

Universidad de Cádiz

Escuela Superior de Ingeniería



Desarrollo de Sistemas Hipermedia

MusicAR

Proyecto final

Proyecto realizado por:

Jesús Serrano Gallán

Índice

1. Introducción	2
1.1. Motivación	2
1.2. Idea y objetivos del proyecto	2
1.3. Repositorio de GitHub	2
2. Planificación	3
2.1. Tareas	3
2.2. Planificación temporal	3
3. Herramientas utilizadas	5
4. Funcionamiento	5
5. Arte	9
5.1. Tarjetas	9
5.1.1. Centro	9
5.1.2. Bombo	9
5.1.3. Tambor	10
5.1.4. Platillos	10
5.1.5. Bocina	11
5.1.6. Voz	11
5.2. Modelos	12
5.2.1. Centro	12
5.2.2. Bombo	12
5.2.3. Tambor	13
5.2.4. Platillos	13
5.2.5. Coche	14
5.2.6. Personaje	14
5.3. Menús	15
5.3.1. Menú principal	15
5.3.2. Cómo usar	16
5.3.3. Créditos	17
6. Conclusiones	18
6.1. Problemas encontrados	18
6.2. Posibles mejoras futuras	18

1. Introducción

A continuación se procede a presentar MusicAR, el proyecto final realizado para la asignatura de Desarrollo de Sistemas Hipermedia.

1.1. Motivación

La música es algo fascinante y hoy en día, gracias a la tecnología, cada vez se vuelve más sencillo crear ritmos y melodías gracias a algún que otro programa. Todos hemos tonteado con aplicaciones como Garage band en el iPad de algún familiar o amigo y lo que se pretende conseguir con esta aplicación es replicar un comportamiento similar pero aplicado a la realidad aumentada.

1.2. Idea y objetivos del proyecto

La idea es muy sencilla: controlar una serie de instrumentos utilizando marcas de realidad aumentada. Esto suena genial pero hay que tener en cuenta que al ser una aplicación de realidad aumentada no se podrán conseguir grandes resultados ni sonatas de Mozart, pero el objetivo sería poder marcar nuestros propios ritmos básicos gracias a la aplicación. También ofrecer unos modelos y herramientas atractivas al usuario estaría dentro de nuestros objetivos.

1.3. Repositorio de GitHub

El unitypackage del proyecto, la presentación, este documento y un vídeo de demostración, se encuentran depositados en <https://github.com/Jesega/MusicAR>.

2. Planificación

A continuación se encuentra especificada la planificación que se ha seguido para el proyecto.

2.1. Tareas

Para la elaboración del proyecto se diferenciaron las siguientes tareas:

- **Lógica de la aplicación:** investigación e implementación de los scripts necesarios para que la aplicación tenga el comportamiento deseado.
- **Tarjetas:** Búsqueda y edición de las imágenes que se usarán como targets para la aplicación. Posteriormente se imprimirán para su uso.
- **Modelos:** Creación de los modelos 3D que se usarán en la aplicación
- **Sonidos:** Búsqueda de los sonidos que se utilizarán en la aplicación. Edición de los mismos si fueran necesarios
- **Menús:** Elaboración de los menús de los que dispondrá la aplicación final.
- **Pruebas del sistema:** Probar el correcto funcionamiento de la aplicación y realizar correcciones si esto fuera necesario.
- **Vídeo y presentación:** Grabación del vídeo para la entrega. Preparación de la presentación en directo.

2.2. Planificación temporal

A continuación se presenta un diagrama de Gantt con la planificación aproximada que se ha seguido durante el desarrollo del proyecto. Hay que tener en cuenta que al ser el yo único participante, he tenido mucho trabajo pero no he tenido que perder tiempo para organizarme con compañeros ni he sufrido las incomodidades de tener que distribuir el trabajo. Básicamente lo primero que hice fue plantear la lógica de la aplicación y resolverla hasta un punto que cumpliera con los objetivos del proyecto. A continuación me puse a elaborar los targets para la realidad aumentada, los sonidos que se reproducirían y los modelos que quería que aparecieran. Agregué después una serie de menús para explicar al usuario cómo funciona la aplicación y exponer los créditos y finalmente elaboré el vídeo y preparé la presentación del proyecto. Durante todo el proceso de desarrollo, a partir del punto en el que la aplicación ya ofrecía una cierta funcionalidad, fui haciendo pruebas con respecto metía pequeños cambios para afinar el resultado final.

