Creación y Práctica de Ejercicios Elementales de Piano mediante un Videojuego Educativo

Creation and Practice of Elementary Piano Exercises through an Educational Video Game



Trabajo de Fin de Grado
Curso 2021–2022

Autor
Guillermo Sánchez de Lamadrid Martínez

Director
Federico Peinado Gil

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

Creación y Práctica de Ejercicios Elementales de Piano mediante un Videojuego Educativo

Creation and Practice of Elementary Piano Exercises through an Educational Video Game

Trabajo de Fin de Grado
Dpto. de Ingeniería del Software e Inteligencia Artificial

Autor
Guillermo Sánchez de Lamadrid Martínez

Director
Federico Peinado Gil

Convocatoria: *Junio 2022*

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

30 de mayo de 2022

Agradecimientos

Me gustaría agradecer a mi familia, a los que han probado la aplicación y a todo el equipo de Narratech Laboratories por su ayuda.

Resumen

Tras la pandemia de COVID-19, la necesidad de adaptar el mundo educativo a una forma de enseñanza telemática se ha convertido en una prioridad. Este es el caso también de la educación musical, la cuál todavía se encuentra en un proceso de adaptación a las nuevas tecnologías.

En el campo de la enseñanza de piano existe una necesidad de aplicaciones que, en lugar de intentar sustituir la educación tradicional en una academia o con profesor particular, se adapten a ella y la complementen. El desarrollo de una herramienta con esas características resulta especialmente útil para los alumnos que se encuentren en los primeros niveles de aprendizaje ya que, si se ludifica de la forma adecuada, podría mejorar mucho su motivación, ayudarles a practicar por su cuenta y brindarles la posibilidad de percibir su progreso a medida que mejoran sus habilidades con el instrumento.

El objetivo de este proyecto es desarrollar un juego serio en Unity que permita a profesores de piano y alumnos trabajar juntos de manera remota y asíncrona. La aplicación permite los docentes crear ejercicios personalizados a sus alumnos, para que estos puedan luego practicar con ellos en sus casas y, además, hacerlo de modo que sea una experiencia divertida y satisfactoria.

Para poder usar esta aplicación, los usuarios necesitarán un ordenador con sistema operativo Windows y un teclado MIDI compatible. Un primer prototipo de la aplicación se encuentra publicado de forma gratuita en la plataforma Itch.io, y el código fuente es abierto y accesible desde un repositorio público de GitHub. El prototipo ha sido sometidos a pruebas de usabilidad siguiendo la metodología SUS (System Usability Scale) y los resultados, aunque todavía discretos, dibujan prometedoras líneas de trabajo futuro para la mejora y el pulido de un futuro producto.

Palabras clave

MIDI, Piano, Música, Juego Educativo, Herramienta Educativa, Desarrollo de Videojuegos, Unity.

Abstract

After the COVID-19 global pandemic, the need to adapt from a physical learning environment to an online learning environment was a necessity; this also includes musical education, which is still in the process of adapting to new technologies.

In the field of piano education, there is a need for applications that are able to adapt and complement traditional education instead of trying to replace a traditional learning environment, whether it be in an academy or a private teacher. The existence of a tool with these characteristics would be especially useful for students who are in their first years of musical education, that if developed accordingly, could improve the students' motivation, help them practice on their own and give them the opportunity to visualise their progress as they improve skills on the instrument.

The purpose of this project is to develop a serious game in Unity that allows piano teachers and students to connect virtually. This tool will give teachers the opportunity to create personalised exercises for their students, therefore allowing them to practice at home in an entertaining and a gratifying way.

The user will only need a Windows computer and a compatible MIDI controller. The app is totally free and avaliable on Itch.io and GitHub. The tool has also been evaluated using SUS (*System Usability Scale*) tests with potiential users and the results, although still discreet, show promising lines of future work for the improvement and polishing of a future product.

Keywords

MIDI, Piano, Music, Educative Game, Educative Tool, Video Game Development, Unity.

Índice

1. Introducción **1**
- 1.1. Motivación . . . 2
- 1.2. Propósito y alcance . . . 2
- 1.3. Asignaturas relacionadas . . . 4
- 1.4. Estructura del trabajo . . . 5

2. Estado de la cuestión **7**
- 2.1. Enseñanza elemental de piano . . . 7
  - 2.1.1. Enseñanza tradicional . . . 7
  - 2.1.2. Enseñanza adaptada a las nuevas tecnologías . . . 8
  - 2.1.3. Complementos tecnológicos a la enseñanza . . . 10
- 2.2. Juegos serios . . . 10
  - 2.2.1. Juegos educativos y ludificación . . . 11
  - 2.2.2. Videojuegos educativos musicales . . . 14
- 2.3. Informática musical y programación de audio . . . 15
  - 2.3.1. Tecnologías de representación musical computacional . 16
  - 2.3.2. Herramientas VST . . . 17
  - 2.3.3. MIDI y VST en la actualidad . . . 17

3. Objetivos y especificación **25**
- 3.1. Objetivos propuestos . . . 25
- 3.2. Especificación . . . 26
  - 3.2.1. Requisitos y funcionalidades . . . 26
  - 3.2.2. Limitaciones . . . 28

4. Metodología y herramientas **31**
- 4.1. Metodología de desarrollo . . . 31
  - 4.1.1. Preproducción . . . 31
  - 4.1.2. Producción . . . 35
  - 4.1.3. Postproducción . . . 35
- 4.2. Herramientas utilizadas . . . 36

4.2.1. Unity . . . 36
4.2.2. MidiJack . . . 36
4.2.3. GitHub . . . 36
4.2.4. Midi Editor . . . 36
4.2.5. Visual Studio 2019 . . . 37
4.2.6. Adobe Photoshop . . . 37
4.2.7. Slack, Google Meet y Google Drive . . . 37
4.2.8. Count Down . . . 37
4.2.9. Runetime File Browser . . . 37
4.2.10. Controlador MIDI Teclado Alesis VI25 . . . 38
4.2.11. OBS Studio . . . 38

5. Análisis, diseño e implementación de Piano Unite 39
5.1. Análisis y diseño . . . 39
5.1.1. Primeros bocetos . . . 39
5.1.2. Prototipo . . . 40
5.1.3. Diseño de la aplicación . . . 42
5.1.4. Interfaz de usuario . . . 47
5.1.5. Modificaciones finales . . . 48
5.2. Desarrollo . . . 49
5.2.1. Esquema de recursos . . . 49
5.2.2. Desarrollo del piano virtual . . . 52
5.2.3. Desarrollo de la lectura MIDI . . . 54
5.2.4. Desarrollo de la escritura MIDI . . . 55
5.2.5. Desarrollo de la jugabilidad . . . 57
5.2.6. Menús y HUD . . . 58
5.2.7. Implementación del audio . . . 60
5.2.8. Demo jugable . . . 61
5.2.9. Funcionalidades auxiliares . . . 62

6. Pruebas y resultados 67
6.1. Pruebas . . . 67
6.1.1. Guión de las pruebas . . . 68
6.1.2. Preguntas realizadas . . . 69
6.2. Resultados y discusión . . . 72
6.2.1. Características de los participantes . . . 73
6.2.2. Resultados del cuestionario SUS . . . 75
6.2.3. Resultados de las preguntas de desarrollo . . . 77
6.2.4. Discusión de los resultados . . . 79

7. Conclusiones 85

7.1. Objetivos alcanzados 86
7.2. Trabajo futuro 87
7.3. Reflexión final 87

A. Conclusions **89**
A.1. Achieved objectives 90
A.2. Future work 91
A.3. Final thoughts 91

B. Manual de uso **93**

C. Tabla Completa de Bytes de estado MIDI **97**

Bibliografía **105**

Índice de figuras

2.1. Diagrama de los dos bloques en los que la enseñanza elemental de piano se suele diferenciar. 8
2.2. Fotograma de uno de los cursos de piano del portal web Musihacks impartidos por Jaime Altozano, en el que explica armonía. 9
2.3. Captura de pantalla de la aplicación de escritorio de Duolingo, donde podemos observar la estructura de la aplicación. Se pueden discernir distintas mecánicas antes comentadas como los trofeos, las bonificaciones, los niveles de experiencia o los distintos ejercicios y su progreso. 12
2.4. Conjunto de gráficas que muestran el concepto de flow y curva de dificultad. (Chen, 2007). 19
2.5. Diagrama que sintetiza la Taxonomía de Bloom, revisada por (Bloom y Krathwohl, 2001). Las categorías que se encuentran en la parte superior son aquellas que indican una mayor comprensión del contenido aprendido. 20
2.6. Captura de pantalla de una partida para principiantes en Synthesia (Synthesia LLC, 2006), en el modo de juego de ambas manos, haciendo uso de un teclado MIDI. 20
2.7. Captura de pantalla de nivel de práctica de interpretación de partitura en tiempo real. Se puede observar cómo la línea azul se va desplazando hacia la derecha, indicando en que momento debe de ser presionada cada nota. 21
2.8. Captura de pantalla de nivel del menú principal de Simply-Piano. 21
2.9. Fragmento de tabla de interpretación de los distintos bytes que componen un mensaje MIDI (MIDI Association, 2021). . 22
2.10. Interfaz de Guitar Rig 6 Demo (Native Instruments, 2020), VSTi que simula una guitarra eléctrica, el cual posee multitud de filtros, efectos, distorsiones y amplificadores para modificar su sonido. 23
2.11. Fragmento de tabla de los distintos identificadores de instrumentos posibles con MIDI 24

2.12. Tablatura MIDI en el programa de producción musical FL Studio (Image-Line Software, 2021). 24

3.1. Diagrama general que describe el flujo de uso de la aplicación. 29

5.1. Primero borrador de diseño de la jugabilidad de la aplicación. 41
5.2. Captura del primer prototipo jugable de la herramienta en el motor *Unity*. 42
5.3. Menú de opciones de sonido. 43
5.4. Pantalla de historial de partidas. 44
5.5. Escena de puntuación. 45
5.6. Diferentes fases de la escena de resolución de ejercicios. Orden: De izquierda a derecha y de arriba a abajo. 46
5.7. Diferentes fases de la escena de grabación. Orden: De izquierda a derecha y de arriba a abajo. 48
5.8. Versiones finales de los botones que posee el juego. Se puede observar como los únicos con tonos rojos son el de salir de la aplicación, y el que se usa para indicar algun tipo de error o advertencia. 49
5.9. Mensaje de tutorial que se muestra en la versión para alumnos, al iniciar la escena de *PlayLevel* por primera vez. 50
5.10. Captura de la jerarquía de carpetas del proyecto en *Uniy*. . . 50
5.11. Captura de todos los sprites usados en el juego. 52
5.12. Diagrama de organización de los eventos MIDI recopilados. . 56
5.13. Captura de objeto *Button* de *Unity*. Se puede ver las distintas opciones de personalización del botón que ofrece. Este componente ahorra mucho tiempo, facilitando la implementación de botones en la aplicación. 59
5.14. Versión final del menú (en la versión para profesores). Empezando por la izquierda, encontramos las siguientes opciones: Grabar ejercicio, jugar un ejercicio, observar las mejores puntuaciones, la pestaña de opciones y el botón de cerrar aplicación. 60
5.15. Proyecto de Photoshop de los sprites de las distintas teclas de una octava en un piano. Posteriormente, serán separadas en notas individuales. 61
5.16. Mezclador de sonido integrado en *Unity*. 62
5.17. Capturas de pantalla de los textos de introducción. Sólo se enseñan la primera vez que se inicia la aplicación. 63
5.18. A la izquierda, la pantalla de menú principal y la de grabación del juego antes de los cambios visuales. A la derecha, las mismas pantallas después de los cambios visuales. 64
5.19. Estructura de los datos de guardado. 65
5.20. Sistema de partículas de Unity. 66

