

Guitar Duel **Profesionalizador (TFP)**

Autor: Carlos Peña Pérez
Tutor: Alberto Sánchez Amo
Profesor: Joan Arnedo Moreno

Grado de Multimedia
Videojuegos

Créditos/Copyright

CC BY-NC-ND 3.0 ES / Carlos Peña Pérez



Esta obra está sujeta a una licencia de Reconocimiento -NoComercial-SinObraDerivada [3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Guitar Duel</i>
Nombre del autor:	<i>Carlos Peña Pérez</i>
Nombre del colaborador/a docente :	Alberto Sánchez Amo
Nombre del PRA:	Joan Arnedo Moreno
Fecha de entrega (mm/aaaa):	<i>05/2022</i>
Titulación o programa:	<i>Grado de Multimedia</i>
Área del Trabajo Final:	<i>TFG Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Videojuego, guitarra, duelo</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>	
<p>El videojuego trata de recrear un clásico juego de mesa, llamado Simon, orientándolo ahora hacia un público objetivo concreto: músicos y guitarristas. En este el jugador trata de repetir una serie de sonidos emitidos de forma aleatoria. Mientras el jugador acierte, el dispositivo continuará emitiendo patrones de sonido cada vez más complicados.</p> <p>Guitar Duel tratará de realizar algo parecido incorporando la utilización de una guitarra real como interfaz de usuario. El juego emitirá una serie de patrones de sonido, dentro de una escala musical determinada, que el usuario deberá repetir instantáneamente con su guitarra. Además, a diferencia de otros juegos que puedan estar una línea parecida, como Guitar Hero o Rocksmith, Guitar Duel también tendrá una línea argumental que ayude al jugador a identificarse con un personaje haciendo que siga la trama hasta completarlo.</p> <p>Para la creación de este juego se necesita profundizar en el campo de la producción de videojuegos en Unity, así como, la programación y procesamiento digital de audio, DSP, en tiempo real. Para ello se requiere de alta dedicación en la investigación, especialmente de este último campo.</p> <p>Enlace al repositorio en GitHub (Se incluye en este un <i>building</i> ejecutable del juego): https://github.com/ignotocharly/GuitarDuel</p>	

Enlace al video explicativo en **Youtube**:

<https://youtu.be/TUmUbl3nRDo>

Enlace al trailer en **Youtube**:

<https://youtu.be/PZumMdvCmQo>

Enlace a la presentación **Youtube**:

<https://youtu.be/0x7OxxZjSxg>

Abstract (in English, 250 words or less):

The videogame pretends to recreate a classic board game called **Simon**. This time the game will be oriented to a target: musicians and guitarists. In this game, the player tries to replicate different randomized sound patterns. The device will continue to play increasingly complicated sound patterns while the player replays them right.

Guitar Duel will try to recreate something similar although using a **real guitar as the user interface**.

The game will play the sound patterns, based on musical scales, and the gamer will try to replay them instantaneously with his guitar. In addition to this, unlike other similar games like *Guitar Hero* or **Rocksmith**, also has a storyline that will make the player been stuck to the game to complete it.

It's necessary to investigate the fields of videogames production with *Unity* as well as audio and digital audio processing, DSP, in real-time. This requires a lot of time dedicated to research.

Dedicataria/Cita

A mi familia y amigos.

Índice

1. Introducción.....	8
1.1. Introducción/Prefacio	8
1.2. Descripción/Definición	9
1.3. Objetivos generales	11
1.3.1. Objetivos principales	11
1.4. Metodología y proceso de trabajo.....	12
1.5. Planificación	14
1.5.1. Mapa de Hitos	14
1.5.2. Desglose de tareas EDT	16
2. Análisis de mercado.....	18
2.1. Estudio de mercado.....	18
2.1.1. Juegos Rítmicos	19
2.1.2. Juegos de Memoria.....	22

2.1.3. Juegos de Lucha	24
2.2. Estado de la cuestión técnica	25
2.2.1. DSP. Tratamiento digital de audio en tiempo real.....	25
1. LibPDIntegration	26
2. JUCE.....	27
3. FMOD	27
4. Wwise	27
2.2.2. Detección de tono.....	27
3. Propuesta	30
3.1. Definición de objetivos/especificaciones del producto.....	30
3.2. Estrategia de marketing	30
3.3. Publicitación	31
3.4. Alojamiento en plataforma de videojuegos	31
4. Diseño	32
4.1. Arquitectura general de la aplicación/sistema/servicio.....	32
4.2. Arquitectura de la información y diagramas de navegación	32
4.3. Diseño gráfico e interfaces.....	33
4.3.1. Estilos	34
4.4. Generación de audio y sonidos	38
4.5. Lenguajes de programación y motor utilizado.....	40

Figuras y tablas

Índice de figuras

Figura 1: Guitarra eléctrica.....	8
Figura 2: Imagen del juego Rocksmith	9
Figura 3: Imagen de la película Crossroads(1986)	10
Figura 4: Procesado de audio analógico / digital.....	25
Figura 5: Menú de preferencias de audio de Unity	26
Figura 6: Tabla de frecuencias.....	28
Figura 7: Boceto de pantalla de inicio.	33
Figura 8: Boceto de pantalla de juego.....	33
Figura 9: Bocetos de logotipo	34
Figura 10: Boceto de pantalla de juego.....	35
Figura 11: Logotipo renderizado. Finalmente se optó por eliminar la calavera	35
Figura 12: Jerarquía de huesos del personaje enemigo	36
Figura 13: Personaje principal texturizado en Unity	37
Figura 14: Escenario generado en Unity	37
Figura 15: imagen de proyecto de Studio One	39
Figura 16: Ignite	39
Figura 17: Pantalla de portada	42
Figura 18: Pantalla de juego	43

Índice de tablas

Mesa 1: Mapa de hitos.....	¡Error! Marcador no definido.
----------------------------	-------------------------------

1. Introducción

1.1. Introducción/Prefacio

Dentro del amplio mundo del videojuego existen algunos campos que no se han explorado de manera efectiva y que abren un nicho de mercado que se podría considerar poco explotado.

La música es, además de una de las principales disciplinas artísticas, una de las formas de entretenimiento más consumida a lo largo de los tiempos y en la actualidad.

Cabe destacar que uno de los instrumentos musicales más populares, si no el que más, es la guitarra. Existe un amplio número de personas que tocan este instrumento de forma más o menos profesional, o quieren aprender a tocarlo. Así que, porque no aprender jugando. Siendo el mundo del videojuego un campo en auge, que bien podría considerarse como un arte multidisciplinar, para algunos el 8º arte, no es descabellado fusionar estas dos disciplinas para crear un producto de entretenimiento diferenciado.



Figura 1: Guitarra eléctrica.

1.2. Descripción/Definición

En la actualidad existen algunos videojuegos de género musical con un éxito bastante amplio, **Rock Band**, **Sing Star**, **Just Dance**, e incluso algunos orientados a la guitarra, como **Guitar Hero** o **Rocksmith**. Sin embargo, **Guitar Hero** o **Rock Band** no utilizan instrumentos reales, como si lo hace **Rocksmith**, y en cualquier caso la motivación principal de estos es simplemente tocar canciones de forma más o menos fiel a las originales

Por ello, el videojuego presentado, **Guitar Duel**, pretende traer a los jugadores entretenimiento, aprendizaje y desarrollo de sus capacidades. Por un lado, entrenará su oído, permitiéndoles reconocer los sonidos de la guitarra a lo largo del mástil, ayudará a aprender el conocimiento de las escalas musicales, así como, mejorará sus habilidades técnicas de interpretación.

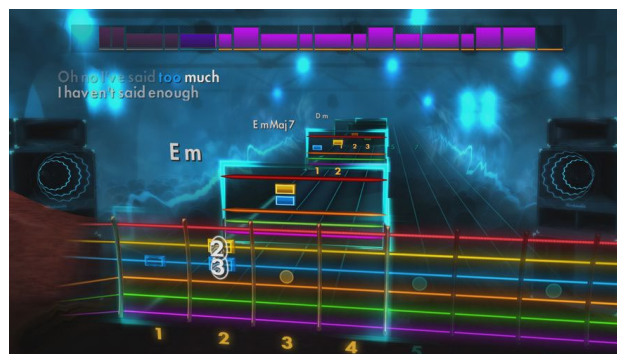


Figura 2: Imagen del juego **Rocksmith**

Guitar Duel cuenta además con una línea argumental que pretende satisfacer la parte más lúdica de los jugadores. Estos podrán meterse en la piel de un guitarrista enfrentándose a una serie de adversarios, al estilo **Street Fighter** o **Mortal Kombat**, que deberá eliminar demostrando su habilidad para replicar los riffs que estos reproduzcan. Véase como referencia una escena de la película **Crossroads (1984)** en la que ocurre un duelo de guitarra en el cual el protagonista, protagonizado por [Ralph Macchio se enfrenta al mismísimo Steve Vai](#).