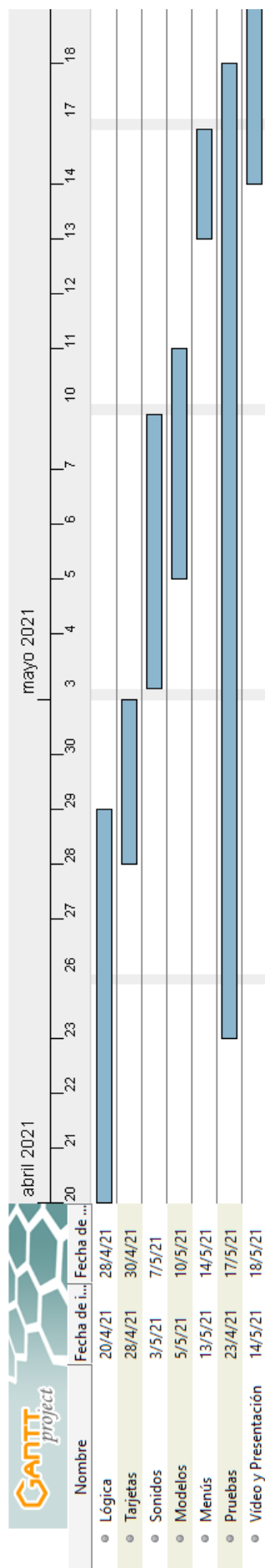


Figura 1: Diagrama de Gantt con la planificación del proyecto

3. Herramientas utilizadas

Se presenta a continuación un listado de todas las herramientas utilizadas durante la elaboración del proyecto y el fin con el que se han utilizado.

- **Unity**[1]: motor principal de desarrollo del proyecto. Se han elaborado varias escenas, una para el uso de la realidad aumentada como tal y otra para los menús de la aplicación. Para este proyecto, lo que más se ha utilizado de Unity base es el API para trabajar con audios que ofrece la clase AudioSource. [3]
- **Vuforia Engine**[8]: kit de desarrollo de realidad aumentada empleado para crear la aplicación con Unity.
- **Iriun webcam**[6]: software empleado para poder utilizar el móvil como webcam del ordenador y hacer las pruebas de la aplicación.
- **Blender**[4]: software utilizado para la creación de los modelos presentes en la aplicación.
- **Audacity**[2]: herramienta gratuita de software libre utilizada para la edición de los sonidos que necesitaban algún retoque. Por ejemplo, extraer sonidos diferentes de una pista de audio que contenía varios instrumentos sonando.
- **Diagrams.net**[5]: herramienta de dibujado online. Con ella se pueden hacer gran cantidad de cosas pero yo la he usado para personalizar las marcas de realidad aumentada a partir de las generadas automáticamente en la página <https://shawnlehner.github.io/ARMaker/>

4. Funcionamiento

Si simplificamos al máximo el objetivo de la aplicación, nos encontramos con que para reproducir ritmos sencillos es necesario ser capaz de dos cosas:

- Reproducir sonidos.
- Alterar la frecuencia con la que se reproducen.

Es decir, necesitamos una forma de hacer que suene el instrumento que queramos (de forma repetida, constantemente) y de poder modificar cuánto tiempo pasa entre reproducción y reproducción de cada sonido.

Para reproducir o no un cierto sonido, lo que haremos es relacionarlo con una marca de realidad aumentada y que cuando dicha marca se detecte, se reproduzca el sonido con el que la hemos relacionado. Cuando se deje de detectar, el sonido parará. Hasta ahí es muy sencillo.