6.1. Plantilla de cuestionario SUS para aplicaciones web. 70
6.2. Foto de persona jugando a Piano Unite. 72
6.3. Gráfica que recopila el nivel de experiencia en piano de los usuarios de prueba. 73
6.4. Gráfica que recopila el nivel de formación en piano de los usuarios de prueba. 74
6.5. Gráfica que recopila la relación de los usuarios con la enseñanza de piano. 74
6.6. Gráfica que recopila las edades de los participantes. 75
6.7. Respuestas de las distintas experiencias de los participantes con aplicaciones similares. 75
6.8. Respuestas de los distintos controladores MIDI usados en las pruebas. 76
6.9. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 7 del cuestionario. 76
6.10. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 8 del cuestionario. 77
6.11. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 9 del cuestionario. 78
6.12. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 10 del cuestionario. 79
6.13. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 11 del cuestionario. 80
6.14. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 12 del cuestionario. 81
6.15. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 13 del cuestionario. 82
6.16. Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 14 del cuestionario. 82
6.17. Capturas de dos de las respuestas a la pregunta 15 del cuestionario. 83
6.18. Captura de pantalla de una de las respuestas a la pregunta 15 del cuestionario. 83

B.1. Menú principal. 94
B.2. Diferentes fases de la escena de grabación. Orden: De izquierda a derecha y de arriba a abajo. 94
B.3. Diferentes fases de la escena de resolución de ejercicios. Orden: De izquierda a derecha y de arriba a abajo. 95
B.4. Escena de puntuación. 95
B.5. Pantalla de historial de partidas. 96
B.6. Menú de opciones de sonido. 96

Capítulo 1

Introducción

Desde la implementación de contenido audiovisual en las aulas, hasta el uso de entornos virtuales interactivos, el empleo de nuevas tecnologías en la educación se ha convertido en uno de los pilares en donde se cimientan muchos de los sistemas educativos actuales en la mayoría de paises de la OCD (Martín-Laborda, 2005).

De la misma manera, existe evidencia de que el uso de la ludificación a la hora de impartir un contenido educativo puede aumentar tanto la motivación, como la capacidad de asimilación de lo aprendido por parte del alumno (Zichermann y Cunningham, 2011). Este fenómeno es más notable en la educación de los niños españoles en la actualidad, para muchos de los cuales los videojuegos forman una parte intrínseca de su vida, como podemos deducir de las estadísticas divulgadas en la publicación *La Industria del Videojuego en España - Anuario 2019* (AEVI, 2019).

Debido al gran éxito de la ludificación como estrategia de enseñanza, en 1970 se acuñó el término de *juego serio* (Abt, 1970), que refiere a un juego cuyo principal principal objetivo no reside en divertir al jugador, sino que su propósito es otro como pueden ser la sanidad, educación, defensa, ingeniería, política o religión, entre otros (Serious Games Society, 2014).

En el mundo de la educación musical este concepto no es en absoluto ajeno, pues existen en el mercado un considerable número de cursos en línea interactivos, aplicaciones móviles y videojuegos orientados al aprendizaje de un cierto instrumento, así como para el aprendizaje de teoría musical.

El piano y la guitarra son comúnmente aceptados como los dos instrumentos más populares para aprender a tocar debido a su versatilidad y sencillez a la hora de iniciarse en ellos, así cómo su omnipresencia en la música popular de la actualidad. Esto se puede ver reflejado en el éxito de juegos como *Synthesia* (Synthesia LLC, 2006) o *Rocksmith 2014* (Ubisoft, 2014), dos de los mayores exponentes de juegos musicales centrados en estos instrumentos respectivamente.

1.1. Motivación

Como se expone anteriormente, existe una gran cantidad de software enfocado al aprendizaje de nociones básicas de un instrumento musical. Sin embargo, muchos de estos programas y juegos se presentan como sustitutos de la educación presencial individualizada que podemos conseguir con un profesor, en lugar de un complemento a esta. Por ello, no están diseñados con flexibilidad y adaptabilidad suficiente para usarse con distintos métodos de enseñanza o adecuarse a las necesidades específicas de cada alumno.

Las evidencias apuntan a que un tipo de educación musical que pueda combinar la enseñanza presencial e individualizada de un profesor con las nuevas tecnologías parece ser la opción preferible a la hora de aprender a tocar un instrumento (Calderón, 2002). Además, la necesidad de un tipo de educación que pueda ser parcialmente impartida de manera telemática se hace mucho más evidente en la actualidad donde, debido a la pandemia del *COVID-19* la realización de una clase donde alumno y profesor puedan compartir espacio no siempre es una alternativa viable.

Por otro lado, publicaciones como la de *Retuning Our Ambition for Music Learning* (The Music Commission, 2019) apuntan a que la educación musical tiene problemas para adaptarse a las nuevas tecnologías, lo cual califica como obstáculo para su avance y conexión con las nuevas generaciones de estudiantes. Más concretamente, una de las mayores problemáticas a la hora de enseñar un instrumento a los niños es la de motivarles a que practiquen y realicen los ejercicios que les son asignados como deberes para casa, ya que se pueden ver desincentivados a realizarlos debido a sus estudios o a la gran cantidad de distracciones a las que están expuestos hoy día en sus hogares.

Por ello, las intenciones de este proyecto son dos. Por un lado, brindar a los docentes (más concretamente, a los profesores de piano) una herramienta con la que puedan motivar a sus alumnos a practicar y mejorar en la digitación, adaptable a sus diversos métodos de enseñanza, sencilla e intuitiva de usar y que permita al profesor proponer ejercicios de manera telemática. Por otro, ofrecer a los alumnos un videojuego con el que puedan practicar lo aprendido en clase, divertido, accesible y que les brinde recompensas tanto intrínsecas como extrínsecas (Zichermann y Cunningham, 2011), para motivarles a practicar con el instrumento el máximo tiempo posible.

1.2. Propósito y alcance

Como se ha adelantado en la Sección 1.1, este proyecto aspira a creación una plataforma de carácter educativo que pueda servir como complemento telemático para la enseñanza de piano. Más concretamente, el juego se centrará en la práctica de digitación en piano elemental, para niños de 8 a 14 años, ya que los 8 años es la edad mínima para acceder al primer curso del

Conservatorio.

Aunque se expondrán algunas ideas al respecto en el Capítulo 2, este videojuego no aspira a enseñar Teoría Musical ni Historia de la Música, y de ninguna manera pretende ser usado como sustitución de un curso completo de piano impartido por un profesor. Este juego está pensado para funcionar como complemento educativo a disposición del docente, permitiéndole adaptar los juegos que se generan a su programa educativo, pudiendo crear los ejercicios que el docente considera más adecuados.

Para el correcto desarrollo de la plataforma, se elaborará un marco teórico con los principios que deben seguir los juegos serios de carácter educativo y sobre cómo debe realizarse la enseñanza elemental de piano. Para ello se contará con las aportaciones de docentes y ejercitantes en la materia para perfeccionar lo máximo posible todos los aspectos de la aplicación.

Este videojuego educativo recibirá el nombre de *Piano Unite* y actuará como una herramienta con la que el profesor de piano podrá grabar la melodía o pieza musical que desee. Tras ello, se generará de manera automática un nuevo nivel para un juego del estilo de *Synthesia* (Synthesia LLC, 2006) (adaptado de manera audiovisual y en cuanto usabilidad para un público infantil). El alumno podrá jugar y practicar ese nivel más adelante, todas las veces que lo necesite, sin que sea necesaria la presencia ni supervisión del profesor. Este nivel consistirá básicamente en un fichero de datos muy ligero que el docente podrá guardar de forma local entre los demás ficheros del juego, o que podrá enviar por Internet al alumno para que este lo use en su propia versión del programa. Asimismo, *Piano Unite* evaluará el rendimiento del alumno en cada nivel, y de este modo permitirá al profesor plantear una serie de objetivos para el propio juego con la finalidad de mantener motivado al estudiante. El programa contará también con algunas partituras y piezas utilizadas comúnmente en la enseñanza elemental de piano.

A nivel tecnológico, no se pretende desarrollar ninguna librería ni protocolo nuevo de tratamiento musical ni sonoro. Se trata de explorar las posibilidades que ofrece el estándar tecnológico MIDI (*Musical Instrument Digital Interface*), de usar instrumentos virtuales a la hora de desarrollar videojuegos, y de integrar en el sistema el control de un teclado MIDI conectado al PC del usuario.

Por último, aunque este videojuego educativo trate de conectar telemáticamente a profesores y alumnos, no estará enfocado como una plataforma web que permita un sistema de registro y seguimiento de usuarios. En su lugar se ha planteado una aplicación sencilla de uso exclusivamente local, cuyos ficheros de datos puedan ser intercambiados manualmente mediante métodos comunes como el correo electrónico o la mensajería instantánea. Aún así, este programa será desarrollado dejando abierta la posibilidad de implementar funcionalidades web en el futuro, así como poder añadir más dinámicas de juego enfocadas a la enseñanza de otros aspectos del piano o

de otros instrumentos.

1.3. Asignaturas relacionadas

En esta sección se describe la relación de este Trabajo de Fin de Grado (TFG) con los contenidos impartidos a lo largo del Grado en Desarrollo de Videojuegos.

Primero, la realización del diseño del programa, así como su GDD (del inglés, *Game Design Document* o *Documento de Diseño de Juego*) estará fuertemente basado en los contenidos impartidos en asignaturas como *Diseño de videojuegos*, *Desarollo de Sistemas Interactivos* y la asignatura optativa de *Juegos Serios*. Se hace hincapié en está ultima, puesto que el proyecto desarrollado, al tratarse precisamente de un *juego serio*, o más concretamente juego educativo, seguirá de manera fiel los conceptos de diseño desarrollados en dicha asignatura.

Por el lado más técnico, el proyecto se apoyará de manera sólida en las asignaturas de *Sonido en Videojuegos* e *Informática musical*, debido a su enfoque como videojuego musical. También en la asignatura de *Motores de Videojuegos* ya que se utiliza Unity, mismo motor que se utiliza allí. Ciertos sistemas se inspiran en conceptos vistos en *Inteligencia Artificial para Videojuegos*; y por último asignaturas como *Estructuras de Datos y Algoritmos*, *Fundamentos de la Programación I y II*, *Tecnología de la Programación de videojuegos*, *Métodos Algorítmicos en Resoluciones de Problemas*, *Videojuegos en Consola* o *Videojuegos para dispositivos móviles* también resultan de utilidad porque, aunque no tengan relación directa con el tema del trabajo, todas conforman nuestra base de conocimientos de programación, arquitectura software y resolución formal de problemas que tan importante es aplicar en el desarrollo de software.

Las pruebas de usuario del proyecto seguirá las nociones principales impartidas en la asignatura *Usabilidad y Análisis de Juegos*. Y para la parte artística, se aplican conceptos y técnicas aprendidas en *Modelado en 2D y 3D* y *Principios de Dibujo, Color y Composición*.

Por último, mencionar *Proyectos I, II y III*, *Programación de Videojuegos en Lenguajes Interpretados*, *Metodologías Ágiles de Producción* y *Game Lab V* (esta última asignatura cursada gracias a la beca Erasmus en el Grado *Games and Entertainment Technology* de la *Nord University, Steinkjer, Norway*), porque me han aportado experiencia y conocimientos necesarios para sistematizar las fases de producción de un proyecto como este.