Figura 3: Imagen de la película **Crossroads(1986)**.

El producto es, además, altamente escalable ya que podrían llegar a ofrecerse opciones de multijugador que permitieran realizar duelos entre guitarristas, incluso de forma online, aprovechando tecnologías como las empleadas por programas como [Jammr](#). En este caso el juego se beneficiaría de que ambos jugadores no tocan de manera simultánea con lo cual los problemas de latencia no afectarían.

1.3. Objetivos generales

Mediante la creación del presente **TFG** se pretende profundizar en los conocimientos adquiridos a lo largo de la realización del **Grado en Multimedia** para crear un producto nuevo. Además, servirá para investigar en la **programación de Audio y DSP**.

1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Cubrir un nicho de mercado poco explotado.
- Crear un producto novedoso.

Objetivos para el cliente/usuario:

- Mejorar sus habilidades técnicas.
- Mejorar el conocimiento de las escalas musicales.
- Aumentar su oído musical.
- Entretenimiento.

Objetivos personales del autor del TF:

- Investigar y aprender sobre la creación de videojuegos.
- Investigar y aprender técnicas de modelado y algoritmos de audio y DSP.
- Aunar disciplinas de 3D, creación de videojuegos y musicales.

1.4. Metodología y proceso de trabajo

Entre las diferentes estrategias para llevar a cabo el trabajo se ha optado por la adaptación de un producto o una serie de productos ya existentes para crear algo diferente. En el fondo se trata de integrar una serie de funcionalidades que existen en diferentes productos para poder generar algo nuevo:

- Se reutilizará la idea de generación y repetición de patrones aleatorios de **Simon** aunque manteniéndolos dentro de escalas musicales concretas.
- La idea de combate de juegos de lucha, como **Mortal Kombat**, será aprovechada al menos como referente estético y argumental.
- La captación de sonido y procesamiento digital, **DSP**, en tiempo real utilizada en juegos como **Rocksmith** permitirá al jugador disfrutar de una experiencia lúdica mientras toca la guitarra.

En cuanto a la parte técnica. Se utiliza **Unity** como motor del videojuego. No obstante, la elaboración de este juego enfrenta ciertas dificultades a nivel técnico puesto que el procesamiento en tiempo real del audio emitido por la guitarra requiere de dos algoritmos principales que podría resultar difícil implementar:

- Reconocimiento de la nota tocada en la guitarra para poder compararla con las notas emitidas por el juego.
- **DSP**. Procesamiento y reproducción del sonido en tiempo real procurando que la jugabilidad y la experiencia del jugador sea lo más agradable y divertida posible.

Teniendo esto en cuenta, ya se ha realizado un estudio de posibles soluciones a este problema que recurriría a software de terceros para realizarlo:

- [Pure Data](#). Este software de programación de audio, abierto y gratuito, puede implementar dichos algoritmos. Su librería [LibPdIntegration](#), permitiría añadir el *patch* creado como *plugin* de **Unity**.
- [JUCE](#). Se trata de un software de creación de *plugins* de procesamiento de audio digital que permite [conexión con Unity](#). Es de uso completamente libre para estudiantes y creadores independientes de bajos ingresos.
- [Pitch Detector](#). Aunque existe una motivación principal para hacer el desarrollo de cada una de las partes del videojuego, por motivos de limitación de tiempo, también podría utilizarse alguna extensión de **Unity**, como **Pitch Detector**, que facilitaría esa parte del trabajo.

Así pues, teniendo en cuenta las debilidades del proyecto se tratará de resolver dichas dificultades técnicas lo antes posible para poder afrontar la resolución del proyecto con expectativas de éxito. Posteriormente podrá avanzarse en el prototipado del videojuego para lo que se pretende elaborar un nivel con dos personajes enfrentándose.

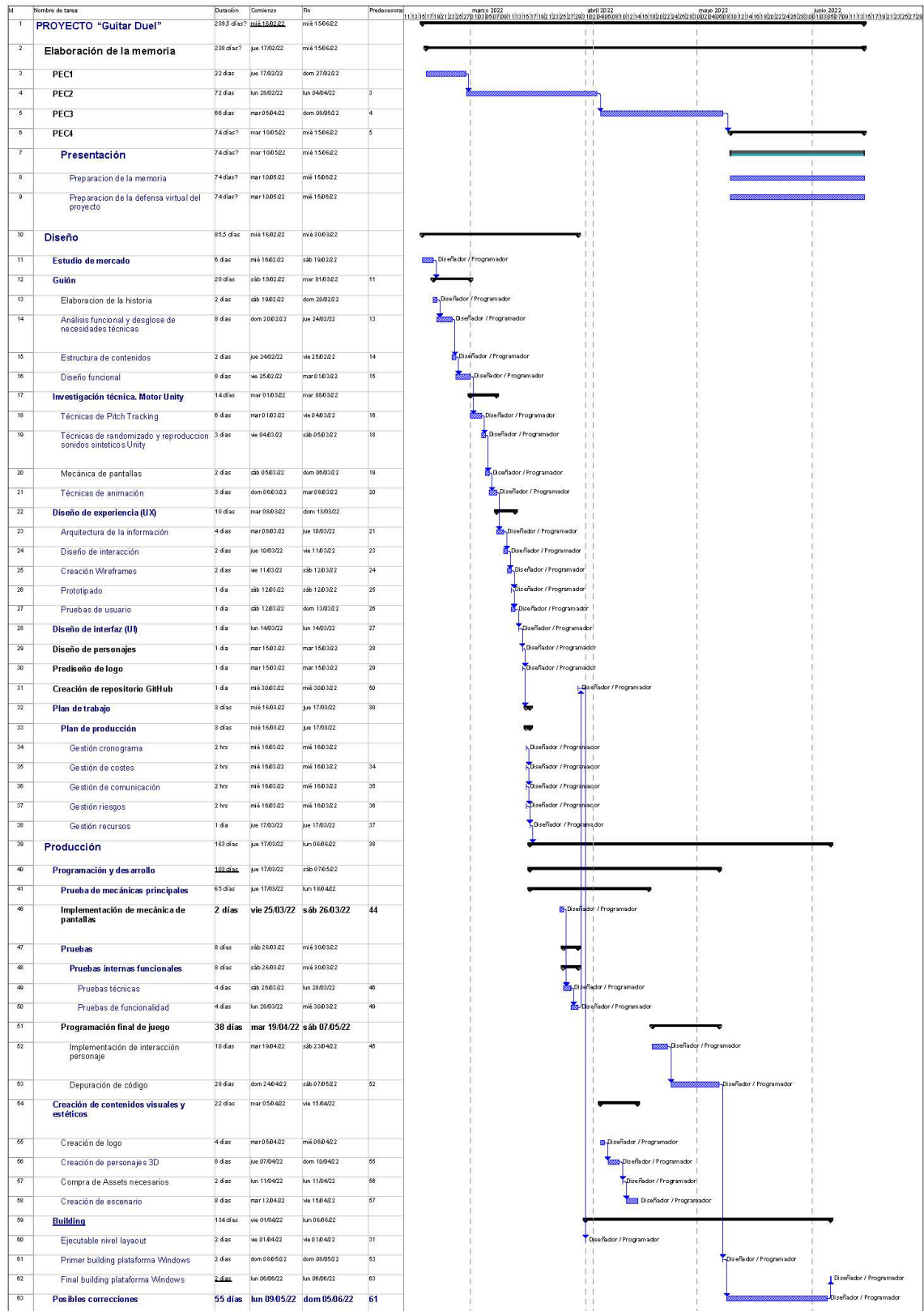
1.5. Planificación

1.5.1. Mapa de Hitos

Nombre	Comienzo	Fin
PROYECTO <i>“Guitar Duel”</i>	mié 16/02/22	mié 15/06/22
Elaboración de la memoria	jue 17/02/22	mié 15/06/22
PEC1	jue 17/02/22	dom 27/02/22
PEC2	lun 28/02/22	lun 04/04/22
PEC3	mar 05/04/22	dom 08/05/22
PEC4	mar 10/05/22	mié 15/06/22
Diseño	mié 16/02/22	vie 15/04/22
Estudio de mercado	mié 16/02/22	sáb 19/02/22
Guion	sáb 19/02/22	jue 03/03/22