La parte de poder variar la frecuencia con la que se reproducen los sonidos, es la complicada. Para ello, utilizaremos, algo que también podemos hacer que varíe en la realidad: las distancias entre marcas. Lo que haremos es usar una marca de referencia (a la que llamaremos marca de referencia). El resto de marcas, las que están relacionadas con un sonido, las llamaremos marcas de sonido. Cuando la aplicación las detecte, estas marcas de sonido reproducirán su audio con una frecuencia inversamente proporcional a la distancia a la que se encuentra la marca del sonido de la marca de referencia. Es decir, teniendo en pantalla tanto la marca de referencia como la marca de sonido, cuanto más acerquemos la marca de sonido a la referencia, más rápido se reproducirá el sonido repetidamente. Aprovecharemos

esta característica que queremos implementar e incluiremos también la posibilidad de modificar el volumen con el que se reproduce el sonido. Veamos todo esto más detenidamente. Contamos con una marca de referencia en la superficie en la que queremos colocar el resto de marcas de sonido. Esta marca de referencia no reproduce ningún sonido y su única utilidad es proveer información para los cálculos que deben ejecutar los scripts. A partir de la posición de la marca de referencia, tomamos la superficie como si fuera una especie de eje de coordenadas. Para que sea más simple comprenderlo, usaremos únicamente el primer cuadrante. Para nuestra aplicación hemos decidido que el volumen decrementaría, con respecto a aumenta la coordenada Z y el período de tiempo entre reproducción y reproducción, incrementaría junto con la coordenada X.



Figura 2: Figura de referencia

De esta forma, podemos modificar los ritmos y la presencia de los sonidos utilizando las tarjetas de realidad aumentada, sin tener que tocar el dispositivo móvil en el que correría la aplicación. En la siguiente imagen podemos ver una simplificación de la distribución de varias tarjetas de sonido en una superficie. Tan solo viendo las posiciones relativas de estas con la tarjeta de referencia, podemos concluir que la tarjeta con la trompeta sería la que reproduciría más bajito, mientras que la del bombo sería la que tendría un sonido más potente. El bombo se reproduciría con un ritmo mucho más rápido que el del tambor, que se reproduciría más despacio.

Para poder hacer realidad este funcionamiento hemos tenido que tomarnos un par de licencias para acometer los diversos problemas que surgen a la hora de plantear la lógica de esta aplicación. Veamos cómo se hacen los cálculos. Para calcular el tiempo que ha de pasar entre una reproducción del sonido y la siguiente, lo hacemos en función de la distancia. El problema es que no podemos utilizar la distancia como tal, ya que hemos visto que la distancia a la que podemos colocar una tarjeta de otra sin tener problemas con la detección de los targets, puede llegar a ser de unas 30 unidades de las que utiliza unity. No tiene sentido por tanto, que pasen hasta 30 segundos entre reproducción y reproducción. Lo que hacemos entonces es colocar un punto de referencia en el que el tiempo entre reproducción y reproducción es de 1 segundo. Llamemos a este punto "punto de segundo" para referirnos a él. La distancia a