Adicionalmente, ha sido necesario buscar conocimientos ajenos al Desarrollo de Videojuegos como son los de Teoría Musical, el instrumento del Piano y los métodos para la Enseñanza de Piano.

1.4. Estructura del trabajo

El presente trabajo se estructura como sigue:

- En este Capítulo 1 se presenta el propósito de este TFG y el contexto en el que se va a realizar.
- En el Capítulo 2 se revisa el estado de la cuestión en materia de informática musical, ludificación en juegos serios y nuevas tecnologías aplicadas a la educación musical, siendo estos los tres pilares en los que se basa el marco teórico de este proyecto.
- En el Capítulo 3 se presentan los principales objetivos del trabajo, así como una explicación de todos los requisitos necesarios para el desarrollo del videojuego.
- En el Capítulo 4 se explica la metodología seguida para la realización del proyecto, junto a una enumeración y justificación de las herramientas usadas.
- En el Capítulo 5 se realiza una explicación detallada del diseño e implementación del proyecto, así como del juego educativo resultante en su totalidad, explicando los distintos obstáculos y problemas que se han tenido que superar para su completo desarrollo.
- En el Capítulo 6 se recopilan las opiniones de los especialistas consultados acerca del prototipo desarrollado y se plantean las pruebas de usuario según la demografía del público objetivo, es decir, tanto para docentes del piano como para alumnos de dicho instrumento.
- En el Capítulo 7 se da una conclusión a este trabajo, repasando si se han cumplido los objetivos previamente definidos, y planteando posibles ampliaciones y líneas de trabajo para el futuro.

Capítulo 2

Estado de la cuestión

Como se deduce del capítulo anterior, dado el propósito de este proyecto, los pilares teóricos en los que se debe sostener el trabajo son los siguientes: la enseñanza elemental de piano, la ludificación en juegos serios y la informática musical . Por ello este capítulo se divide en tres secciones, en las cuales se explica, de manera general, la actualidad de cada uno de estos temas, así como aquellas referencias que resultan de especial utilidad para el presente trabajo.

2.1. Enseñanza elemental de piano

En este apartado, indagaremos brevemente en la actualidad de la enseñanza elemental de piano, desde la enseñanza tradicional impartida tanto en conservatorios como en escuelas de música, hasta las nuevas formas de enseñanza online.

2.1.1. Enseñanza tradicional

Existen distintos métodos de enseñanza de piano. El método denominado *El Piano* (Tchokv/Gemiu, 1994), es uno de los mas usados en escuelas de música y conservatorios en España. Otro de los más populares a nivel europeo (Roldán Alcázar, 2010) es la *Escuela Rusa de Piano* (Nikolaev, 2007), proveniente de la Unión Soviética.

Las diferencias principales en los distintos métodos suelen residir en el tipo de ejercicios que se proponen, las partituras que se estudian o el orden en el que se inculcan ciertos conocimientos, ya que el funcionamiento de una clase de piano estándar suele ser, por lo general, similar en todas ellas. En la clase, el profesor se sienta delante del piano con el alumno durante un periodo de alrededor de una hora, donde se trabaja de forma individualizada y supervisada. Al finalizar la clase, el profesor mandará tareas para que el alumno realice por su cuenta. Estos ejercicios podrán ser de carácter tanto

práctico como teórico.

Esto nos lleva al otro punto común que se suele encontrar entre los diferentes métodos, que consiste en el hecho de que se suelen distinguir las competencias impartidas en dos bloques. El primer bloque se centra en la digitación, la correcta identificación de las distintas notas a lo largo del piano y la adaptación a distintos tipos de compases, patrones melódicos y rítmicos. El segundo bloque tiene el énfasis en teoría musical y todo lo que ello engloba, como escalas, acordes, arpegios o lectura y escritura de pentagramas, entre otros conocimientos.

Esta distinción entre dos bloques suele estar basada en que, para aprenderlos bien, el primero requiere de práctica delante de un piano, mientras que el segundo está mas basado en el estudio y la asimilación de conocimientos teóricos.

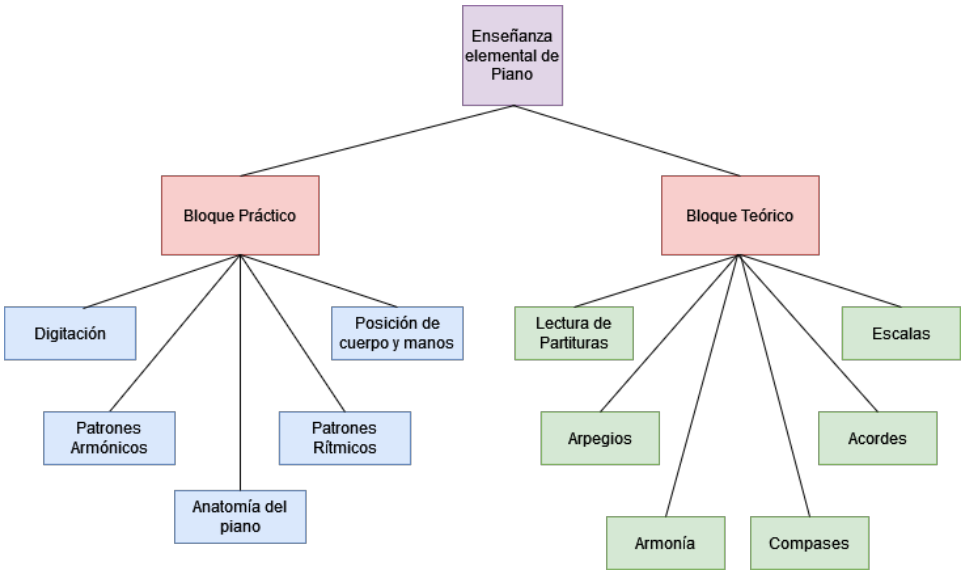


Figura 2.1: Diagrama de los dos bloques en los que la enseñanza elemental de piano se suele diferenciar.

2.1.2. Enseñanza adaptada a las nuevas tecnologías

Podría decirse que la inclusión de las nuevas tecnologías en la enseñanza de piano se remonta a los años 70, con la introducción de cintas de casete y, mas adelante, CDs que los alumnos pueden escuchar en su casa. De hecho, tanto los libros mencionados en el apartado anterior, como casi cualquier libro de aprendizaje de piano que se encuentre actualmente a la venta, incluye en la gran mayoría de ocasiones un contenido sonoro, ya sea por medio de CDs o de archivos MP3 para descargar. Este contenido suele consistir principalmente en la interpretación de las distintas partituras que se imparten en el método,

así como posible acompañamiento instrumental o instrucciones guiadas para realizar ciertos ejercicios.

Cada vez es más habitual, gracias a la expansión de Internet, la oferta y la demanda de cursos online que permitan al alumno aprender de forma autodidacta y sin la necesidad de asistir a una academia ni contratar a un profesor particular. Métodos cómo *Musihacks: Piano Básico* (Altozano, 2021) o *Aprender Música Online* (Valbuena) gozan de bastante éxito en España y otros países hispanohablantes, ya que prometen una formación de piano efectiva, rápida, cómoda y a un precio muy reducido si se compara con las tasas de una escuela de música. Este tipo de cursos basan su método en guías escritas, ejercicios, posibilidad de tutorías y, sobre todo, en vídeos que, a modo de tutorial, condensan y exponen todas las lecciones impartidas. Otro pilar fundamental que siempre está presente en esta clase de enseñanza online es el aprendizaje de partituras tanto clásicas como modernas, con el objetivo de que el alumno se mantenga motivado. Estos cursos, además, promueven la creación de comunidad online de alumnos que se apoyen y ayuden mutuamente.



Figura 2.2: Fotograma de uno de los cursos de piano del portal web Musihacks impartidos por Jaime Altozano, en el que explica armonía.

A pesar de lo convenientes que pueden ser estos cursos para un público adulto que, gastando poco dinero, puede aprender a su ritmo, su carácter impersonal así como su enfoque únicamente online hacen que no sea la manera idónea para educar una demografía infantil, la cual requiere de un profesor que se adapte a sus necesidades y gustos, que le proporcione motivación, que pueda resolver dudas y que siga de cerca su progreso.

Una alternativa para este tipo de usuarios que necesitan una mayor mo-

tivación y un acercamiento más lúdico al aprendizaje elemental de piano pueden ser los videojuegos y aplicaciones de aprendizaje de piano, como *FlowKey* (FlowKey, 2015), *SimplyPiano* (JoyTunes, 2017) o su versión para niños *Piano Dust Buster* (JoyTunes, 2012). Estas aplicaciones, por medio de estrategias de ludificación, consiguen convertir el proceso de aprendizaje de piano ortodoxo en un juego, donde los alumnos deberán ir completando objetivos a corto y largo plazo constantemente. Se analizará el funcionamiento de estas aplicaciones más detalladamente en la siguiente sección.

2.1.3. Complementos tecnológicos a la enseñanza

De esta sección anteriormente expuesta se puede sacar una serie de conclusiones para el correcto enfoque y desarrollo de la herramienta *Piano Unite*.

De la popularidad de los cursos y de las aplicaciones de aprendizaje online mencionadas anteriormente se puede extraer que hay una gran demanda de este tipo de servicios. Además, existe una necesidad de un mayor nivel de interactividad que un vídeo no es capaz de proporcionar, así como de un superior nivel de libertad a la hora de aplicar un método personalizado que los juegos y aplicaciones del mercado no consiguen alcanzar.

Por ello, con el fin de hacer una herramienta que pueda ser compatible con la enseñanza tradicional de un profesor de piano, esta no deberá de entrometerse en la parte teórica ni en la metodología del docente en particular, no deberá aspirar a 'sustituirla', sino que la aplicación se centrará unicamente en reforzar el bloque práctico de aprendizaje. Será planteada como un complemento a la enseñanza que realiza el profesor. Además, como cada método posee una serie de partituras y ejercicios propios, esta deberá de dar la opción al profesor de diseñar e introducir los que él considere.

2.2. Juegos serios

Como se ha comentado en el capítulo de introducción, el concepto de juego serio suele vincularse de forma automática a una aplicación de software que, con propósitos formativos, hace uso de estrategias de ludificación. Sin embargo, no todo juego serio tiene la intención de enseñar, sino que tambien se puede considerar juego serio a uno que trate de, por ejemplo, promover una alimentación sana o servir como parte de terapia psicológica (Dörner et al., 2016). Esto nos sugiere el enorme potencial que puede tener el uso de juegos para propósitos más allá del sólo entretenimiento.

Es por ello que, en el mundo académico, se ha escrito mucho sobre juegos serios y su efectividad. De la misma manera, existen múltiples asociaciones a lo largo del mundo con el fin de discutir sobre ellos y promoverlos como parte indispensable en las nuevas formas de educación, como la Serious Games Society (Serious Games Society, 2014).

2.2.1. Juegos educativos y ludificación

El diccionario de Cambridge define el término *gamification* (gamificación o ludificación en español) como "La practica de incluir elementos de juegos en una actividad, para hacerla más interesante o disfrutable". Sin embargo, estos no son los únicos beneficios que nos trae el hecho de ludificar una actividad educativa, sino que también nos ofrece otras ventajas como mejorar la motivación del usuario, aumentar el número de participantes, mejorar la fidelización o incluso sacar un mayor retorno de inversión económico de nuestro producto educativo. Además, los videojuegos también resultan ser uno de los entornos más adecuados para aprender o aplicar un conocimiento determinado, ya que se trata de un entorno seguro donde el usuario no tiene miedo a fallar o equivocarse. En un juego siempre existe la posibilidad de volver a intentarlo.