Investigación técnica. Motor <i>Unity</i>	jue 03/03/22	mar 22/03/22
Diseño de experiencia (UX)	mar 22/03/22	vie 15/04/22
Creación de repositorio GitHub	lun 28/03/22	lun 28/03/22
Diseño de interfaz (UI)	mar 29/03/22	mar 29/03/22
Diseño de personajes	mié 30/03/22	mié 30/03/22
Diseño de logo	jue 31/03/22	jue 31/03/22
Plan de trabajo	vie 01/04/22	dom 03/04/22
Producción	mié 23/03/22	lun 06/06/22
Programación y desarrollo	mié 23/03/22	mié 11/05/22
Creación de contenidos visuales y estéticos	mar 05/04/22	mar 19/04/22
<i>Building</i>	lun 04/04/22	dom 05/06/22
Posibles correcciones	mar 10/05/22	dom 05/06/22

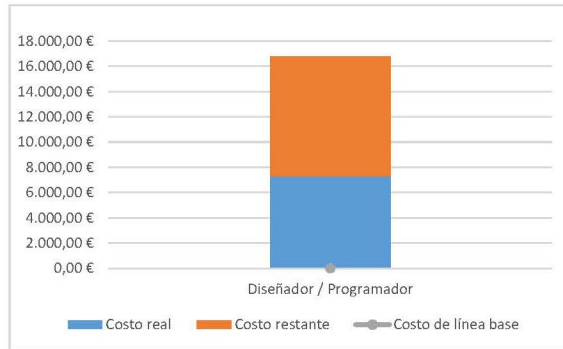
1.5.2. Desglose de tareas EDT



VISIÓN GENERAL DE COSTO DE RECURSOS

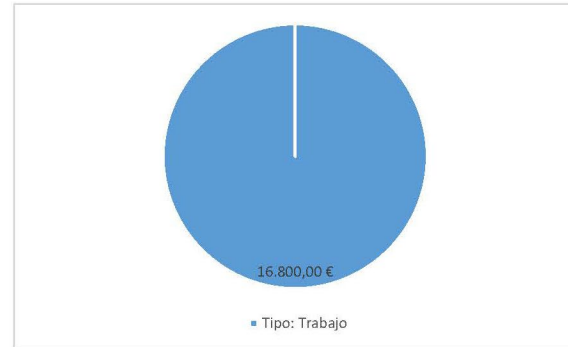
ESTADO DEL COSTO

Estado de costo de los recursos de trabajo.



DISTRIBUCIÓN DE COSTOS

Cómo los costos están distribuidos entre tipos de recursos diferentes.



DETALLES DE COSTOS

Detalles de costos de todos los recursos de trabajo.

Nombre	Trabajo real	Costo real	Tasa estándar
Diseñador / Programador	364 hrs	7.280,00 €	20,00 €/hr

2. Análisis de mercado


Tratándose este de un videojuego que aúna diferentes géneros y requiere de técnicas no demasiado desarrolladas en este ámbito, como pueda ser el procesamiento de una entrada de audio en tiempo real, parece justificado establecer algunos apartados que diferencien el análisis de mercado del estado de la cuestión que estudiará como implementar dichas técnicas.

2.1. Estudio de mercado

Guitar Duel es un videojuego atípico. Reúne algunas características propias los llamados juegos rítmicos, de los juegos de memoria, o incluso de los juegos de lucha, al menos en lo que a estética se refiere.


A continuación, se expone un resumen de las diferentes opciones que se encuentran en el mercado en cada ámbito incluyendo fichas dentro de las cuales se trata de extraer, en el apartado de utilidades, mecánicas o técnicas implementadas en *Guitar Duel*.

2.1.1. Juegos Rítmicos


Nombre	Guitar Hero	
Plataformas	PlayStation 2, PlayStation 3, Xbox 360, Wii, Windows, Mac	
Web	Guitar Hero está suspendido	
Descripción	<p>Es precursor de este tipo de videojuegos. Consiste en pulsar, en un tiempo concreto, una o varias de las 5 teclas de una interfaz, con forma de guitarra, que responden a ciertas notas musicales de la canción. Contra más notas musicales son acertadas durante la reproducción de la canción mayor es la puntuación obtenida.</p> <p>Guitar Hero consta de numerosas versiones llegando hasta la versión Live, en <i>streaming</i> con la plataforma GHTV, en la que incorporó una historia que consiste en ir tocando conciertos hasta llegar a llenar estadios.</p> <p>Actualmente, tras haber dedicado títulos a grandes bandas del rock como Metallica, Van Halen o Aerosmith, Guitar Hero ha quedado descatalogado, al menos de manera suspendida.</p>	
Utilidades	<p>El modo en que se representan las notas, con una relación entre tiempo y espacio, ayudando al jugador a identificar el momento en que ha de pulsar la nota correspondiente. Resulta inspirador y queda relacionado a como el personaje correspondiente al computador emite algunas “magias” que lanza contra el del jugador. Si el jugador pulsa la nota en el momento adecuado dicha “magia” no afectará a su porcentaje de vida, haciendo perder a la computadora 20 puntos sobre 100. De lo contrario, será el jugador el que los pierda.</p>	


Mesa 2: Guitar Hero

Nombre	Rockband	
Plataformas	PlayStation 2, PlayStation 3, Wii, Xbox 360, PSP, Nintendo DS	
Web	https://www.rockband4.com/	
Descripción	Rockband copia el formato de Guitar Hero, aunque realmente fue desarrollado por Harmonix creadores del primer título de la saga, solo que esta vez lo lleva al formato multijugador incorporando bajo batería y voz al juego.	
Utilidades	En Rockband se permite el enfrentamiento entre equipos en el modo Rivals de forma online. Es una idea que daría una nueva dimensión al proyecto, aunque no sea desarrollada actualmente.	


Nombre	Rocksmith+	
Plataformas	PS5, PS4, Xbox Series X S, Xbox One y PC.	
Web	https://www.ubisoft.com/es-es/game/rocksmith/plus	
Descripción	<p>Rocksmith+ resulta de una evolución claramente basada en Guitar Hero sólo que utilizando una guitarra real como interfaz de usuario. Por ello, fueron incluso acusados por violar la patente de Guitar Hero. Legalidades aparte, este juego trae como novedades que el jugador escucha su guitarra en tiempo real con una serie de efectos aplicados y utiliza todos los trastes y cuerdas de la guitarra frente a los 5 botones de Guitar Hero.</p>	
Utilidades	<p>Ya que el sistema utilizado para la muestra de las notas es, a efectos prácticos igual que el de Guitar Hero lo que resulta de aquí interesante es el sistema de detección de notas y el procesado del audio en tiempo real. Si bien no es perfecto puesto que se aprecia cierta latencia, retraso entre la nota tocada y la escuchada tras su procesamiento, esta no afecta demasiado a la experiencia del usuario. La detección de notas resulta correcta, aunque se debe hacer notar que cuando la cadencia entre notas es corta el algoritmo se vuelve menos exigente con la corrección de estas, en pro de la jugabilidad probablemente.</p>	

2.1.2. Juegos de Memoria

Nombre	<i>Simon</i>	
Plataformas	<i>Online</i> , juego de mesa	
Web	Juegos Hasbro - Simon	
Descripción	<p>Simon es un juego electrónico que ya tiene 44 años y continúa en producción, ya que fue lanzado en 1978. El sistema reproduce una serie de sonidos, asociados a un botón de color que se retroilumina, el jugador tiene que repetirlos en el mismo orden pulsando los botones.</p>	
Utilidades	<p>La forma aleatoria en la que se emiten las notas, y se van sumando al patrón a repetir, resulta muy adictiva e incrementa la jugabilidad y la adherencia al juego. Por ello, se decide optar por esta mecánica para el juego. Además, las notas que la computadora toca aparecen representadas cuando son reproducidas en un mástil en mitad de la pantalla ayudando al jugador a encontrarlas en el mástil.</p>	

Nombre	Entrena tu Cerebro - Juegos de Memoria	
Plataformas	Android	
Web	Google Play - Entrena tu cerebro	
Descripción	Este juego para teléfonos inteligentes consta de una serie de juegos orientados al desarrollo de la memoria del jugador. Entre ellos se puede encontrar una implementación digital de Simon , esta vez con 6 botones.	
Utilidades	La forma aleatoria en la que se emiten las notas, y se van sumando al patrón a repetir, resulta muy adictiva e incrementa la jugabilidad y la adherencia al juego. Por ello, se decide optar por esta mecánica para el juego. Además, las notas que la computadora toca aparecen representadas cuando son reproducidas en un mástil en mitad de la pantalla ayudando al jugador a encontrarlas en el mástil.	

