


Ilustración 3.- Unity3D - File - Build Settings

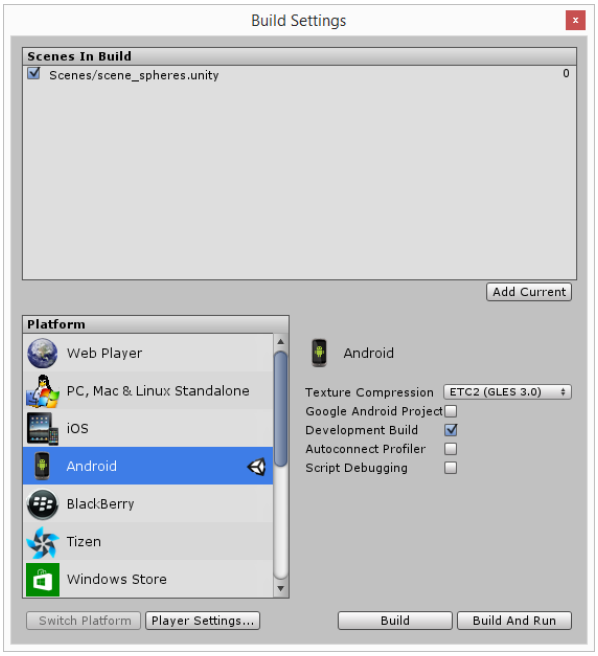
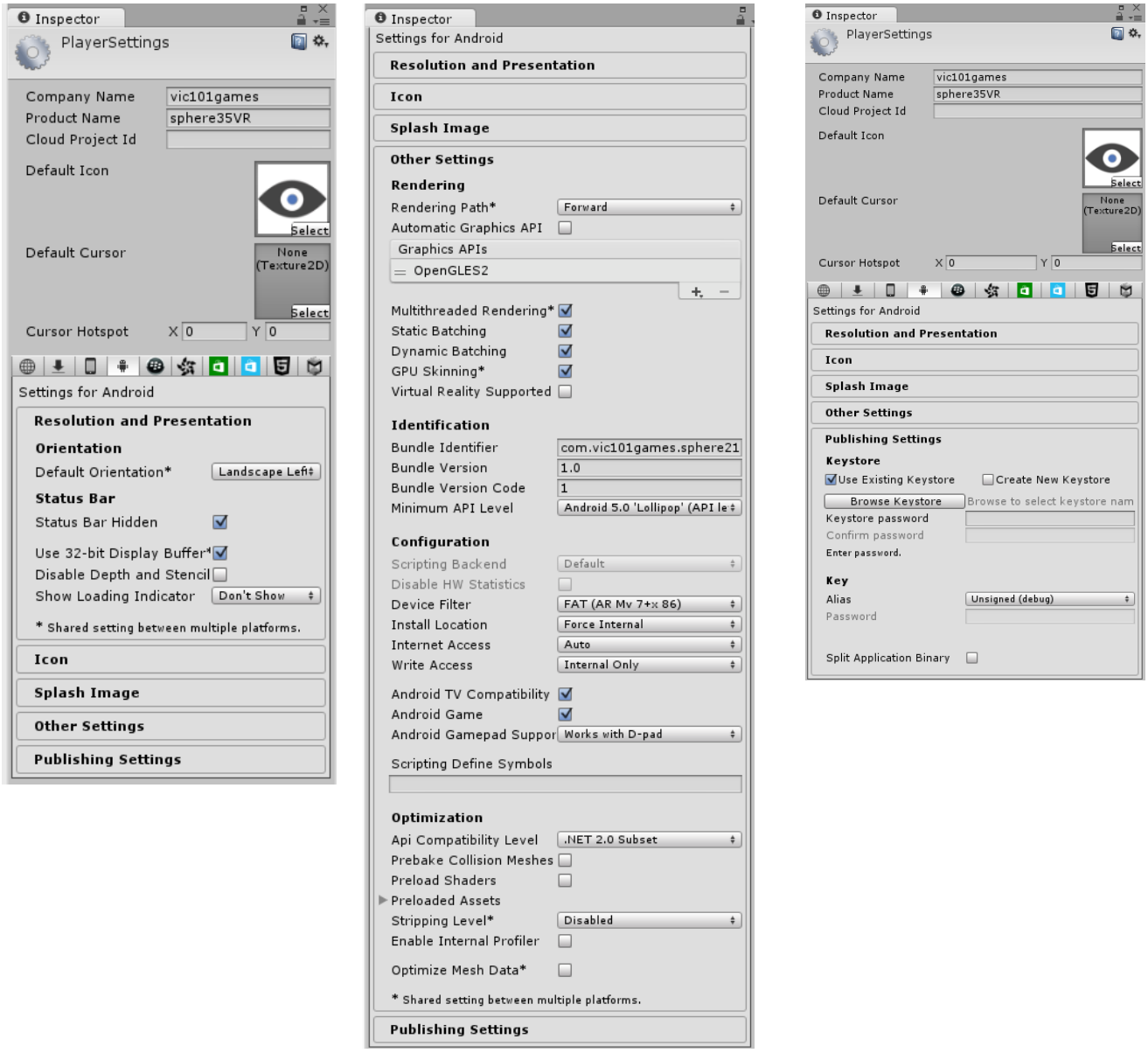