Por ello, existen una gran cantidad de juegos serios con propósitos educativos(juegos educativos para abreviar) que han gozado de gran éxito como Code Combat (CodeCombat Inc., 2013), juego orientado a un público infantil para enseñar fundamentos de programación o ALSIM ALX (Aslim, 2008), un simulador de vuelo que es usado en multitud de academias de pilotos. Otra herramienta educativa que ha aprovechado de manera excelente el uso de la ludificación y que además goza de gran éxito es *Duolingo* (Duolingo Inc., 2011), la cual con más de 8 millones de usuarios activos diariamente se ha convertido en la aplicación de referencia para aprender idiomas a nivel mundial. Por ello, esta será la herramienta que se utilizará para ejemplificar algunos aspectos sobre ludificación que comentaremos más adelante.

En el escrito *Good Video Games and Good Learning* (Gee, 2013), el autor nos expone 16 principios educativos que un buen videojuego debe incorporar, de manera que este cumpla con el propósito de enseñar, entretener y generar interés al jugador en el tema tratado.

Cuando se desarrollan herramientas educativas interactivas, hay dos aspectos principales a tener en cuenta a la hora de diseñarlas: la forma en la que se va a motivar al usuario y la manera en la que se va a evaluar su valor educativo.

2.2.1.1. Técnicas de motivación al jugador

Existen dos formas en las que se puede motivar al jugador, de manera intrínseca o de manera extrínseca. De forma ideal cualquier videojuego debería potenciar ambas (Klopfer et al., 2018).

La motivación extrínseca es aquella que cuyo origen es externo al jugador, es decir, aquellas recompensas que los desarrolladores ofrecen al jugador con el fin de darle una ilusión de progresión. Por ejemplo, en *Duolingo* hay una serie de niveles, trofeos y bonificaciones que los usuarios van adquiriendo a medida que van avanzando en los distintos temarios y ejercicios. Estos niveles

de experiencia y trofeos se registran en tu perfil, y pueden ser comparados con otros usuarios, de manera que fomente una competitividad sana. Los bonificadores consisten en elementos que te brindan mejoras dentro del juego, como pueden ser las vidas, que te permiten tener más intentos por ejercicio. Estas vidas suelen conseguirse utilizando dinero real o esperando una cierta cantidad de tiempo, por lo que poder hacerse con ellas gracias a tu buen rendimiento produce una sensación de satisfacción notable en el jugador.

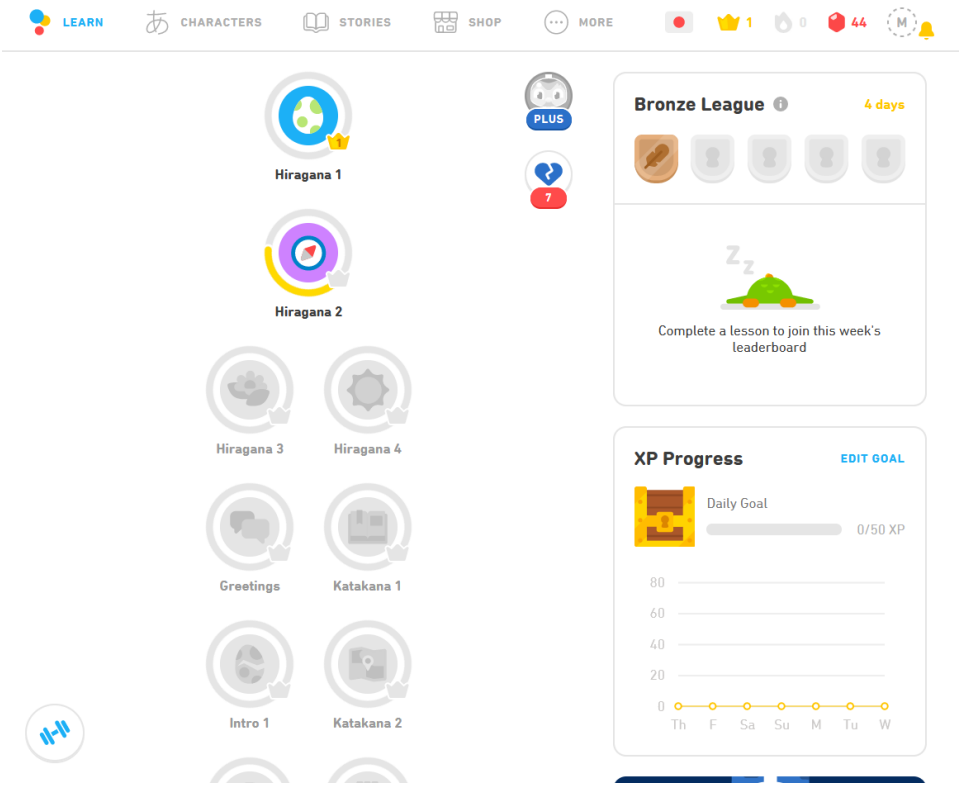


Figura 2.3: Captura de pantalla de la aplicación de escritorio de Duolingo, donde podemos observar la estructura de la aplicación. Se pueden discernir distintas mecánicas antes comentadas como los trofeos, las bonificaciones, los niveles de experiencia o los distintos ejercicios y su progreso.

Por otro lado, la motivación intrínseca es aquella que proviene de la propia sensación de mejora del jugador. Para que esta se produzca se debe de presentar al jugador una serie de retos que le fuercen a mejorar y adaptarse para poder superarlos. Para ello, el juego idealmente debería proporcionar un espacio más o menos seguro al consumidor donde pueda aprender y practicar sus habilidades para, cuando se estime que debería de estar preparado, proponerle una actividad que ponga a prueba sus habilidades. En el caso de un juego educativo que pretenda enseñar al usuario alguna disciplina externa a las propias mecánicas del juego, es conveniente asegurarse de que estos

retos puedan servir como exámenes que midan el nivel en el que se encuentra el jugador tras completarlos. Por ejemplo, en *Duolingo*, hay una serie de controles al final de cada temario llamados çheckpoints"que cumplen esta función. Sin embargo, *Duolingo* no se queda ahí, si no que en idiomas como el inglés también ofrece exámenes online certificados por un gran número de instituciones, lo cual evita al usuario tener que acudir a academias o centros convencionales, las cuales exigen un mayor desembolso económico y de tiempo (Duolingo Inc.).

Duolingo también incorpora una modalidad en la que un docente puede dirigir y monotorizar el progreso de sus alumnos, proponerles una serie de ejercicios en concreto e incluso personalizar los contenidos que se imparten.

Esta aplicación posee un gran número de funcionalidades que no se han mencionado pero que también tienen como fin motivar a los usuarios, como pueden ser los duelos multijugador, las opciones de personalización del avatar o los informes del progreso diario, entre otras.

Si se quisiera indagar de manera más profunda en el campo del diseño de videojuegos y técnicas para que el jugador se mantenga entretenido y sienta una progresión satisfactoria se podrían comentar conceptos como el *flow* (Figura 2.4) de juego, la curva de dificultad o análisis de los sentimientos que se quieren evocar en el jugador, entre otros. Todos estos son nociones que se desarrollan de manera sublime en el libro *The Art of Game Design: A book of lenses* (Shell, 2008), y que deberían ser aplicadas en cualquier clase de juegos, sean estos juegos serios, juegos competitivos o juegos puramente lúdicos.

2.2.1.2. Técnicas de enseñanza en videojuegos

Lo primero que se debe señalar acerca del aprendizaje en videojuegos es que, en absolutamente todos los juegos, sean estos educativos o no, el aprendizaje por parte del jugador es un ingrediente fundamental. Este aprendizaje puede ser sobre las propias mecánicas del juego, sus dinámicas, las habilidades del jugador, el comportamiento de los enemigos, la historia o el mundo donde se desarrolla el mismo, por nombrar unos pocos ejemplos. Además, estos pueden llegar a ser la forma ideal de adquirir habilidades como las siguientes: reconocimiento y percepción espacial desarrollo de la capacidad y atención visual, desarrollo de lógica inductiva, desarrollo cognitivo en aspectos científicos y técnicos, desarrollo de habilidades complejas, representación especial, descubrimiento inductivo o construcción de códigos icónicos. (De Aguilera M and Mendiz A., 2003).

Como se ha comentado anteriormente, los videojuegos pueden llegar a ser una herramienta muy adecuada para el aprendizaje de contenidos ajenos a la idiosincrasia del mismo. Esto se puede deber a elementos como el aumento de motivación, el feedback inmediato, la intensidad del reto, la capacidad de tomar decisiones o la posibilidad de fallar y volverlo a intentar en un entorno

seguro. El aprendizaje en los videojuegos debe ser continuo y progresivo.

Continuo porque debe de ser constante y bien repartido a lo largo del juego, y progresivo porque el reto y la dificultad debe ir incrementando conforme se avanza en el mismo. Estos principios se deben de mantener de la misma manera en los juegos educativos.

La conocida *Taxonomía de Bloom* o *Taxonomía de objetivos de la educación* (Bloom y Krathwohl, 1956) es un modelo que nos ayuda a jerarquizar por orden de efectividad las diferentes maneras que posee el ser humano de aprender un contenido determinado (Figura 2.5). A lo largo de los años esta ha tenido diversas revisiones, pero algo que podemos observar en todas ellas es cómo el poner en práctica lo aprendido se encuentra siempre en los niveles intermedios en esta jerarquía. Asimismo, el crear se encuentra a la cabeza y el simple memorizar y comprender en la cola. Por ello, si se quiere realizar un juego serio exitoso en su objetivo de maximizar su valor educativo, este debe permitir al usuario pasar por la mayoría de estas fases de forma ascendente y progresiva. Esto se traduce en que lo primero que se debe de hacer es familiarizar al usuario con el tema que se quiera tratar; seguidamente dejar que lo comprenda y aplique de forma adecuada; y finalmente dejar que este cree y se exprese haciendo uso de esos conocimientos aprendidos.

2.2.2. Videojuegos educativos musicales

En el mercado existen diversos videojuegos o aplicaciones interactivas de aprendizaje de piano, tanto gratuitas como de pago. Una que es muy usada por usuarios que ya tiene nociones en piano para aprender una pieza en concreto es la anteriormente mencionada Synthesia, la cual permite a personas no tan familiarizadas con la lectura e interpretación de partituras aprender poco a poco y de forma visualmente intuitiva. Esto se debe al diseño visual que posee la aplicación, que permite al jugador común visualizar la posición de las próximas notas en el piano, su duración y el momento exacto en el que deben de ser pulsadas en un solo vistazo (Figura 2.6). Synthesia permite ser usada con el ratón, un teclado normal, un teclado MIDI e incluso anima al jugador a usar un piano analógico, aunque sin embargo en esta última opción no habrá feedback del juego. También permite importar piezas musicales en forma de archivos MIDI de manera automática, así como modificar y personalizar una gran cantidad de opciones de partida y mecánicas.