2.1.3. Juegos de Lucha

<p>Nombre</p>	<p>Street Fighter</p>	
<p>Plataformas</p>	<p>Arcade, NES, Master System, PC, Super NES, Sega Megadrive, GameBoy, GameBoy Color, GameBoy Advance, Sega Saturn, Sega Dreamcast, PSX, PS2, PS3, PS4, PSP, PSVita, Xbox, Xbox 360, Xbox One, Nintendo 3DS, Nintendo Switch</p>	
<p>Descripción</p>	<p>Se trata de un clásico de los videojuegos y fue lanzado hace ya 25 años. Este juego, que ha trascendido a prácticamente todas las plataformas de videojuegos, consiste en el enfrentamiento directo entre dos personajes que tratarán de golpearse hasta que la barra de vida de uno de los dos es consumida.</p>	
<p>Utilidades</p>	<p>Realmente este tipo de juegos influye en Guitar Duel en el aspecto más estético y de narrativa de la historia.</p> <p>Los personajes también disponen de una barra de vida que disminuye según son golpeados por una “magia”, al estilo <i>Hadouken</i>, que se lanzará al emitir una nota. Como aquí, el personaje principal de Guitar Duel irá enfrentándose a diferentes personajes que utilizarán diferentes escalas musicales diferenciándose del resto de enemigos.</p>	

2.2. Estado de la cuestión técnica

Como ya se ha comentado existen una serie de problemas a resolver en este tipo de videojuegos que merece la pena estudiar de forma más pormenorizada para entender el estado de la cuestión.

2.2.1. DSP. Tratamiento digital de audio en tiempo real

Cuando se quiere reproducir una fuente de audio a través del motor de un software, en este caso un motor de videojuegos como *Unity*, hay que abordar un problema importante como lo es la latencia.

La latencia es el tiempo transcurrido entre que la interfaz de audio capta el sonido hasta que vuelve a ser emitido a través de la salida de esta y llega al usuario. Durante todo ese tiempo se realizan una serie de procesos como la conversión de analógico a digital (**Analogic Digital Converter o ADC**), realizada por la interfaz, así como el procesamiento de esa señal digital con un tratamiento específico, por la CPU, y la vuelta a convertirse la señal de digital a analógico (**Digital Analogic Converter o DAC**), nuevamente por la interfaz, para ser amplificada y posteriormente emitida por los altavoces.

flujo

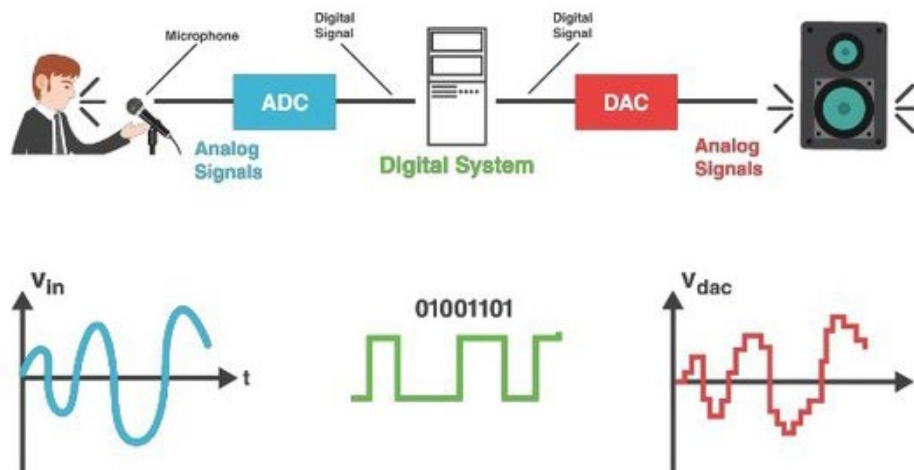


Figura 4: Procesado de audio analógico / digital.

De los procesos citados, el mayor problema se encuentra en el procesamiento en la **CPU** mediante el software utilizado. Por ejemplo, en *Unity 3D* se puede añadir distorsión o reverberación a la señal para obtener un sonido concreto deseado, pero este procesamiento requiere de un tiempo determinado el cual hace que sea escuchado por el usuario con cierto retardo.

Actualmente, **Unity** ofrece algunas opciones para reducir la latencia lo máximo posible. Sin embargo, la latencia obtenida es notablemente mejorable y puede resultar molesta para este tipo de videojuegos ya que, aunque se utilice una interfaz de audio adecuada, **Unity** no es capaz de explotar los drivers **ASIO** que este tipo de tarjetas disponen.

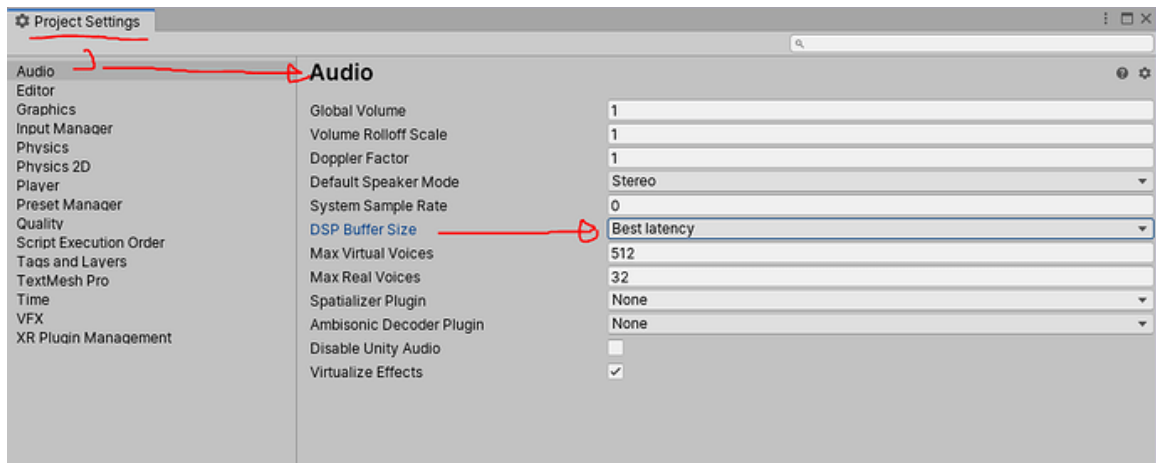


Figura 5: Menú de preferencias de audio de **Unity**.

Aunque el resultado obtenido por el momento resulta aceptable se han estudiado diferentes formas para superar este problema y se detallan a continuación:

1. LibPDIntegration

<https://github.com/LibPdIntegration/LibPdIntegration>

Esta librería permite la utilización del software de tratamiento y generación sintética de audio, *open source*, **Pure Data** que si utiliza drivers ASIO para reducir la latencia.

Pese a que se ha desarrollado un *patch*, como así se llaman los proyectos de **Pure Data**, que permite la escucha de la entrada de audio en tiempo real y se ha utilizado la librería para integrarlo en **Unity** el resultado no ha sido satisfactorio. El problema es que dicha librería no sustituye realmente el motor de audio de **Unity** y es necesario utilizar un **Audio Source**, de **Unity** al cual se asocia un objeto **Microphone** como fuente de clip de audio. Es decir, la captura de audio sigue realizándose por **Unity** enviando después a procesar a **Pure Data** y volviendo a **Unity** tras ser procesado. Además, en este caso es posible seleccionar los drivers **ASIO** en **Pure Data**.

2. JUCE

<https://juce.com/>

Este software permite crear aplicaciones orientadas al audio o incluso *plugins* que se puedan integrar en otros softwares, por ejemplo, **DAWs (Digital Audio Workstations)** o motores de juegos como **Unity**.

Al importarse como plugin cualquier aplicación desarrollada en JUCE para importar en *Unity* es utilizada a través del motor de su motor de audio por lo que no existe mejora en la latencia y experiencia de juego.

3. FMOD

<https://www.fmod.com/>

Se trata de un motor de audio que sustituye al motor de **Unity** el cual si permite el aprovechamiento de drivers **ASIO**. Si bien esto es cierto no se ha conseguido, por el momento, conectar la entrada de audio con para poder ser tratada.

4. Wwise

<https://www.audiokinetic.com/en/products/wwise>

Otro motor de audio que permite integración en **Unity**, o **Unreal**, es **Wwise**. Este motor empleado por videojuegos como **Rocksmith**, como puede leerse en el reportaje de este [link](#) parece ser la opción ideal para utilizar en un futuro desarrollo. No obstante, requiere de una curva de aprendizaje pronunciada y no ha podido ser utilizado en esta versión.

2.2.2. Detección de tono

El otro problema que se encuentra en este tipo de juegos es la detección de tono correspondiente a la cuerda y la altura del traste pulsado. Sin embargo, **Unity** provee de herramientas que si permiten el análisis espectral del audio con una latencia reducida permitiendo la implementación de un algoritmo eficiente para la tarea.