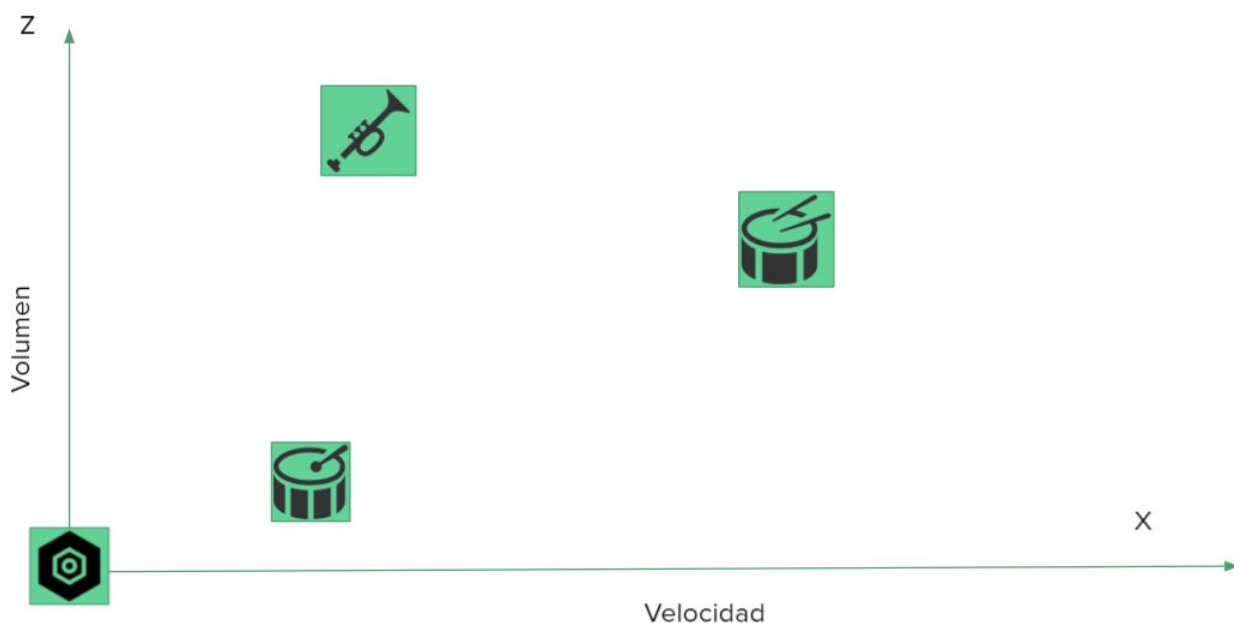


Figura 3: Ejemplo de distribución de tarjetas

la que se encuentra este punto de la tarjeta de referencia se ha obtenido de forma empírica con las pruebas que se han realizado. Si el objeto está entre la tarjeta de referencia y el punto de segundo, el sonido se reproducirá cada cierta fracción de segundo. Por el contrario, si la tarjeta se encuentra más allá del punto de segundo, el tiempo que pasará entre reproducción y reproducción será mayor de un segundo, concretamente de forma proporcional a la distancia.



Figura 4: Funcionamiento de la variación de velocidad

Por lo que hemos visto ahora, la función que determina el período de reproducción de un sonido con respecto a su distancia, sería:

$$f(x) = \text{abs}(Rx - Sx)/P$$

siendo Rx , la coordenada x de la tarjeta de referencia, Sx , la coordenada x de la tarjeta de sonido y P la distancia a la que colocamos el punto de segundo de la tarjeta de referencia. De esta forma, conseguimos que cuando una tarjeta está a la misma distancia en el eje x que el punto de segundo, el sonido que reproduce se repita cada segundo.

Sin embargo, cuando llevamos esto a la práctica, los resultados eran muy malos. La función que acabamos de proporcionar, como bien se puede ver se trata de una función continua, y con los valores que proporciona no podemos trabajar. Resultaba imposible encajar en un ritmo un sonido que se reproducía cada 1.01564 segundos y otro cada 0.5635 segundos. Lo interesante es quedarse con múltiplos de 0.5 para que sean sencillos de encajar visualmente,

a partir de las distancias. Para ello, lo que hacemos es multiplicar por dos el cociente que tenemos en la fórmula, utilizar una función de redondeo a la unidad, y dividirlo otra vez por la mitad. De esta forma nos quedamos con un período que siempre será múltiplo de 0.5 y encajar varios sonidos en un ritmo se hace muy sencillo. Si queremos que se escuchen dos golpes de tambor por cada golpe de bombo, solo tenemos que colocar la tarjeta del bombo más o menos al doble de distancia de la referencia de la que está la del tambor. La función final con la que sacamos el período de repetición de los sonidos sería:

$$f(x) = \text{round}(2 * \text{abs}(Rx - Sx/P))/2$$

Con el volumen hacemos algo más sencillo. Establecemos también una distancia de referencia pero esta vez en el eje Z, y lo que hacemos es que el volumen a esa distancia de referencia se vuelve 0. Cuanto más cerca esté la tarjeta de sonido, más fuerte se reproducirá el sonido. La función para el volumen sería:

$$f(x) = 1/((\text{abs}(Rz - Sz)/V)$$

siendo Rz la coordenada z de la tarjeta de referencia, Sz la coordenada z de la tarjeta de sonido y V la distancia que hemos establecido de forma empírica en la que pondremos el volumen del sonido a 0.

Con esto ya tenemos resuelta la lógica principal de nuestra aplicación de realidad aumentada,

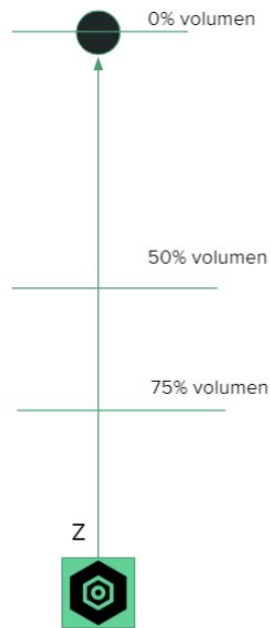


Figura 5: Funcionamiento de la variación de volumen

podemos crear ritmos sencillos, con varios sonidos, variar su frecuencia de repetición y su volumen.