Otra aplicación para móviles que tiene como objetivo enseñar piano desde cero es SimplyPiano (JoyTunes, 2017). Este programa combina el uso de vídeos explicativos con ejercicios de práctica en forma de minijuegos y tests donde poner en evaluación las habilidades aprendidas. Posee una amplia variedad de piezas musicales para aprender con el fin motivar a los usuarios, desde música clásica hasta canciones pop actuales. A diferencia de Synthesia, SimplyPiano si que se propone realizar un seguimiento del progreso del

jugador, así como tratar de enseñar y habituar al usuario a leer e interpretar partituras a tiempo real (Figura 2.7). Además, esta aplicación permite usar tanto la pantalla táctil como un piano o teclado regular, ya que es capaz de identificar la nota que está siendo interpretada a través del micrófono. Sin embargo, esta última funcionalidad puede causar problemas si hay ruido que pueda interferir en la habitación. SimplyPiano contiene adicionalmente elementos de motivación extrínseca como son niveles de experiencia o bonificaciones por jugar regularmente.

En líneas generales, podemos diferenciar los dos tipos de enfoques a la hora de realizar una plataforma de enseñanza de piano en las dos aplicaciones mencionadas, *Synthesia* y *SimplyPiano*. La primera tiene un enfoque de "sandbox", ofreciendo al usuario personalizar e interpretar las piezas que desee, pero que sin embargo puede resultar abrumadora y poco conveniente de usar, ya que se debe de poseer o generar de antemano el archivo MIDI de la pieza o ejercicio que se quiera tocar. La segunda posee un plan de estudio específico que el usuario debe de seguir, teniendo la finalidad de poder enseñar los conceptos básicos de piano y teoría musical sin la presencia de un profesor, dejando de lado la capacidad de libre elección de ejercicios, contenidos y piezas a interpretar.

En la Figura 2.8 podemos visualizar una captura de pantalla de *SimplyPiano* Se puede observar su estética simple y sencilla de usar, su manera de dividir los contenidos en distintas unidades y su forma de monetización, que distingue entre contenidos "premium"(de pago) y contenidos gratuitos, los cuales constituyen las 3 primeras lecciones, para que el usuario pueda probar sin tener que realizar un desembolso económico.

Un proyecto de final de grado que expone el desarrollo de una herramienta educativa musical es "Herramientas para el aprendizaje de técnicas de mezcla en producción musical"(Pérez, 2019), en el cual se. Esta herramienta se encarga de enseñar los aspectos elementales de la mezcla musical estéreo.

2.3. Informática musical y programación de audio

La digitalización de la música y el sonido ha evolucionado de manera significativa con el paso de los año. Desde los primeros chips de 8 bits capaces de reproducir música sencilla, conocida ahora como *chiptune*, hasta dispositivos de alta definición como el HD DVD (DVD FORUM, 2005) o el Blue-ray Disc (Sony, Philips, TDK, Microsoft, BDA, 2005) capaces de alcanzar los 96kHz de frecuencia de muestreo.

Gracias a estos avances que permiten que la música digitalizada goce de una calidad que la hace indistinguible al oído humano de la analógica, en la actualidad la gran mayoría de música está grabada, producida y masterizada de manera digital. Programas de producción musical como Abbleton

(Ableton AG, 1999), Fl Studio (Image-Line Software, 2021) o Audacity (Dannenberg y Mazzoni, 2000) están a la orden del día, y son usados por la gran mayoría de personas a la hora de producir música de manera digital. En el mundo de los videojuegos, motores de audio como FMOD (Firelight Technologies, 1995) permiten realizar composiciones dinámicas y tridimensionales.

La informática musical es un tema muy complejo y amplio, del que se requerirían libros enteros para hablar de su historia, estándares, técnicas... etc. Por ello, en esta sección nos vamos a centrar en la idiosincrasia del estándar MIDI, el cual no tiene competencia real en la actualidad como forma de representar música digitalmente, más allá del propio análisis de la onda de audio.

2.3.1. Tecnologías de representación musical computacional

Las tecnologías de representación musical computacional son aquellas que permiten representar partituras musicales de manera digital de forma alternativas a las partituras tradicionales. De esta manara se pueden almacenar piezas musicales digitalmente en archivos que sean mucho más ligeros que un fichero de audio y que, además, sean sencillos de interpretar por distintos tipos de software.

Existen varios protocolos en este campo, como pueden ser la tecnología **OSC** (Open Sound Control), **CV** (Control Voltages). Sin embargo, estos se encuentran obsoletos o casi totalmente en desuso a día de hoy. Por ello, esta sección se centrará en el protocolo MIDI, el cual es el más usado universalmente en la mayoría de software y hardware musical y de audio.

“ **MIDI** (*Musical Instrument Digital Interface*) es el nombre que se le da a uno de los estándares industriales más comunes a la hora de comunicar entre sí instrumentos musicales, equipos informáticos y procesadores de sonido de variada índole. Este estándar define mecanismos para el control de instrumentos, la generación y transmisión de información procedente de estos y su interconexión física con otros instrumentos o equipos informáticos”. (Ibáñez, 2021)

Explicado de forma más concisa, MIDI es el sistema estándar que se utiliza a la hora de digitalizar la entrada de un instrumento físico compatible, como puede ser un teclado MIDI o a la hora de generar partituras que vayan a ser interpretadas por un instrumento virtual.

Este sistema se basa en la transmisión de mensajes de 1, 2 o 4 bytes de tamaño (dependiendo del tipo de mensaje), que contienen información como puede ser el tono de la nota musical interpretada, su intensidad, el instante de su pulsación, su timbre y otras muchas variables que permiten que la modulación del input pueda ser traducida e interpretada de la manera más fiel posible (Figura 2.9). Esto permite que un archivo MIDI tenga un peso muy reducido en comparación a un archivo de audio MP3 o WAV, debido a

que no almacena ondas de sonido ya interpretadas.

Como en la mayoría de archivos en binario, los archivos MIDI contienen una serie de cabeceras y protocolos de que hay que seguir a la hora de leer o escribir en ellos. En este caso, MIDI se estructura en pistas o *tracks*.Este sistema en pistas permite una división más lógica de los distintos elementos de la pieza, mientras que el sistema de canales o *channels* es utilizado para poder reproducir cada uno de los distintos instrumentos o efectos de forma simultánea. Una peculiaridad de los archivos MIDI es que, como en la mayoría de casos los mensajes poseen un único byte, si se quieren representar valores mayores a 255 se utiliza la siguiente estrategia: Si el primer bit del mensaje es 0, significa que el byte en el que se encuentra contenido es el final de un valor numérico mientras que si este bit es igual a 1, el valor continúa en el siguiente byte. De esta manera la transmisión de información entre el instrumento MIDI y el ordenador puede ser mucho más ágil que enviando mensajes de un mayor tamaño.

Gracias a la tecnología MIDI, es posible traducir la naturaleza de la nota interpretada de manera significativamente más precisa y eficiente que teniendo que analizar la onda de sonido digitalizada de manera algorítmica.

2.3.2. Herramientas VST

Como se ha visto, el estándard MIDI es muy útil a la hora de guardar y comunicar información musical. Sin embargo, a la hora de reproducirlo, se necesita otra herramienta que nos permita transformar estos datos en una onda de sonido reproducible. Para ello, se puede hacer uso de sintetizadores o, en el caso de estar utilizando un equipo informático, un VST.

VST (*Virtual Studio Technology*) es un software de audio en forma de plug-in que integra sintetizadores, interfaces y efectos de audio virtuales. Existen varios tipos de VSTs, como pueden ser los VSTi, que emulan uno o varios instrumentos, o los VSTfx, que se usan para añadir efectos de sonido (Steinberg Technologies, 1984). Los VST pueden emular el sonido de un instrumento o efecto determinado a partir de archivos de audio, generándolos de forma completamente artificial o combinando ambos métodos.

Existe una gran cantidad de VSTIs de todo tipo, que suelen ser compatibles con la gran mayoría de software de producción, edición y reproducción de audio, tanto para música regular como música para videojuegos. Estos plugins suelen incluir una interfaz en la que es posible modificar multitud de parámetros del instrumento virtual como puede ser el *attack*, el *decay* o el tono, entre otros (Figura 2.10).

2.3.3. MIDI y VST en la actualidad

Debido a las ventajas expuestas anteriormente que el sistema MIDI nos proporciona, este se encuentra de manera omnipresente en el mundo de la

producción y la informática musical actualmente.

Existen en el mercado una gran cantidad de tipos de controladores MIDI, que pueden adoptar la forma de un teclado, un set de PADS, una batería, una guitarra o incluso un instrumento de viento. Gracias a ellos, se hace posible grabar o interpretar en directo sin tener que hacer uso de instrumentos tradicionales. Eso conlleva un gran número de posibles ventajas. Por ejemplo, gracias al uso combinado de un instrumento MIDI y una variedad de plug-ins VSTi, permite cambiar el timbre del instrumento interpretado tanto a tiempo real como en postproducción de manera sencilla e instantánea, habilitando a un músico que posea únicamente un teclado MIDI interpretar cualquier instrumento que desee, además de que no deberá cargar con equipo tan aparatoso como puede ser un piano o un set completo de batería (Figura 2.11).

Otro uso bastante habitual de esta tecnología es el de hacer uso de las tablatauras MIDI para componer de manera intuitiva para cualquier instrumento, sin necesidad de saber tocar el instrumento en cuestión o de saber escribir y leer partituras (Figura 2.12).

La adopción del estándard MIDI también ha sido clave en la historia de la música y el sonido de videojuegos. Incluso antes de la aparición de las primeras consolas de 8 bits como la Nintendo Entretainment System (NES) o los primeros ordenadores para jugar como la Commodore 64, el uso del sistema MIDI ha permitido a dispositivos con muy poca memoria y chips de audio de 8 bits la capacidad de almacenar y reproducir música adaptativa y sonidos medianamente complejos .

Pero el MIDI no sólo ha sido utilizado para juegos del siglo pasado. A pesar de que las mayores capacidades en consolas y PC han permitido estandarizar la inclusión de grandilocuentes bandas sonoras orquestales, incluso juegos más actuales tanto musicales como el Guitar Hero 3 (Neversoft, 2007), como no musicales que hacen uso de música generada proceduralmente como Spore (Maxis, 2008) hacen uso de este sistema (Collins, 2009).

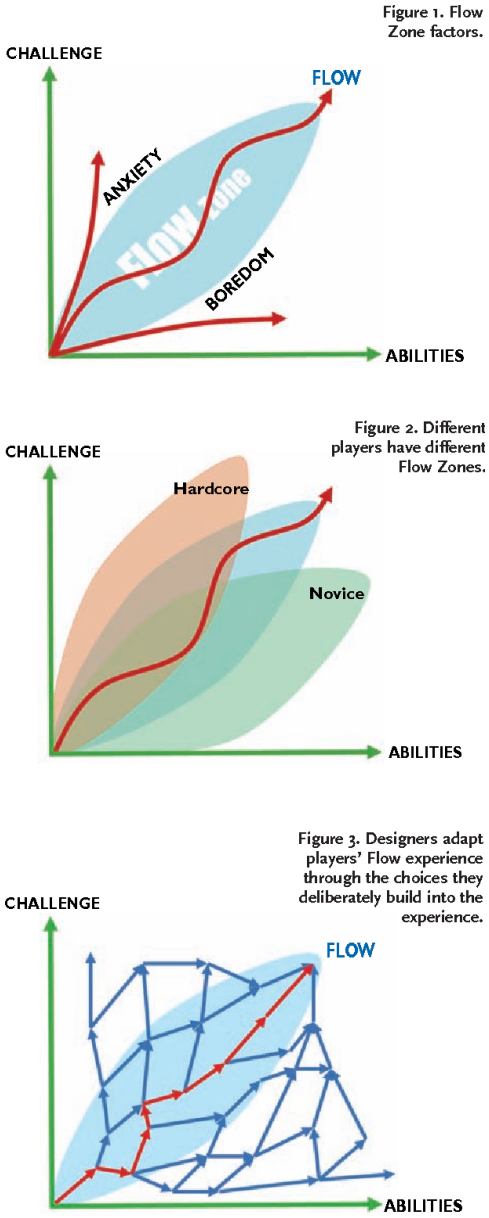


Figura 2.4: Conjunto de gráficas que muestran el concepto de flow y curva de dificultad. (Chen, 2007).