El método **GetSpectrumData()**, del objeto **Audio Source**, permite obtener una análisis realizar un análisis de espectro utilizando el algoritmo de **Transformación Rápida de Fourier (FFT)**. Dicho algoritmo descompone la señal en una serie de frecuencias que permitirá encontrar la frecuencia fundamental que se traduce en una nota concreta y se comparará con la nota emitida por el computador.

En el caso concreto se han estudiado algunos casos para poder desarrollar el código e incluso investigado algunos **assets** de **Unity** que permitieran realizar la tarea en cuestión en caso de no lograr ninguna solución sin la ayuda de terceros.

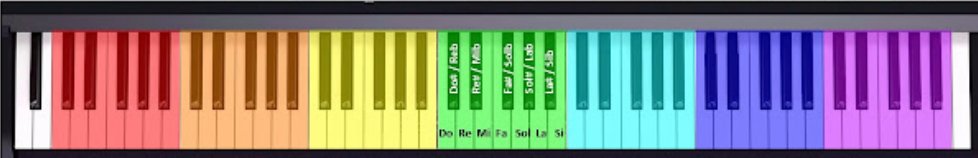
LASP

<https://github.com/keijiro/Lasp>

Se trata de un repositorio de GitHub donde se implementa una aplicación para **Unity** que utiliza el análisis de espectral para hacer una representación gráfica del audio captado por la entrada de micrófono.

Dicho repositorio sirvió de gran utilidad para entender como hallar la frecuencia fundamental utilizando **C#** en **Unity**. A partir de ahí sólo hubo que deducir como encontrar la nota correspondiente a dicha frecuencia. Para lo cual basta con seguir una tabla de correspondencias de nota <=> frecuencia.

FRECUENCIA DE LAS NOTAS MUSICALES EN HERCIOS (Hz)									
	OCTAVA 0	OCTAVA 1	OCTAVA 2	OCTAVA 3	OCTAVA 4	OCTAVA 5	OCTAVA 6	OCTAVA 7	OCTAVA 8
Do	16,3516	32,7032	65,4064	130,813	261,626	523,251	1046,50	2093,00	4186,01
Do# / Reb	17,3239	34,6479	69,2957	138,591	277,183	554,365	1108,73	2217,46	4434,92
Re	18,3540	36,7081	73,4162	146,832	293,665	587,330	1174,66	2349,32	4698,64
Re# / Mib	19,4454	38,8909	77,7817	155,563	311,127	622,254	1244,51	2489,02	4978,04
Mi	20,6017	41,2035	82,4069	164,814	329,628	659,255	1318,51	2637,02	5274,04
Fa	21,8268	43,6536	87,3071	174,614	349,228	698,456	1396,91	2793,83	5587,66
Fa# / Solb	23,1246	46,2493	92,4986	184,997	369,994	739,989	1479,98	2959,96	5919,92
Sol	24,4997	48,9995	97,9989	195,998	391,995	783,991	1567,98	3135,96	6271,92
Sol# / Lab	25,9565	51,9130	103,826	207,652	415,305	830,609	1661,22	3322,44	6644,88
La	27,5000	55,0000	110,000	220,000	440,000	880,000	1760,00	3520,00	7040,00
La# / Sib	29,1353	58,2705	116,541	233,082	466,164	932,328	1864,66	3729,31	7458,62
Si	30,8677	61,7354	123,471	246,942	493,883	987,767	1975,53	3951,07	7902,14



OCTAVA 1 OCTAVA 2 OCTAVA 3 OCTAVA 4 OCTAVA 5 OCTAVA 6 OCTAVA 7

Figura 6: [Tabla de frecuencias](#)

PITCH DETECTOR

<https://assetstore.Unity.com/packages/tools/audio/pitch-detector-19864>

Se trata de un *asset* de *Unity* 3D que realiza la detección del tono facilitando la tarea sin demasiada dedicación. Se planteó su utilización, pero finalmente no ha sido necesario.

HUMAN VOICE PITCH DETECTOR

<https://assetstore.Unity.com/packages/tools/audio/human-voice-pitch-detector-109019>

Al igual que el anterior este *asset* facilita la detección de notas siempre y cuando su frecuencia fundamental se encuentre entre 50 y 1000 Hz, lo cual resulta insuficiente para algunas notas de guitarra.

3. Propuesta

Guitar Duel es un videojuego que pretende desarrollar las habilidades de los guitarristas a nivel de conocimiento del mástil, localizando la altura y sonido de las notas, interiorizar las escalas, y todo mientras se divierten.

Existen diferencias de forma y estilo frente a otras opciones de la competencia que permiten que este sea un juego único. El hecho de no saber cuál será la próxima nota que elegirá la computadora, así como memorizar el orden en que han ido apareciendo las notas anteriores, mientras las relaciona con su posición en el mástil, hacen que este pueda resultar un juego sumamente atractivo y adictivo.

En el siguiente link puede verse una muestra de su funcionamiento, a nivel de layout.

[Guitar Duel - Video explicativo](#)

3.1. Definición de objetivos/especificaciones del producto

En la fase actual el juego sólo se desarrollará un nivel, un enemigo, correspondiente a una escala musical completa en una octava. Se trata de la escala de Do Mayor tocada a partir del 5 traste en la tercera cuerda.

La computadora lanza notas aleatorias que van sumándose a las ya emitidas, dentro de dicha escala, lo que ayuda al jugador a interiorizar y relacionar el sonido de estas con su posición en el mástil.

Dichas notas emitidas por la computadora son lanzadas contra el jugador a modo de magias, justo un compás después de ser emitidas, cuatro tiempos, el jugador deberá reproducir la nota con su guitarra.

3.2. Estrategia de marketing

Para conseguir hacer llegar el juego al mayor número de usuarios posible cabe destacar algunos pasos mínimos a seguir.

3.3. Publicitación

Siendo **Guitar Duel** un juego con un público objetivo tan específico tiene sentido enfocar una campaña de promoción hacia el sector de guitarristas. Por lo tanto, establecer algún tipo de contacto con páginas web como Guitarristas.info o foros como Sevenstring, con una amplia comunidad de usuarios, sería una buena forma de publicitar el juego.

Otra forma de publicitación es la realización de un spot con una atractiva factura visual para distribución por **RRSS** y plataformas como **Youtube**. Incluso contactar con *influencers* orientados al público de guitarristas para que realizaran una reseña podría ser una forma de que por su parte ofrezcan un contenido diferente a su contenido habitual haciendo llegar el juego a un público más amplio.

3.4. Alojamiento en plataforma de videojuegos

Para acceder a un videojuego de ordenador a día de hoy una de las plataformas más extendidas es Steam. Dicha plataforma ofrece servicios de alojamiento y una distribución del juego a nivel mundial en unos pocos pasos.

No obstante, hay que tener en cuenta que **Steam** requiere de un pago por anticipado de **100\$** por cada producto que se quiera vender a través de su plataforma. Además, un porcentaje pactado de comisión sobre los ingresos generados por el juego será absorbido como forma de pago por los servicios ofrecidos, este puede ser de hasta un 30%.

Sin embargo, cabe tener en cuenta otras plataformas como **Epic Games** o **Itch.io** que pese a no ser tan conocidas ofrecen condiciones más favorables para desarrolladores que empiezan o poco conocidos. En el caso de **Itch.io** la comisión por sus servicios es de sólo un 10% sin ningún tipo de pago inicial. Incluso permite la distribución y descarga gratuita del videojuego, algo muy interesante para nuevos desarrolladores independientes que pretenden dar a conocer sus productos.

Por todo esto Itch.io parece la opción más adecuada para distribuir **Guitar Duel** una vez desarrollada la versión final.

4. Diseño

4.1. Arquitectura general de la aplicación/sistema/servicio

La estructura organizativa de pantallas que componen **Guitar Duel** no requiere de mayor complejidad. Se trata únicamente de tres pantallas:

- Una pantalla de portada.
- Acceso a las preferencias técnicas del juego, que se incluye como pestaña accesible en la pantalla de portada.
- Pantalla de juego donde se desarrolla la acción principal.
- Pantalla de *Game Over*, superpuesta a la pantalla de juego tras la finalización de este.

En una realización del juego más completa, con más niveles y personajes, se deberían incluir otras pantallas como por ejemplo cinemáticas que ayudaran a relatar una historia interna en el videojuego.

4.2. Arquitectura de la información y diagramas de navegación

Se incluye a continuación los diagramas creados para la arquitectura de navegación del juego.