5. Arte

5.1. Tarjetas

Se presentan a continuación las imágenes de los targets de realidad aumentada que utilizaremos en la aplicación.

5.1.1. Centro



Figura 6: Tarjeta de la referencia

5.1.2. Bombo



Figura 7: Tarjeta de sonido para el bombo

5.1.3. Tambor



Figura 8: Tarjeta de sonido para el tambor

5.1.4. Platillos



Figura 9: Tarjeta de sonido para los platillos

5.1.5. Bocina



Figura 10: Tarjeta de sonido para la bocina

5.1.6. Voz



Figura 11: Tarjeta de sonido para la voz

5.2. Modelos

Se presentan a continuación algunas capturas de los modelos creados para su utilización en la aplicación.

5.2.1. Centro

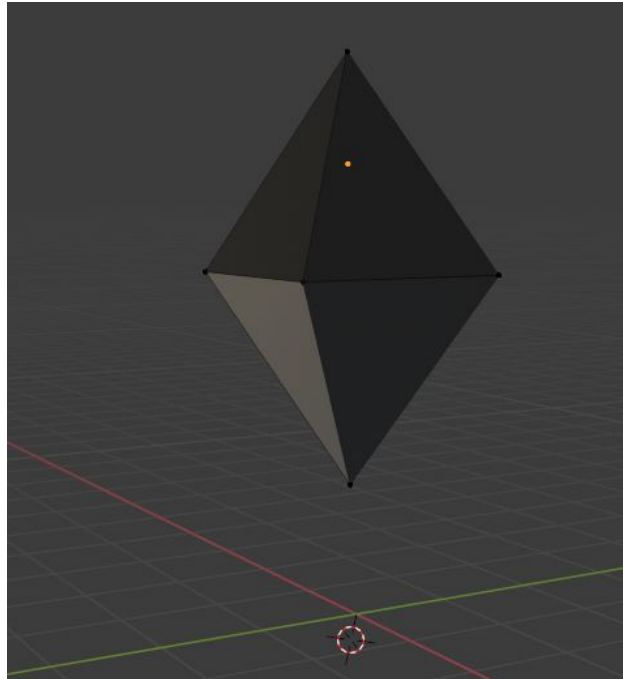


Figura 12: Modelo de la referencia

5.2.2. Bombo

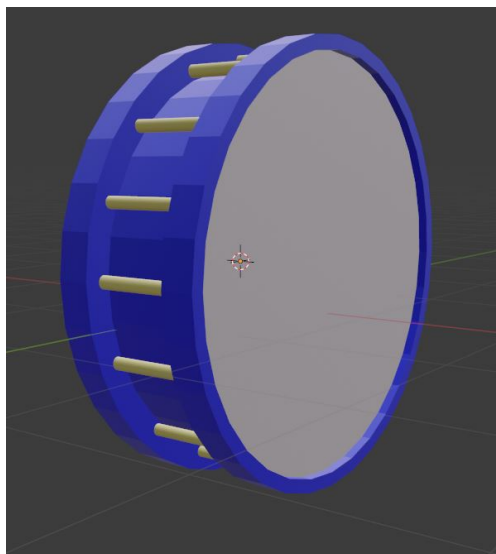


Figura 13: Modelo para el bombo

5.2.3. Tambor

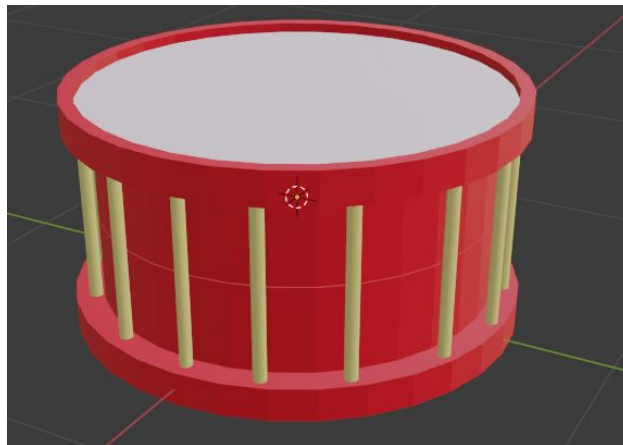


Figura 14: Modelo para el tambor

5.2.4. Platillos

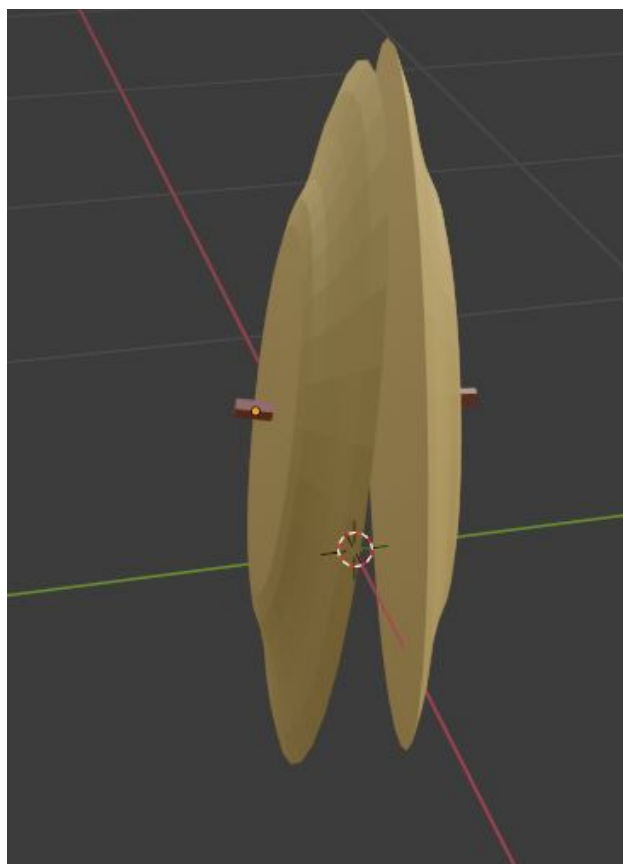


Figura 15: Modelo para los platillos

5.2.5. Coche

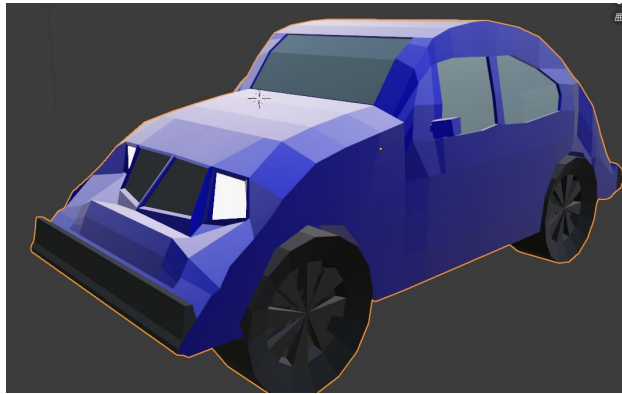


Figura 16: Modelo del coche creado en las clases de Blender

5.2.6. Personaje

Se ha añadido el modelo de un personaje sacado de Models Resource [7].



Figura 17: Modelo del personaje descargado

5.3. Menús