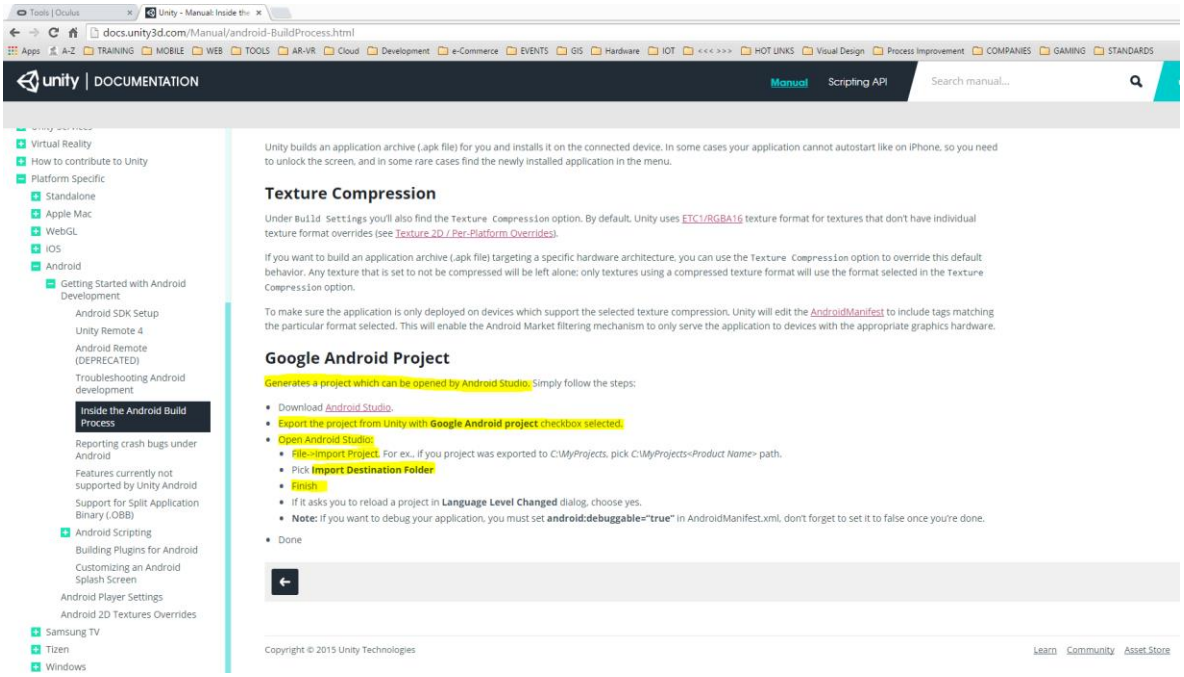
Ilustración 4.- Player Settings - Resolution / Other / Publishing Settings



3. Generación del proyecto para Android Studio

Unity es capaz de generar un apk para Android, pero también podemos crear una versión para depurar/modificar o ampliar desde Android Studio.