4.3. Diseño gráfico e interfaces

En cuanto al diseño de pantallas se han realizado algunos bocetos, a un nivel muy básico, sobre los que se trabajó parte de la programación del videojuego. Aquello que llamaríamos las mecánicas principales, o *core*, del videojuego.

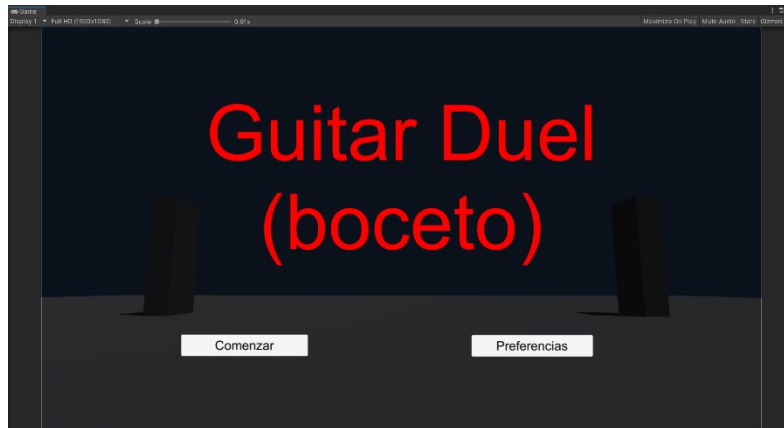


Figura 7: Boceto de pantalla de inicio.

Dichas versiones a nivel boceto se realizan con materiales de trabajo a un nivel básico que permitan trabajar sobre el videojuego sin invertir un tiempo de trabajo, que puede ser aplicado posteriormente, permitiendo concentrar los esfuerzos en el desarrollo.

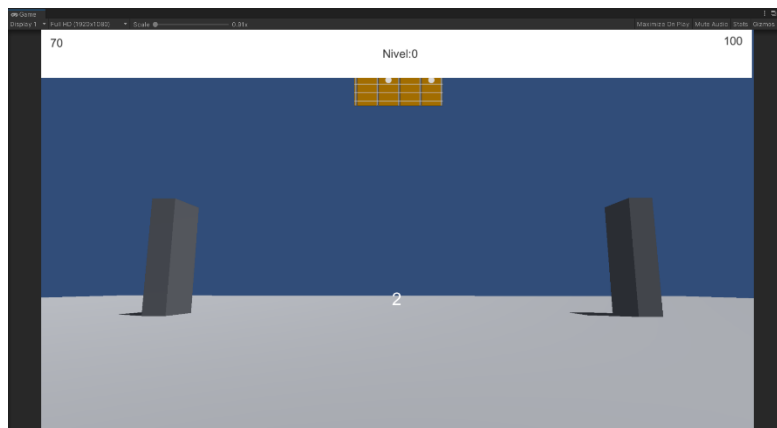


Figura 8: Boceto de pantalla de juego.

Tras haber desarrollado esa parte básica y central de programación y mecánicas es el momento de ejercer las labores de diseño necesarias para que el videojuego sea algo atractivo para los jugadores.

4.3.1. Estilos

Puesto que el videojuego tiene un público objetivo definido, la comunidad de guitarristas, y dentro de este los orientados a la guitarra eléctrica, se ha optado por una estética bastante común a todos ellos como lo es la estética del Rock.

Diseño de logotipo

Antes de realizar un diseño en digital se realizó un boceto a lápiz digital para ver encontrar que tipo de estética se podría buscar. En el caso del creador de este el mundo del **Rock** está bastante interiorizado por lo que la búsqueda de referencias visuales no fue muy necesaria en este punto salvo el revisar la parte referente a videojuegos que ya se citan anteriormente: **Guitar Duel**, **Rocksmith**, **Guitar Band**...



Figura 9: Bocetos de logotipo

Tras este primer diseño inicial el paso siguiente fue la búsqueda de una tipografía adecuada que pueda funcionar bien con el prediseño presentado. En ese sentido se opta por la tipografía **Altrashed**, con licencia comercial totalmente libre para uso personal y comercial. Sin embargo, hubo que modificarla con fines estéticos y técnicos para poder extruir sus caracteres perfectamente en 3D.

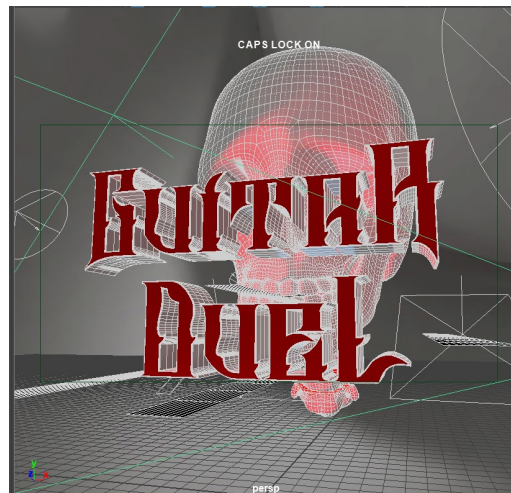


Ilustración 1

Figura 10: Boceto de pantalla de juego.

Además, en este tipo de juegos de lucha suelen utilizarse colores de alto contraste visual que hacen de sus imágenes algo llamativo. Por ello la utilización del rojo, como color principal, y el azul, como secundario, en el logotipo.

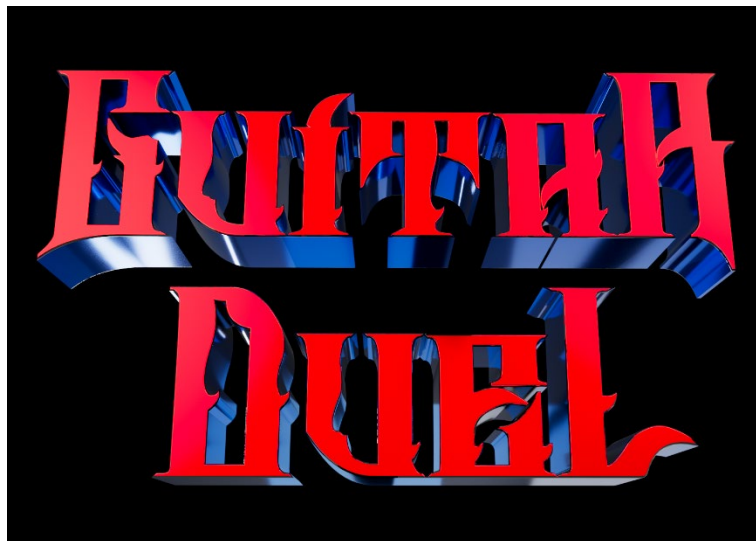


Figura 11: Logotipo renderizado. Finalmente se optó por eliminar la calavera y guitarras con objetivo de mejorar la visibilidad

Diseño de personajes

En cuanto al diseño de personajes se optó por la utilización de **Adobe Fuse**. Aunque se trata de un software ya abandonado por **Adobe** continúa siendo de gran utilidad para crear personajes de una forma rápida y con una topología adecuada para su animación.

Tras su diseño se utilizó **Mixamo** para extraer una animación de pose básica que hubo que modificar para adecuarla a los objetivos deseados. Para este cometido se utiliza el software **Autodesk Maya** en el cual además se realizaron el resto de clips de animación necesarios.

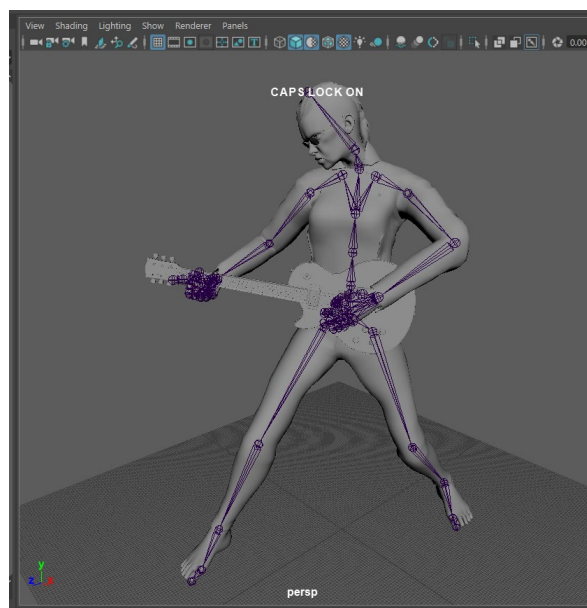


Figura 12: Jerarquía de huesos del personaje enemigo

Tras su diseño se utilizó **Mixamo** para extraer una animación de pose básica que hubo que modificar para adecuarla a los objetivos deseados. Para este cometido se utiliza el software **Autodesk Maya** en el cual además se realizaron el resto de clips de animación necesarios. Además, se añadió un modelo de guitarra básico que se utilizó en ambos personajes cambiando el color de sus materiales.