Figura 2.5: Diagrama que sintetiza la Taxonomía de Bloom, revisada por (Bloom y Krathwohl, 2001). Las categorías que se encuentran en la parte superior son aquellas que indican una mayor comprensión del contenido aprendido.

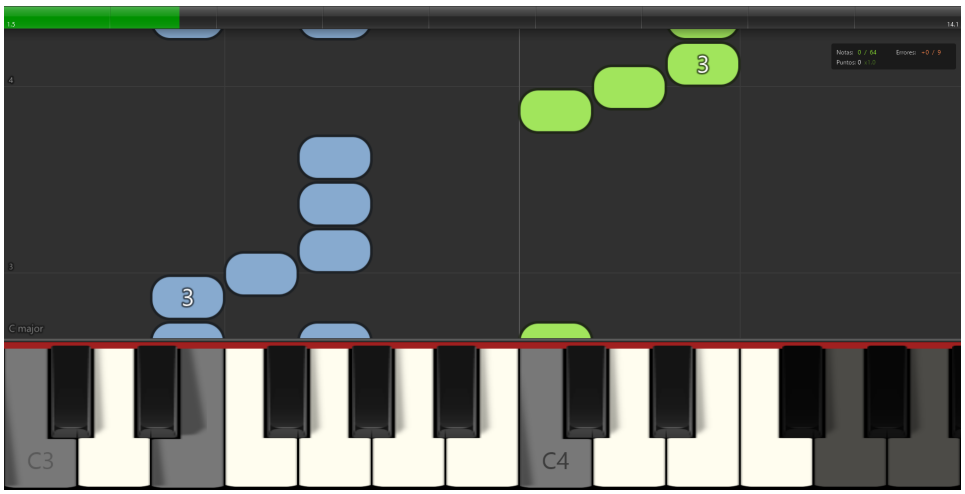


Figura 2.6: Captura de pantalla de una partida para principiantes en Synthesia (Synthesia LLC, 2006), en el modo de juego de ambas manos, haciendo uso de un teclado MIDI.

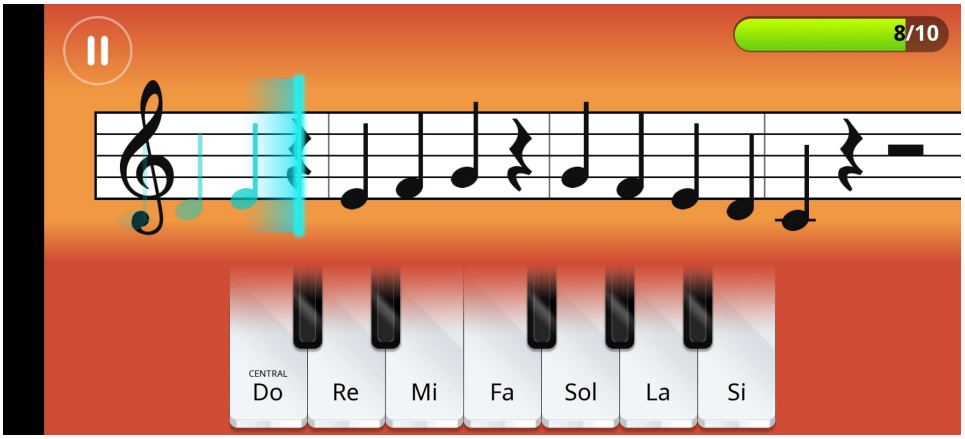


Figura 2.7: Captura de pantalla de nivel de práctica de interpretación de partitura en tiempo real. Se puede observar cómo la línea azul se va desplazando hacia la derecha, indicando en que momento debe de ser presionada cada nota.



Figura 2.8: Captura de pantalla de nivel del menú principal de SimplyPiano.

Table 2: Expanded Status Bytes List			
STATUS BYTE		DATA BYTES	
1st Byte Value Binary |Hex| Dec	Function	2nd Byte	3rd Byte
10000000= 80= 128	Chan 1 Note off	Note Number (0-127)	Note Velocity (0-127)
10000001= 81= 129	Chan 2 Note off	Note Number (0-127)	Note Velocity (0-127)
10000010= 82= 130	Chan 3 Note off	Note Number (0-127)	Note Velocity (0-127)
10000011= 83= 131	Chan 4 Note off	Note Number (0-127)	Note Velocity (0-127)
10000100= 84= 132	Chan 5 Note off	Note Number (0-127)	Note Velocity (0-127)
10000101= 85= 133	Chan 6 Note off	Note Number (0-127)	Note Velocity (0-127)
10000110= 86= 134	Chan 7 Note off	Note Number (0-127)	Note Velocity (0-127)
10000111= 87= 135	Chan 8 Note off	Note Number (0-127)	Note Velocity (0-127)
10001000= 88= 136	Chan 9 Note off	Note Number (0-127)	Note Velocity (0-127)
10001001= 89= 137	Chan 10 Note off	Note Number (0-127)	Note Velocity (0-127)
10001010= 8A= 138	Chan 11 Note off	Note Number (0-127)	Note Velocity (0-127)

Figura 2.9: Fragmento de tabla de interpretación de los distintos bytes que componen un mensaje MIDI (MIDI Association, 2021).



Figura 2.10: Interfaz de Guitar Rig 6 Demo (Native Instruments, 2020), VSTi que simula una guitarra eléctrica, el cual posee multitud de filtros, efectos, distorsiones y amplificadores para modificar su sonido.

Number	Class	Instrument
0	Piano	Acoustic Grand Piano
1	Piano	Bright Acoustic Piano
2	Piano	Electric Grand Piano
3	Piano	Honky-tonk Piano
4	Piano	Rhodes Piano
5	Piano	Chorused Piano
6	Piano	Harpsichord
7	Piano	Clavinet
8	Chromatic Percussion	Celesta
9	Chromatic Percussion	Glockenspiel
10	Chromatic Percussion	Music box
11	Chromatic Percussion	Vibraphone
12	Chromatic Percussion	Marimba
13	Chromatic Percussion	Xylophone
14	Chromatic Percussion	Tubular Bells
15	Chromatic Percussion	Dulcimer
16	Organ	Hammond Organ
17	Organ	Percussive Organ
18	Organ	Rock Organ
19	Organ	Church Organ
20	Organ	Reed Organ
21	Organ	Accordion
22	Organ	Harmonica

Figura 2.11: Fragmento de tabla de los distintos identificadores de instrumentos posibles con MIDI



Figura 2.12: Tablatura MIDI en el programa de producción musical FL Studio (Image-Line Software, 2021).

Capítulo 3

Objetivos y especificación

El propósito de este proyecto, como ya está reflejado en el Capítulo 1, es desarrollar un videojuego educativo que permita a los profesores de piano conectar con sus estudiantes de forma remota, de manera que unos puedan enviar ejercicios personalizados a sus alumnos y los otros practicarlos cómodamente, consiguiendo resultados de manera sencilla y amena. La aplicación a desarrollar debe resultar sencilla e intuitiva para sus usuarios, brindar una gran versatilidad al profesor para que pueda elaborar los ejercicios que crea más convenientes, manteniendo la accesibilidad y la motivación del alumno, promoviendo su mejora personal.

Para ello, una vez conocido cuál es el estado de la cuestión, este trabajo se propone una serie de objetivos más concretos que se detallan en este capítulo, además de entrar a especificar las funcionalidades que debe poseer la aplicación, así como sus limitaciones.

3.1. Objetivos propuestos

Los principales objetivos de este proyecto son los siguientes:

- Identificar las oportunidades que ofrecen las nuevas tecnologías y la ludificación para mejorar la experiencia docente tanto de los profesores como de los alumnos en enseñanza elemental de piano.

- Diseñar un sistema que transforme cualquier pieza musical en contenido audiovisual interactivo, disfrutable de manera intuitiva por un amplio espectro de jugadores realizando la digitación sobre un teclado de piano.

- Desarrollar el prototipo de un videojuego educativo para ordenadores personales que aproveche el sistema anterior de manera que los profesores, por un lado, puedan crear ejercicios personalizados y los alumnos,

por otro, practicar repetidamente con ellos y ser evaluados de manera automática.

- Validar la utilidad y la usabilidad de este prototipo con usuarios reales, para poder plantear un proceso completo de producción. La aplicación debería ser conveniente e intuitiva para profesores y motivadora y entretenida para los alumnos. Pudiendo, de esta manera, llegar a ser un producto viable de cara al futuro.

3.2. Especificación

En esta sección se desarrollan más en detalle las características y requisitos funcionales que cumplirá la herramienta, así como sus limitaciones.

3.2.1. Requisitos y funcionalidades

En esta subsección se listarán todos los requisitos necesarios que poseerá la herramienta para que cumpla con las funciones educativas requeridas, así como los elementos que brindarán a la aplicación un cierto atractivo y facilidad de uso para sus potenciales usuarios. El diagrama general con el flujo de uso de la aplicación aparece en la Figura 3.1.

Se puede distinguir entre requisitos funcionales y requisitos no funcionales.

Los requisitos funcionales son:

- **Reconocimiento de input de controlador MIDI** en tiempo real. Esta funcionalidad es básica para el funcionamiento de la aplicación, ya que tanto para la grabación como para la realización de los ejercicios se hará uso de un teclado virtual, el cual será controlado por el teclado digital del usuario.

- **Instrumento Virtual o VST**. Para el funcionamiento del piano virtual, es necesario la inclusión de un VST que gestione el sonido de cada una de las teclas, para que suene lo más realista posible. En este caso, con usar una única librería de audio de notas de piano libre de derechos es suficiente, ya que variables cómo su intensidad pueden ser modificadas por código en Unity.

- **Sistema jugable**, es decir, la implementación de las mecánicas y elementos necesarios de *gameplay* para una experiencia de juego satisfactoria por parte de los jugadores. Para ello, este proyecto se basará en el funcionamiento de juegos anteriormente mencionados como *Synthesia*. Este punto no solo incluye las interacciones entre el jugador y el

entorno virtual, sino también todos aquellos elementos como animaciones, efectos de sonido o efectos visuales que mejoran la experiencia de jugador.

- **Estructura de menús y UI** que permitan a los usuarios navegar por la aplicación y elegir los distintos parámetros de los ejercicios de manera intuitiva y cómoda. Con los parámetros de los ejercicios se hace referencia a opciones como las octavas a usar o la velocidad de reproducción del nivel, así como opciones de volumen y usabilidad.

- ***Assets***. Este punto hace referencia a todos los elementos visuales, sonoros que posee la aplicación, así como una batería de archivos MIDI previamente creados, que corresponden a piezas comúnmente interpretadas por alumnos de piano principiantes. De esta manera, los usuarios tendrán piezas que interpretar si quieren ir más allá de lo mandado por el profesor. Los elementos visuales serán todos diseñados y creados por el autor de este proyecto para evitar posibles problemas con su propiedad intelectual. Los elementos sonoros

Los requisitos no funcionales son:

- **Lectura de archivos MIDI**. Para poder convertir una pieza musical en formato MIDI a un ejercicio o nivel que pueda ser jugado en la aplicación, es necesario construir un script que sea capaz de traducir secuencialmente un archivo MIDI a valores de tiempo, tono y espacio que se puedan usar lúdicamente.