El siguiente enlace muestra cómo hacerlo: http://docs.unity3d.com/Manual/android-BuildProcess.html



A continuación muestro un conjunto de screenshots con los pasos que seguí para hacerlo en mi PC:

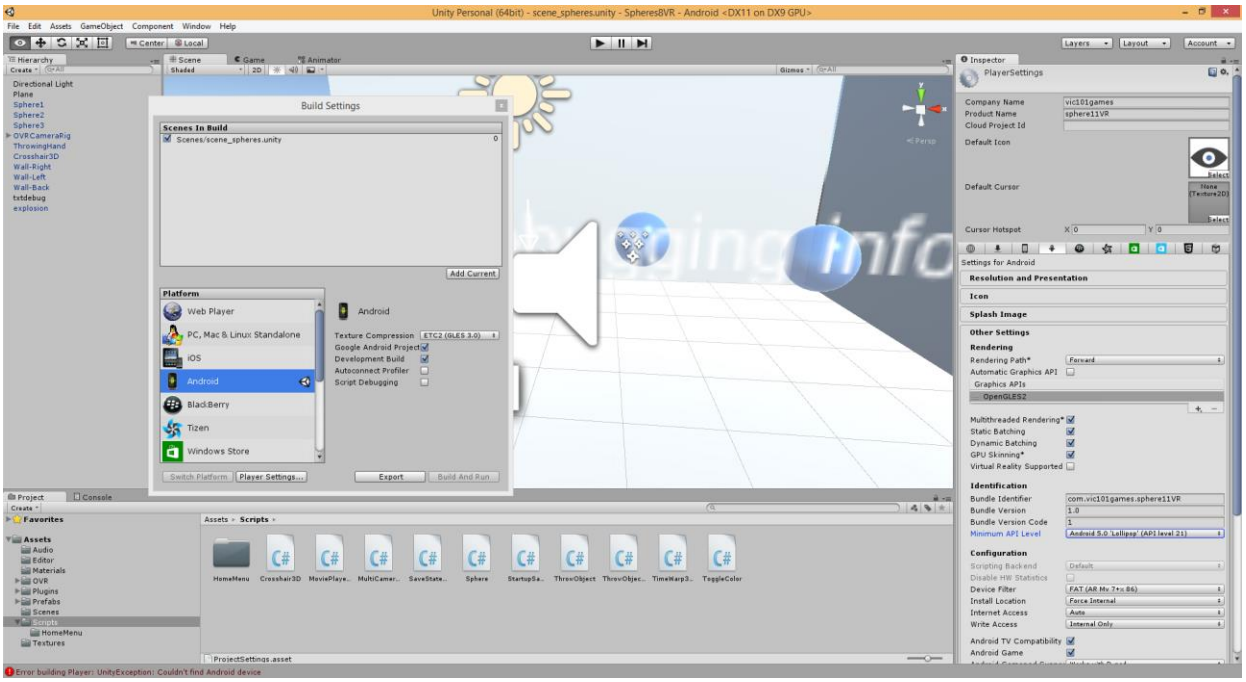


Ilustración 5.- Paso 1 - Seleccionamos opción Google Android Project y Development Build