Tras estas labores los objetos **3D** son importados en **Unity** donde se aplican los materiales necesarios utilizando las texturas generadas por **Fuse**.

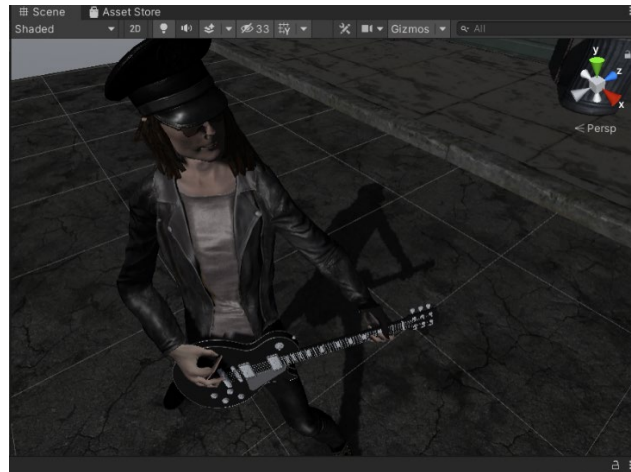


Figura 13: Personaje principal texturizado en **Unity**.

Diseño de escenario

Respecto al escenario se ha utilizado un pack 3D adquirido a través de la **Asset Store** de **Unity**, se trata de [Modular City Alley Pack](#). Dicho pack otorga derechos de libre uso sobre cualquiera de los recursos contenidos y su utilización facilita grandemente la generación de escenarios de calle. Estos recursos resultaban muy adecuados al tipo de estética que se quería mantener, de barrios bajos, calles sucias, tan encontrados tanto en juegos de lucha callejera como videoclips de rock.

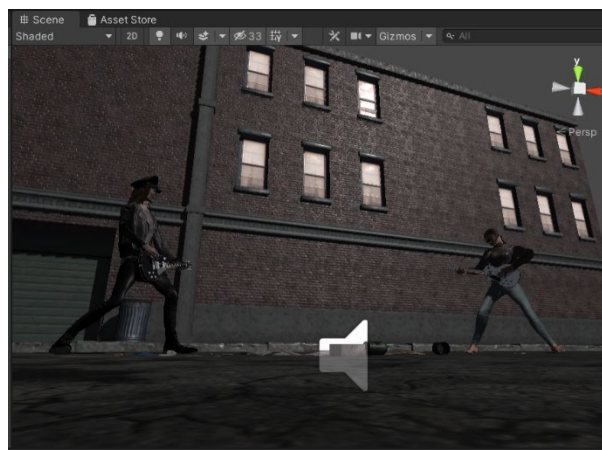


Figura 14: Escenario generado en **Unity**.

4.4. Generación de audio y sonidos

La importancia del audio en los videojuegos es altamente infravalorada – fenómeno curiosamente compartido, y estudiado, también por artes hermanas como el cine o medios audiovisuales – o pasa desapercibida para la mayoría de los usuarios y no pocos desarrolladores. Sin embargo, no debe serlo puesto que, cuando se trata del audio, se está tratando de una percepción sensorial que abarcaría el 50% de los estímulos sensoriales captados por una persona que conserve todas sus capacidades visuales y auditivas. En honor a este planteamiento se ha puesto una atención acorde a las necesidades del producto a desarrollar.

El software utilizado para generar los sonidos ha sido **Studio One 5**. Dicho software goza de gran popularidad actualmente debido a sus prestaciones y la relación calidad/precio que otorga. Además su oferta de pack completo **Presonus Sphere**, incluyendo una infinidad de librerías de sonido, *presets* y *plugins*, hacen que **Studio One 5** se convierta en una de las ofertas más competitivas en lo que a programas de creación musical se refiere.

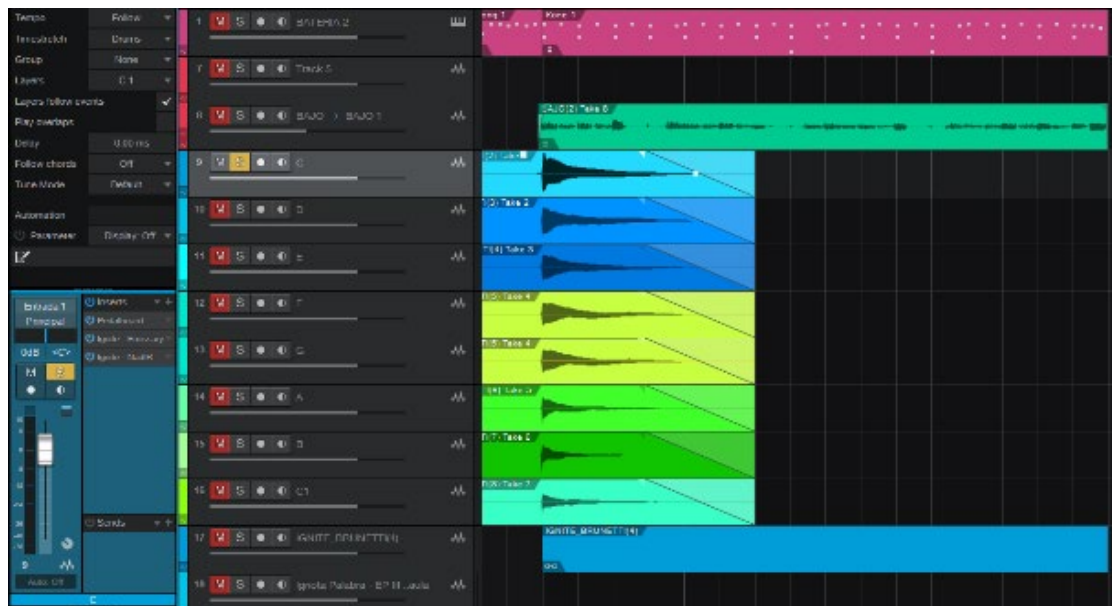
Loop de la pantalla de portada

En la pantalla de portada se utiliza un bucle extraído de la canción “**Abre la Jaula**” de la banda **Ignota Palabra**, de la cual se disponen todos los derechos por autoría de la obra. El bucle se repite constantemente mientras no se seleccione el comienzo del juego.

Generación de loops en el juego

Para crear los bucles necesarios utilizados en el juego se decidió recurrir a generar un patrón básico de batería y la grabación de un simple riff de bajo en escala de La menor, escala relativa de Do Mayor. Al patrón de batería se le aplicó el plugin **VST Impact XT**, de **Studio One**, mientras que al bajo se le aplicó un efecto de distorsión con **Ampire** para darle un sonido más acorde a la estética del juego.

Se generaron dos bucles: uno de **Intro**, que marcará cuando empieza el juego, y otro que se repetirá constantemente a lo largo del nivel y sobre el que se generarán las notas del jugador Pc.

Figura 15: imagen de proyecto de **Studio One**

Generación de sonidos de nota

Para los sonidos de nota emitidos por el ordenador se decidió grabar las notas sueltas directamente de la guitarra y utilizarlas en **Unity** para ser lanzadas como si de un sampler se tratara. El tratamiento del audio también se realizó en **Studio One 5**, aunque en esta ocasión se utilizó un plugin de terceros, gratuito, llamado **Ignite**. Dicho plugin es un simulador de amplificador de guitarra con un sonido muy realista y apropiado para el juego, aunque también hubiese sido perfectamente válido usar el plugin de **Presonus Ampire**.

Figura 16: [Ignite](#)

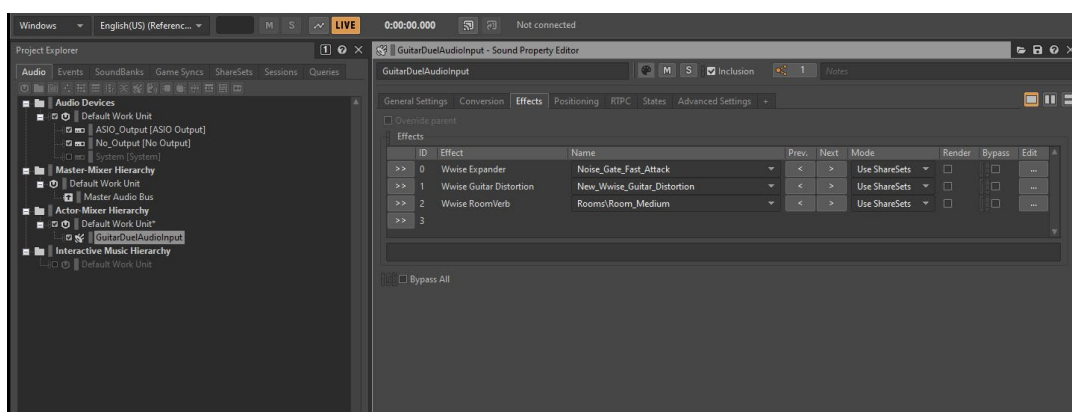
4.5. Lenguajes de programación y motor utilizado

Como ya se ha adelantado en apartados anteriores el motor de videojuegos utilizado para **Guitar Duel** es **Unity 3D**, utilizando **C#** como lenguaje de programación.