Se presentan a continuación unas capturas con los menús que integra la aplicación.

5.3.1. Menú principal



Figura 18: Menú principal

Se trata del menú inicial, el que nos encontraremos nada más abrir la aplicación. En la opción *Empezar* podremos comenzar a utilizar la aplicación de realidad aumentada, en *Salir* cerraremos la aplicación y *Cómo usar* y *Créditos* nos llevará a las demás pantallas que vemos a continuación.

5.3.2. Cómo usar



Figura 19: Cómo usar

En esta pantalla se explica brevemente cómo podemos comenzar a utilizar la aplicación de realidad aumentada.

Texto: Para empezar, recomendamos utilizar la aplicación en modo horizontal, ya que así tendrás más espacio en pantalla. Para comenzar a hacer ritmos, deberás colocar la tarjeta de referencia (la que tiene un hexágono) de tal forma que quede en la esquina inferior izquierda de la pantalla. Verás que aparece un diamante negro. Ahora coge alguna de las demás tarjetas y colócala a la derecha de la tarjeta de referencia. Con respecto al eje horizontal, cuanto más cerca la coloques, más rápido se repetirá el sonido. Y cuanto más lejos la pongas más lento se repetirá. Con respecto al eje vertical, cuanto más cerca esté la tarjeta del sonido de la de referencia, más fuerte sonará, y cuanto más lejos, más bajito lo hará.