- **Escritura de archivos MIDI**. Es necesario poder convertir una pieza musical interpretada en la herramienta a un archivo MIDI, para que esta pueda ser escuchada en una aplicación externa, como posteriormente jugada en la propia herramienta. Para ello se debe traducir todos esos *inputs* a lo largo del tiempo a binario y *little endian*, para posteriormente adaptarlos a los formatos exigidos por el sistema MIDI.

- **Sistema de dinámicas de juego**. Esto se refiere principalmente a elementos como son el sistema de puntuación y evaluación del rendimiento del jugador tras realizar un ejercicio; u elementos como los puntos de experiencia o niveles que el usuario va ganando a medida que completa y mejora su puntuación en los ejercicios. Estos elementos son súmamente importantes, no sólo porque permiten al profesor acceder a datos y estadísticas de sus alumnos, sino porque le brinda a los alumnos un *feedback* inmediato y una sensación de progresión y mejora en sus habilidades, lo que es vital para mantener su motivación a lo largo del tiempo.

- **Corrección de bugs y rendimiento**. Es necesario que, tras el desarrollo de la herramienta, exista una fase de testeo en la cual tanto el propio autor como usuarios ajenos al desarrollo se encarguen de señalar errores, tanto de diseño como de ejecución. Estos errores serán corregidos posteriormente para asegurar el buen funcionamiento de la herramienta. Adicionalmente, Piano Unite debe ser capaz de funcionar a 60 fotogramas por segundo en la mayoría de ordenadores para poder ser usado de manera óptima, por lo que durante el desarrollo se deberá optimizar lo máximo posible.

3.2.2. Limitaciones

Existe una principal limitación, que por el carácter unipersonal del equipo de desarrollo del proyecto no es posible su implementación, y es la de añadir la funcionalidad de poder *loggear* y conectar a profesor y alumno vía online. De esta manera, el docente podría gestionar a todos sus alumnos, mandarles ejercicios y ver sus resultados a través de la herramienta, sin necesidad de recurrir a herramientas de mensajería externas como puede ser el correo electrónico. De esta manera los datos de cada alumno como su nivel o sus puntuaciones quedarían registradas en su usuario en un servidor remoto, por lo que no tendrían la posibilidad de alterarlos ilícitamente. Lo mismo pasaría con los archivos de cada uno de los ejercicios, que podrían ser compartidos con mayor facilidad.

Otra limitación que hay que asumir es la de la necesidad por parte del usuario de poseer un teclado MIDI para poder hacer uso correcto de la aplicación. En un comienzo se contempló la posibilidad de poder usar la aplicación en un piano analógico, recibiendo el input desde un micrófono y analizando la frecuencia de sonido para identificar la nota pulsada. Sin embargo, estos sistemas de reconocimiento dependen de una gran cantidad de factores, como son la calidad del micro, el ruido de ambiente o la cantidad de notas tocadas simultáneamente, entre otras.

La última limitación que cabe mencionar es la de limitar la plataforma de uso de la aplicación al sistema operativo de Windows 10. Aunque el motor de Unity de la opción de exportar de manera automática a sistemas como Mac o algunas versiones de Linux, al no poseer ningún equipo con estos sistemas se hace imposible su correcto testeo.

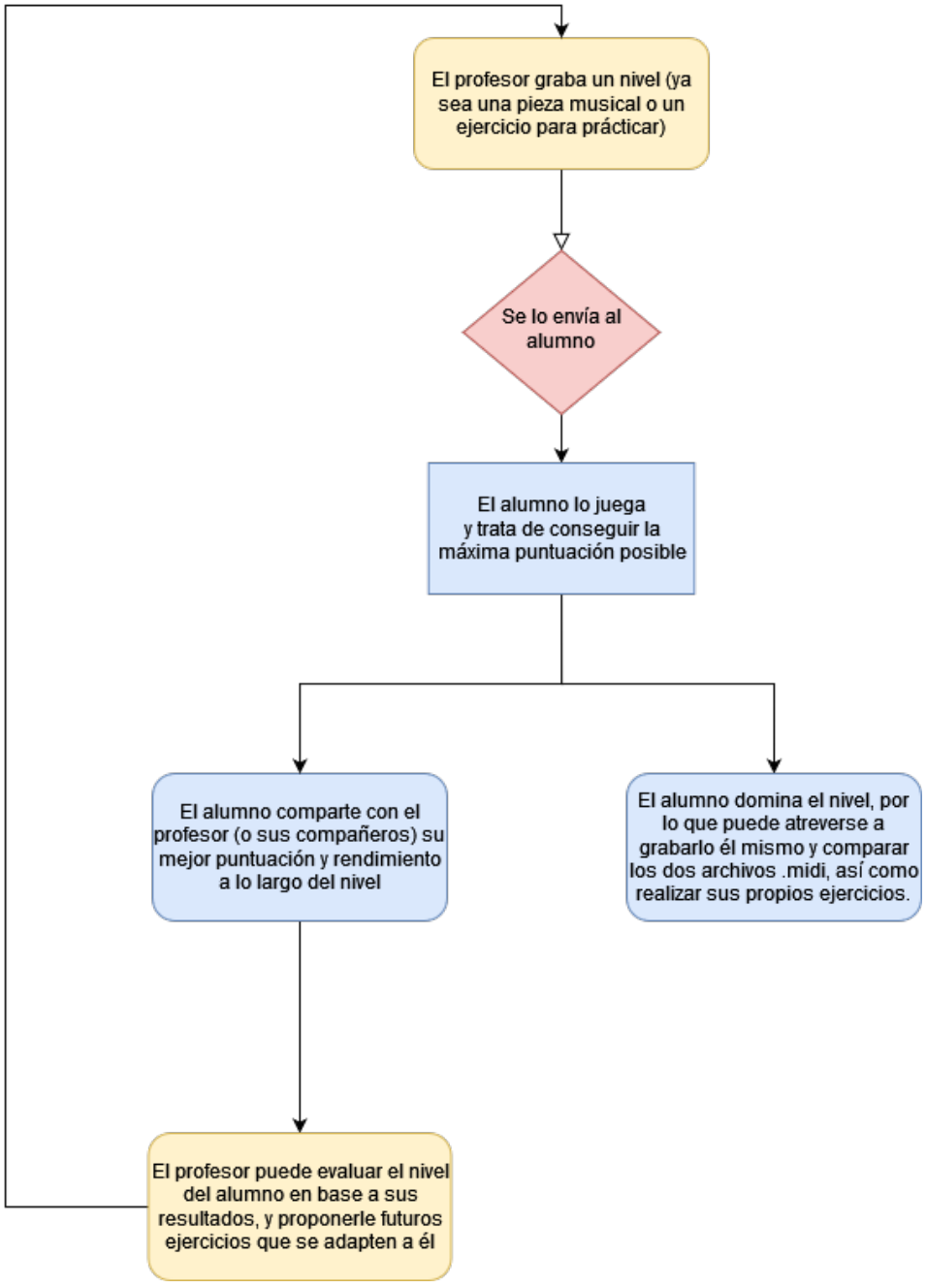


Figura 3.1: Diagrama general que describe el flujo de uso de la aplicación.

Capítulo 4

Metodología y herramientas

En este capitulo se enumeran los recursos y las tecnologías que se usan en el desarrollo del proyecto, así como un breve resumen de su funcionamiento. Además, se comentarán en detalle las distintas fases que ha tenido el proyecto.

4.1. Metodología de desarrollo

El desarrollo del proyecto se dividirá en 3 fases diferenciadas: fase de pre-producción, fase de producción y fase de post-producción.

Durante las tres fases de desarrollo del proyecto, se realizaron reuniones periódicas con el director de este trabajo. Estas reuniones se realizaban cada 2 semanas, posponiendo únicamente aquellas fechas que coinciden con festividades. Estas reuniones cumplen 3 funciones diferenciadas. La primera es que funcionan como forma de organización, ya que en cada una de ellas se enumeran las prioridades en el desarrollo y el orden en el que las distintas tareas restantes deben ser realizadas. La segunda es que funcionan como hitos, ya que para cada reunión se debe de presentar la última versión del proyecto con las funcionalidades que se decidieron implementar en la reunión previa. La tercera es como forma de poder mantener actualizado del progreso al director, así como poder resolver todas las dudas surgidas a lo largo de las últimas semanas.

Además, el equipo de Narratech (Narratech Laboratories) realizó una serie de reuniones en Diciembre y en Abril, donde distintos autores de TFGs y TFMs podían realizar presentaciones de su proyecto y su progreso en el mismo, para así recibir *feedback* sobre el mismo y poder mejorarlo.

4.1.1. Preproducción

La fase de preproducción consta de las siguientes tareas, en orden cronológico:

- **Concepto**: Primeramente, se desarrolló el concepto del proyecto, se puso en común con el director del proyecto y se discutieron aspectos como el enfoque que se debería dar al proyecto, el *target* al que iría dirigido, las funcionalidades que debía poseer o la forma en la que se aplicarían los conocimientos y habilidades aprendidos a lo largo del grado.

  En el comienzo, la única idea que se tenía era la de realizar un proyecto que tuviese relación con la informática musical, rama en la que el autor está especialmente interesado. Tras una lluvia de ideas y tratar de concretar más el tema del proyecto, se acabó optando por la opción de realizar algún tipo de aplicación o juego serio de temática musical. Además, se pensó que sería buena idea trabajar con el sistema MIDI, cuya forma más común de uso es a través de un teclado MIDI.

  A partir de esa premisa el problema a resolver se presentó sólo pues el autor, que llevaba meses tratando de aprender a tocar piano de forma autodidacta, se dio cuenta de que existen ciertos problemas con los métodos de enseñanza on-line. Por un lado, los videotutoriales y clases online pre-grabadas tienen un problema de interactividad y feedback acerca de la mejora o posibles errores que el alumno esté cometiendo. Por el otro, las aplicaciones de educación de piano aplican un método que da por hecho que el alumno carece por completo de conocimientos sobre teoría musical o lectura de partituras, por lo que debe invertir bastante tiempo en conocimientos que ya posee.

  Por ello, el objetivo del trabajo se acabó concretando en la construcción de una herramienta que ayude o facilite el aprendizaje de piano, especialmente para los principiantes. Se decidió así ya que dirigirla para un público de nivel avanzado carecería de utilidad, pues ellos no necesitan de tanta ayuda ni motivación externa para seguir mejorando, además de que requeriría un nivel bastante alto en el instrumento por parte del desarrollador para poder cubrir sus necesidades.

  Con todos los conceptos anteriores sobre la mesa, fue entonces el momento de comenzar la investigación más en profundidad.

- **Investigación**: Como se desarrolla más extensamente en el capítulo de Estado de la Cuestión (Capítulo 2), la investigación se basó en tres pilares fundamentales: La enseñanza elemental de piano, juegos serios (y juegos educativos musicales) y tecnología MIDI.