Las razones que me han llevado a decantarme por este software son su fácil acceso, una curva de aprendizaje no muy pronunciada, motor gráfico de gran calidad y la seguridad, tras cierta investigación, de que podría cubrir las necesidades técnicas requeridas. Asimismo, **Unity** es un software sumamente estandarizado y extendido por lo que encontrar información sobre cualquier problema que pudiera encontrarse parecía fácil. No obstante, las peculiaridades del proyecto, no compartidas por la mayoría de los videojuegos, han complicado su desarrollo más de lo que pudiera parecer habitual.

Por el momento, no se ha utilizado ningún *plugin* o *middleware*, externo a **Unity**. No se descarta la utilización de **Wwise** para el tratamiento del audio en **Unity**, porque es posible utilizar drivers **ASIO** que disminuirían la latencia de audio. Sin embargo, aún no está claro debido a que las pruebas realizadas hasta ahora no han demostrado una reducción de la latencia aunque si se obtiene mejoras en el sonido de la guitarra. Habría que valorar los pros y los contras, pero, por el momento, falta más estudio por parte del desarrollador de **Guitar Duel** para llegar a una decisión.

Además, tras haber eliminado el metrónomo, por ser sustituido por una base de audio en loop, se ha obtenido una grata sorpresa en cuanto al rendimiento de procesamiento del juego con una bajada de latencia importante. Esto ha permitido la introducción de mejoras, distorsión y ecualización, en el procesamiento del sonido de la guitarra llegando a ser bastante satisfactorio.



5. Implementación

5.1. Requisitos del juego

Guitar Duel es un juego que requiere de ciertos requisitos para poder ser jugado de forma satisfactoria. Está pensado para funcionar en plataforma PC y aunque no se necesita de un ordenador potente si es indispensable una interfaz de audio capaz de poder conectar una guitarra al ordenador. Además, este tipo de interfaces disponen de drivers optimizados, como **ASIO**, para poder trabajar con latencias reducidas.

Por otro lado, tratándose de un juego claramente *“de nicho”* si que se requieren por parte del jugador algunos conocimientos mínimos sobre la guitarra eléctrica. Por ejemplo, como se conecta, uso de los potenciómetros, o como se disponen y tocan las notas a lo largo del mástil.

5.2. Instrucciones de instalación

El juego no necesita instalación puesto que es lanzado mediante un ejecutable.

6. Demostración

6.1. Instrucciones de uso

El juego en sí es de gran facilidad de uso. Dispone de pocas opciones para garantizar la inmersión del usuario en la experiencia de juego.

No obstante, se detalla a continuación el funcionamiento básico del juego.

Pantalla de Portada



Figura 17: Pantalla de portada

Preferencias:

Haciendo click sobre el botón se despliega un menú donde se ha de seleccionar la entrada utilizada de la interfaz de sonido.

Umbral de detección:

En caso de que la detección de notas no sea satisfactoria se puede subir o disminuir el umbral de detección.

Pantalla de Juego



Figura 18: Pantalla de juego

Referencia del mástil:

Se representan las notas tocadas sobre el mástil y donde deben tocarse. En este caso en el quinto traste y las tres primeras cuerdas.

Magias:

Bajo las notas se incluyen las notas que deben ser tocadas, en cada momento, por el jugador.

Barras de vida:

Cuando la vida de uno de los dos jugadores llega a cero la partida termina.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

La implementación de un videojuego es una tarea complicada que enfrenta la resolución de no pocos problemas. Sin embargo, cuando el videojuego en cuestión requiere de una interfaz de usuario no estándar, como una guitarra, existen una serie de problemas añadidos sobre los que no es fácil encontrar soluciones.

Se ha de reconocer en estas líneas que el mero planteamiento de este juego, en la primera entrega del **TFG**, resultaba ciertamente arriesgado y que no estaba claro del todo que pudiera resolverse satisfactoriamente. El desconocimiento sobre el motor de juegos utilizado, **Unity** en este caso, aunque sobre cualquier otro, arrojaba muchas sombras sobre el proyecto: Como resolver el problema de la latencia, el reconocimiento de notas, la reacción del personaje a la entrada de audio, etc.

Problemas, todos ellos, que fueron planteados y se reflejan en la presente memoria, pudieron encontrar una solución adecuada sin recurrir a soluciones externas que complicarían aún más la implementación del juego.

Pero como diría Antonio Machado, “**Caminante, no hay camino, se hace camino al andar**”. Lanzarse adelante puede resultar arriesgado pero algo aprendido a largo de este trayecto es que, a veces, merece la pena tomar riesgos.

Sin embargo, la anterior afirmación no debería entenderse desde un punto de vista, filosóficamente hablando, idealista. Para asumir riesgos es necesario conocerlos, señalarlos y realizar una labor de planificación que permita establecer fechas, o periodos temporales concretos, dentro de los cuales ir disgregando problemas de mayor alcance en una suerte de problemas más pequeños y fáciles de solucionar o incluso plantear alternativas.

El juego ha podido ser creado tal y cual se presentaba en el planteamiento inicial. Ninguno de los problemas detectados en un principio ha quedado sin resolver, si bien pudieran ser mejorables como en el caso del sonido de la guitarra, y el videojuego queda perfectamente resuelto e implementado.

Por todo lo planteado en los párrafos anteriores, resulta doblemente satisfactorio que **Guitar Duel** sea hoy una realidad. Por un lado, los objetivos han sido conseguidos y, por otro, existe una confirmación

sobre una lección aprendida. Con esfuerzo y dedicación se pueden superar la mayoría de los obstáculos, pero esos obstáculos deben estar previamente definidos para poder ser esquivados.

7.2. Líneas de futuro

Guitar Duel queda abierto a futuras ampliaciones que hagan de este un videojuego más completo. La posibilidad de añadir nuevos personajes, guionizar una historia interna que permita al jugador identificarse con el protagonista del juego, creación de cinemáticas que la acompañen, etc. Son ideas que ya se indicaban en apartados anteriores de la memoria, como líneas de ampliación, de cara a crear un producto finalizado.

Línea argumental

Es la primera clave que se debería resolver para completar el producto. Se trata de desarrollar una historia guionizada que acompañe al usuario a lo largo de los diferentes niveles del juego logrando que se identifique como el personaje del jugador. Deberá el personaje ¿Salvar el mundo de un terrible villano? ¿Convertirse en el guitarra solista de la mejor banda de Rock del momento? ¿Simplemente ganar un concurso? Las posibilidades sobre la línea argumental son infinitas.

Personajes enemigos y escalas

Solucionada la línea argumental se deberían crear una serie de personajes que respondan a la historia. Cada nivel se convertirá en una misión, con un enemigo concreto y una escala musical en una zona del mástil de guitarra concretos. El tempo de la base empleada en cada nivel sería aumentado obligando a que las notas aparezcan con mayor rapidez incrementando la dificultad. Incluso en niveles finales podrían tocarse *licks* de guitarra completos. No obstante, esto último supondría mayor dificultad puesto que supondría una forma de trabajo diferente, y mucho más compleja, para el reconocimiento de notas.

Modo VS. 2 jugadores

Añadir un modo de juego para enfrentar a dos jugadores resulta muy atractivo para el juego. Sin embargo, abriría de nuevo una serie de problemas en el juego, además de la necesidad de una interfaz con dos entradas de instrumento, que quedan aquí a modo de apunte. ¿Es posible la utilización de dos entradas de audio en Unity? ¿Se incrementará por mucho el procesamiento aumentando la latencia? ¿Se trataría de un juego por turnos? ¿Se limitaría al uso de una escala concreta?

Bibliografía

Hardman, C. (2020) *Game Programming with Unity and C#*. Apress.

Lanham, M. (2017) *Game Audio Development with Unity 5.X*. Packt Publishing.

Hocking, J. (2022) *Unity in Action, Third Edition*. Manning Publications.

Pirkle, W. (2012) *Designing Audio Effect Plug-Ins in C++*. Focal Press.

Zoler, U. (2011) *Digital Audio Effects, Secon Edition*. Wiley.