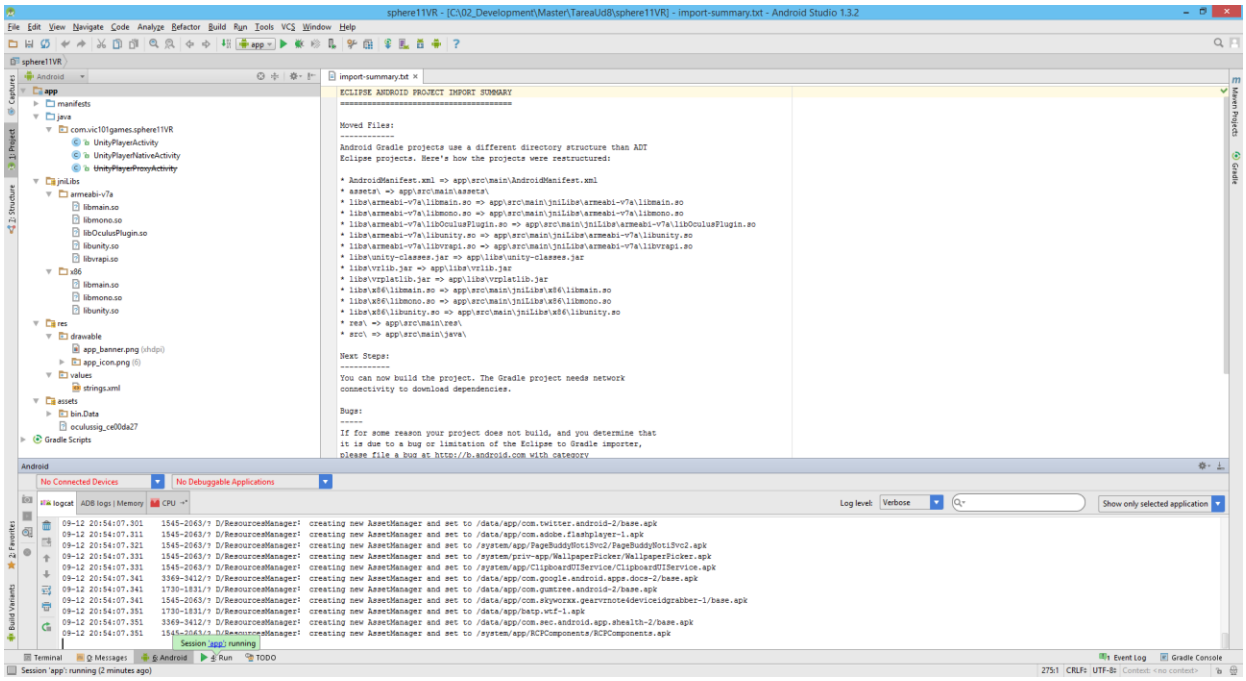


Ilustración 6.- Paso2 - Importamos el proyecto en Android Studio

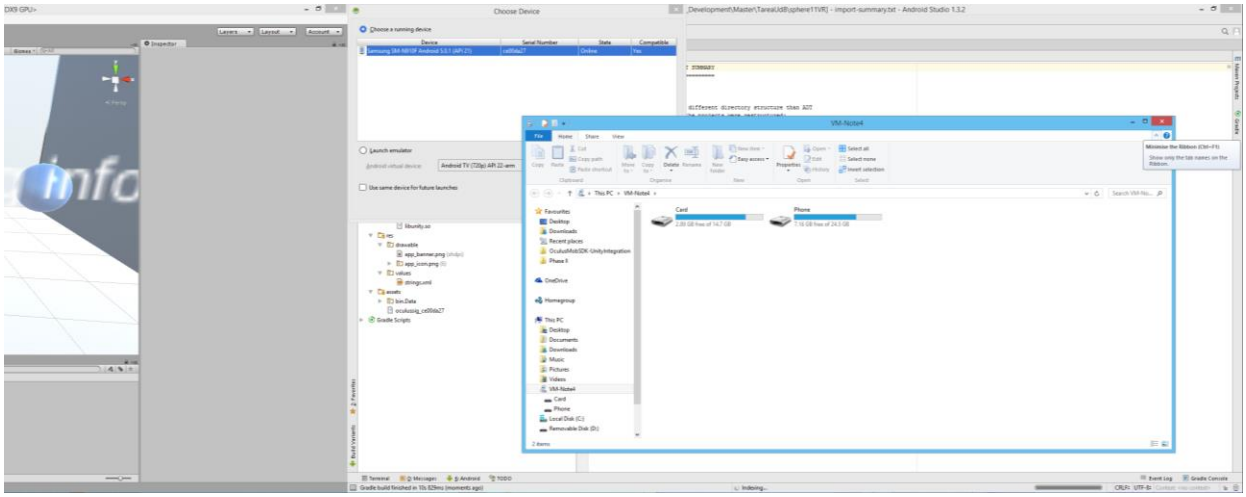


Ilustración 7.- Paso3 - Lanzamos la aplicación en el teléfono desde Android Studio

Unity habrá generado automáticamente todos los paquetes necesarios para que la app se ejecute sin necesitar de ningún otro componente adicional. Al lanzarla nos pedirá que insertemos el telefono en el visor, tal y como lo haría si la abriésemos directamente desde Unity o por separado.

Capítulos adicionales

Estado del Arte de VR

Si los augurios se cumplen, VR se va a convertir en la forma de comunicación del futuro, en la que no solamente intercambiaremos información, sino también experiencias.

Un nuevo medio en el que compartiremos vivencias y que nos permitirán alcanzar la telepresencia (simulada) más allá de la barrera que suponen los dispositivos de interacción habituales (móvil, teclado, ratón, pantalla...) y del contenido digital al que estamos acostumbrados.

Una "VR immersive experience" proporciona una forma de interacción completamente personal pero que puede también compartirse a través de la red. VR no es simplemente una tecnología: con toda probabilidad pasara también a convertirse en un fenómeno social.

Aunque durante muchos años la componente tecnológica no ha estado a la altura si hay finalmente una intención real por parte de diversas empresas multinacionales, start-ups y público consumidor por conseguir que tanto el hardware como el contenido adquieran la dimensión necesaria para producir experiencias VR de calidad.

Hardware