5.3.3. Créditos



Figura 20: Créditos

En esta pantalla se da una breve información sobre quién es el autor de la aplicación y para qué se ha creado.

Texto: MusicAR es un proyecto creado por Jesús Serrano Gallán para la asignatura de Desarrollo de Sistemas Hipermedia. Con esta aplicación se pretende poner en práctica lo aprendido acerca de realidad aumentada y aplicarlo hacia una sencilla aplicación para poder hacer ritmos con distintos sonidos.

6. Conclusiones

Se da por finalizado el proyecto habiendo conseguido la funcionalidad esperada, haciendo un repaso por todo lo aprendido durante la asignatura y profundizando especialmente en la realidad aumentada y la API de sonidos de Unity. A continuación se hace un repaso de los problemas que se han encontrado durante el desarrollo y las posibles mejoras futuras que podrían proveerse a la aplicación.

6.1. Problemas encontrados

- **Gran carga de trabajo:** siendo el final del cuatrimestre y cursando 4 asignaturas más, el tiempo ha sido un gran problema. La aplicación se ha desarrollado principalmente en ratos libres y en el tiempo de clase de la propia asignatura. Gracias a haber trabajado solo, no he tenido que coordinarme con nadie y he podido desempeñar el trabajo cuando he podido. Eso sí, obviamente al trabajar solo la carga de trabajo es mayor.
- **Detección de los targets de realidad aumentada:** en un principio los targets los imprimí en papel y trabajar con ellos era imposible: se doblaban o se movían debido a su poco peso, etc. Tuve que recortar cuadrados de cartulina y pegarlos ahí. De esa forma conseguí que los targets estuvieran un poco más rígidos y se hacía más fácil trabajar con ellos. También hay problemas en cuanto a la distancia de detección. Normalmente hay que acercar los targets a la cámara para que se detecten correctamente antes de situarlos en la superficie donde queremos operar con ellos.
- **Relación período-distancia:** como ya se ha expuesto en la sección de Funcionamiento, es complejo relacionar la distancia con el período de repetición de los sonidos de forma que sean manipulables de una forma intuitiva. Al final se ha conseguido adaptar siempre los períodos a múltiplos de 0.5, lo que hace muy sencilla la tarea de combinar los sonidos de forma rítmica.
- **Exportación de modelos a Unity:** esto es sin duda una de las cosas que más problemas ha dado. No tiene sentido que sea tan complejo que un modelo se exporte correctamente desde Blender a Unity. De hecho, la aplicación tiene algunos modelos, como el del coche, con fallos fruto de la exportación tan problemática a Unity, que no se han conseguido solucionar.

6.2. Posibles mejoras futuras

- **Adaptación a gafas de realidad aumentada:** sería muy interesante utilizar esta aplicación con gafas de realidad aumentada como las HoloLens o algo así. De esa forma no tendríamos que estar pendientes a que las tarjetas se encuadren dentro de una pantalla, si no que habría correspondencia directa entre la aplicación y nuestra visión.
- **Interfaz para personalizar sonidos:** sería interesante poder ofrecer a los usuarios una manera de que subieran sus propios archivos mp3 para que se enlazaran con las tarjetas que ellos desearan. De esa forma haríamos que la aplicación pudiera adaptarse completamente a las necesidades y objetivos de los usuarios.

Referencias

- [1] Unity. URL <https://unity.com/es>.
- [2] Audacity, . URL <https://sourceforge.net/projects/audacity/>.
- [3] Unity audiosource api, . URL <https://docs.unity3d.com/ScriptReference/AudioSource.html>.
- [4] Blender. URL <https://www.blender.org/>.
- [5] Diagrams.net. URL <https://www.diagrams.net/>.
- [6] Iriun webcam. URL <https://iriun.com/>.
- [7] Models resouce. URL <https://www.models-resource.com/>.
- [8] Vuforia engine. URL <https://developer.vuforia.com/>.