Sin los últimos avances en procesamiento gráfico y miniaturización de hardware y sensores no sería posible conseguir crear efectos creíbles para el cerebro humano. Pero el increíble nivel de integración en chips y el aumento de la velocidad de procesamiento han alcanzado cuotas que permiten desplegar aplicaciones VR incluso en dispositivos móviles que son asequibles al público en general, como por ejemplo Google Cardboard.

Oculus Rift

Sony Morpheus

Samsung Gear VR

HTC Vive

El “form factor” del hardware es ahora mismo el de un visor o “headset”, sin embargo ya existen empresas que investigan como extenderlo a otros sentidos aparte del de la visión o del sonido. Lo hacen combinando diferentes sensores en el mismo espacio en el que el usuario ejecuta la experiencia y sobreponiendo objetos del mundo virtual en el mundo real. Un ejemplo que realmente promete en el espacio del gaming: The Void.

https://thevoid.com/

https://youtu.be/cML814JD09g

Software

Avances en técnicas de diseño gráfico, game engines, etc... van a permitir que programar o crear experiencias de VR sea una tarea al alcance de cualquiera. Un hecho que constituye un "milestone" a superar ya que hasta ahora este tipo de experiencias solo aparecían en soluciones de alto coste en parques temáticos, museos, etc.

Existen incluso entusiastas y empresas que ya proponen soluciones open source como se muestra a continuación.

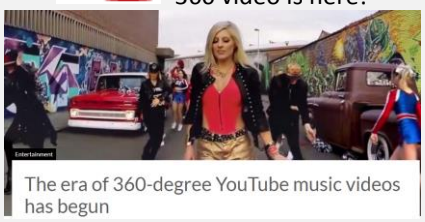
OSVR - http://www.osvr.org/

http://www.freevr.org/

Realidad y No realidad (Contenido Digital)

Aunque el termino realidad virtual puede interpretarse como una proyección de la realidad su verdadero potencial reside en la posibilidad de crear experiencias "imposibles" es decir, que no existen en la realidad. Desde el punto de vista de los videojuegos por ejemplo, podemos convertir al usuario de VR en un personaje de ficción que puede moverse a libertad por mundos virtuales que no se asemejan para nada a lo que conocemos como realidad.

Previo a esta tecnología, los mundos de fantasía existían en libros, dibujos, arte, TV y eran interpretados subjetivamente según el lector o espectador. Ahora y gracias a esta tecnología existe la posibilidad de crear mundos que diferentes espectadores van a visualizar de la misma manera y con un nivel de inmersión total en la experiencia. Prácticamente, podemos hablar de la creación de nuevos mundos que solo existen en el contexto de VR: contenido puramente digital que toma vida propia gracias a la interacción humana en tiempo real y a la representación en 3D.

Nuevos formatos de contenido digital	***Realidad en 360 grados***
YouTube 360 video is here!  http://www.dailydot.com/entertainment/360-degree-youtube-music-video-kolor/	360 cameras and camera rigs  https://www.google.co.uk/get/cardboard/jump/  http://www.bbc.co.uk/news/technology-33704103

Aplicaciones

Han aparecido y están apareciendo numerosas aplicaciones prácticas para la realidad virtual, desde sus inicios con simuladores de vuelo (que han existido durante muchos años) a nuevas aplicaciones en formación, visualización de objetos en diseño y arquitectura, videojuegos, etc. Existen diversos estudios que sugieren que el mundo de la VR se va a convertir en sí mismo en un nuevo segmento de la economía y se estima que el nuevo mercado puede alcanzar los **$150B** en 2020.

> Analysis:
> "We forecast that AR/VR could hit $150B revenue by 2020,
> with AR taking the lion's share around $120 billion and VR at $30 billion"
>
> http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.VdREqflVhBc

Conclusiones

Como conclusión final me gustaría resaltar los objetivos conseguidos.

En primer lugar, he realizado un extenso trabajo de investigación y dedicado un gran número de horas a probar dispositivos, software y considero que ha sido un éxito total sobre todo la parte de la evaluación de la tecnología.

He aprendido muchísimo en todo lo que respecta a VR.

Aunque me desvié de mi concepto inicial (experiencia de visualización de fotos) creo que fue la decisión acertada el optar por implementar una funcionalidad menos original pero con menos retos de implementación. En otro caso posiblemente me hubiera excedido en los tiempos y no hubiera sido capaz de presentar una aplicación final.

Quizá no conseguí todos mis objetivos adicionales, basados en usar interacción humana a base de gestos, debido en parte a que los kits de desarrollo no estaban todavía disponibles o no tuve acceso directo a los mismos pero terminado el proyecto tengo una aplicación de VR que funciona muy bien y que si quisiera podría extender para publicarla en el Marketplace de Android o la Oculus Store.

Además mi backup plan para usar un GamePad ha funcionado con lo que también ha conseguido experimentar otras alternativas a los gestos.

En mi opinión un 90% de los objetivos conseguidos, mucho más de lo que hubiera esperado antes de empezar el proyecto.

Líneas abiertas

A partir de aquí llega la parte más interesante. Seguir explorando AR / VR y quizás en 2016 adquirir un visor en versión final (Oculus Rift / Gear VR) y publicar mi aplicación en la Oculus Store.

Consideraciones personales

Es una satisfacción poder explorar nuevas tecnologías y he disfrutado mucho durante el Master con Glass, OpenCV, AndroidWear, AndroidTV, etc.

Considero que esta tecnología de VR en particular va a constituir un cambio radical en el futuro de todos. Como "technology enthusiast" y aficionado a la ciencia ficción recomiendo la lectura del libro "Ready Player One" (by Ernest Cline). Este libro fue una de mis inspiraciones para empezar a creer en esta tecnología.

Anexos

Listado de fuentes entregadas / Código fuente en GitHub

Los fuentes se encuentran disponibles en Poliformat.