  La investigación en el tema de enseñanza elemental de piano permitió poder descubrir cuales son las necesidades tanto de los profesores como de los alumnos a la hora de introducir nuevas tecnologías en el programa docente. Además, permitió tener claras las habilidades y contenidos que se imparten transversalmente a los distintos métodos, para así poder crear una aplicación que se pueda adaptar a todos ellos. Las

conclusiones fueron que, especialmente con los alumnos más jóvenes, uno de los problemas es mantenerlos motivados a la hora de realizar deberes y ejercicios en casa, fuera de horario de clases. Además, la correcta digitación e interiorización de patrones melódicos y escalas es una habilidad que se debe trabajar siempre en las primeras etapas del aprendizaje de un instrumento, por lo que la aplicación debería poder servir para trabajar precisamente estos conceptos. Por último, se puede sacar en claro también que el mundo de la enseñanza musical tradicional se resiste todavía a trabajar con nuevas tecnologías de apoyo, avances que en un mundo post-pandemia se están convirtiendo en algo inevitable, por lo que existe una ventana de oportunidad muy grande en este sector.

Profundizar en el tema de los juegos serios aportó las claves y elementos que un juego educativo debe de tener para ser exitoso en su cometido didáctico, así como para mantener motivados a los jugadores. Sin embargo, lo más útil de esta parte de la investigación es la relacionada con el análisis de otros juegos educativos musicales, ya que muchas de las ideas y mecánicas del proyecto como la representación visual de los eventos MIDI o la forma y distribución de los menús están inspiradas en aplicaciones ya mencionadas como *Synthesia* o *SimplyPiano*.

Por último, toda la investigación relacionada con las tecnologías MIDI, su historia y su uso ha sido vital para poder trabajar con el sistema, ya que para la realización de ciertas funcionalidades como la lectura o escritura de archivos MIDI hay que conocer detalladamente todas sus características y protocolos.

- **Diseño**: Los primeros diseños de la herramienta se enfocaron en desarrollar el bucle de funcionamiento de la herramienta (Figura 3.1). También se concretó el diseño de *gameplay* inspirado en la aplicación *Synthesia*. Seguidamente se decidieron las diferentes *pantallas* o escenas de *Unity* con las que contaría la herramienta: una para la realización del ejercicio, otra para la creación de los ejercicios, otra para la visualización de las puntuaciones de cada nivel, el menú principal y una última que aparezca al final del nivel para indicar la puntuación conseguida. Por último, se tomó la decisión de que la aplicación debería contar de un sistema de niveles y experiencia, ya que es una forma ideal de *gamificar* la herramienta e incrementar la motivación del alumno a lo largo del tiempo.

  Este diseño realizado estaría sujeto a posibles cambios y modificaciones a medida que se desarrollara el proyecto.

- **Redacción**: Antes de comenzar el desarrollo, se decidió completar la escritura de los capítulos de introducción (Capítulo 1) y estado de la cuestión (Capítulo 2), ya que ya se habían investigado todos los

conocimientos teóricos y técnicos que serán posteriormente aplicados en el proyecto, así como el diseño general de la aplicación.

- **Prototipo**: Tras la redacción de los primeros capítulos, un primer prototipo en *Unity* fue desarrollado. Este prototipo consistió en la realización de un piano virtual que pudiese ser controlado a través de un teclado MIDI. Para ello se implementó al proyecto la librería de *MidiJack* y la librería de sonidos de piano en formato WAV con licencia *Creative Commons* extraída de *FreePats* (Free Pats Project, 2017). Esto demostró que se podía realizar un piano virtual que registrase y respondiese perfectamente, dando la ilusión de estar tocando un piano físico. Además, también se añadió la funcionalidad de que este piano también pudiera registrar la velocidad con la que se pulsa la tecla, y ajustar el volumen de reproducción de la nota en función de ella.
- **Planificación**: Una vez implementado el piano virtual, el orden en que se irán desarrollando el resto de funcionalidades será el siguiente:
  - Sistema de lectura y traducción de MIDI a elementos visuales.
  - Sistema de grabación y exportación de MIDI a tiempo real.
  - Programación de jugabilidad de la escena de realización de ejercicios.
  - Escena de final de nivel.
  - Escena de menú principal.
  - Escena de visualización de resultados.
  - Equilibrado de gameplay.
  - Sistema de guardado de partida y opciones.
  - Sistema de experiencia y niveles.
  - Permitir al usuario modificar ajustes de sonido.
  - Cambio de *placeholders* por *assets* definitivos. Correcciones de color, tamaño y resolución de pantalla.
  - Añadir efectos de partículas y animaciones, para mejorar el *gamefeel* y experiencia de usuario.
  - Corrección de errores y limpieza de código.
  - Lanzamiento de una demo y la creación de un cuestionario para realizar las pruebas con usuarios, y poder recopilar *feedback*.
  - Últimos retoques basados en el *feedback* recibido
  - Publicación de la herramienta en (`https://narratech.itch.io/piano-unite`).

La fase de preproducción comenzó el 1 de Octubre de 2021 y finalizó el 20 de Diciembre de 2021.

4.1.2. Producción

La fase de producción consta de las siguientes tareas, en orden cronológico:

- **Desarrollo**: El paso a paso del desarrollo de la aplicación se explicará con detalle en el capítulo de Desarrollo (Capítulo 5).
- **Redacción**: Simultáneamente al desarrollo, se escribirán los capítulos de Metodología y Herramientas(Capítulo 4), Objetivos(Capítulo 3) y Desarrollo (Capítulo 5).
- **Preparación de las pruebas de usuario**: En este punto, se decidió sacar dos versiones distintas de la aplicación, una para profesores y otra para alumnos, para de esta manera clarificar mejor el propósito y funcionamiento de la herramienta. Ambas poseen las mismas funcionalidades, pero están presentadas de formas distintas para que sean más claras y fáciles de usar. Para la preparación de las pruebas de usuario se hará uso de *Google Drive* y *Google Forms* (Google Drive, 2012). En *Google Drive* se subirá un archivo comprimido .zip con ambas versiones, cuyo link se encontrará en el cuestionario realizado en *Google Forms*. La estructura del cuestionario se especifica en el capítulo de Tests y Resultados (Capítulo 6), aunque cabe mencionar que parte de el mismo está desarrollado basado en los cuestionarios de usabilidad tipo SUS (*System Usability Scale*).

La fase de producción comenzó el 1 de Febrero de 2022 y finalizó el 30 de Abril de 2022.

4.1.3. Postproducción

La fase de postproducción consta de las siguientes tareas, en orden cronológico:

- **Testeo y corrección de errores**: En esta fase se testeará el juego y se realizarán los cambios pertinentes basados en el feedback recibido por los usuarios. Una vez finalizada la herramienta, será publicada. Se puede acceder a ella en el siguiente link:
- **Resultados**: Tras la finalización del desarrollo de *Piano Unite*, se escribirán los capítulos de Tests y Resultados (Capítulo 6) y Conclusiones (Capítulo 7).
- **Finalización**: Se presentará el primer borrador al director este trabajo. Tras su lectura y análisis, se contará con unas semanas adicionales para realizar correcciones y mejorar el documento, así como preparar la presentación ante el juzgado.

La fase de producción comenzó el 1 de Mayo de 2022 y finalizó el 30 de Mayo de 2022.

4.2. Herramientas utilizadas

Para la creación de este proyecto se ha hecho uso de diferentes herramientas, aplicaciones y librerías. A continuación, se listarán y comentarán brevemente:

4.2.1. Unity

Unity (Unity Technologies, 2005) Versión: 2019.4.16f1 . Unity es un motor de videojuegos tanto 2D como 3D cuyo uso es gratuito para la realización de proyectos sin fines comerciales. Posee una manera de trabajar de programación basada en componentes. Se hace uso de este motor en concreto debido a la gran cantidad de plugins y assets que posee su *Asset Store*, su utilización del lenguaje *C#*, la calidad de su documentación, su muy amplia comunidad online que facilita la resolución de dudas en foros y la familiaridad que tiene el autor con el mismo.

4.2.2. MidiJack

MidiJack (Keijiro Takahashi, 2013-2015). Midi Jack es una librería en C# que permite detectar *input* MIDI en tiempo real. Además, está adaptada para poder ser usada en Unity y que se sincronice con los FPS (*Frame Per Second*) con los que se esté jugando.

4.2.3. GitHub

GitHub (Microsoft, 2008). GitHub es una aplicación de control de versiones que hace uso del sistema Git. Aunque el proyecto sea desarrollado por una única persona, es útil utilizar un sistema de control de versiones para poder guardar el proyecto remotamente, así como poder mantener un registro de los avances día a día. El proyecto puede encontrarse en el siguiente enlace: `https://github.com/ErGuille33/ProjectPianoHero`.

4.2.4. Midi Editor

Midi Editor (Markus Schwenk, 2019). Midi Editor es una aplicación gratuita que permite crear archivos MIDI, ya sea grabándolos en tiempo real o editándolos nota a nota. Esta aplicación será de grán utilidad

para la creación archivos MIDI usados en el testeo de los sistemas de lectura y escritura MIDI, o para la realización de archivos MIDI que el autor no sea capaz de tocar.

4.2.5. Visual Studio 2019

Visual Studio 2019 (Microsoft, 2019). Entorno de desarrollo integrado. Además, posee compatibilidad con Unity, lo que permite administrar el proyecto y debuggear de forma más cómoda.

4.2.6. Adobe Photoshop

Adobe Photoshop (Adobe Systems Incorporated, 1990). Programa utilizado para el diseño y creación de los elementos visuales y elementos de la interfaz de usuario.

4.2.7. Slack, Google Meet y Google Drive

Slack (Slack Technologies, 2013), **Google Meet** (Google Meet, 2019) y **Google Drive** (Google Drive, 2012). Estas herramientas web permitieron la correcta comunicación con el director del trabajo de fin de grado.

4.2.8. Count Down

Count Down (Easy Game Studio, 2018). Count Down es una herramienta de pago publicada en la *Unity Asset Store* que ofrece un script que facilita la implementación de una cuenta atrás de 3 a 0. Los *assets* visuales han sido reemplazados, pero los sonoros han sido utilizados. Esta herramienta se utilizó para reducir ligeramente el tiempo de desarrollo.

4.2.9. Runetime File Browser

Runetime File Browser (yasirkula, 2018). Runetime File Browser es una herramienta gratuita publicada en la *Unity Asset Store* que ofrece un explorador de archivos funcional tanto en Windows, MacOsX y Android al que se puede acceder a través de una API propia.

4.2.10. Controlador MIDI Teclado Alesis VI25

Controlador MIDI Teclado Alesis VI25. Teclado MIDI de 2 octavas, con detección de velocidad de pulsación. Esta es la marca y el modelo del teclado MIDI usado por el autor en el desarrollo de la aplicación. Enlace a la página oficial del fabricante: `https://www.alesis.com/products/legacy/vi25`

4.2.11. OBS Studio

OBS Studio (OBS Project, 2012). Programa que permite grabar la pantalla del ordenador, para de esta manera poder realizar los vídeos y capturas de pantalla de la reamienta para su presentación.

A este proyecto en su totalidad le he dedicado alrededor de 400 horas, en un periodo de 8 meses.

Capítulo 5

Análisis, diseño e implementación de Piano Unite

5.1. Análisis y diseño

En esta sección se detallará el proceso de diseño de la herramienta y su evolución en el tiempo. Esto incluye tanto el diseño gráfico y visual de la aplicación, como el diseño de todas sus funcionalidades, el diseño de jugabilidad y el diseño de cara a promover la forma en la que se espera que el usuario haga uso de *Piano Unite*.

5.1.1. Primeros bocetos

En un primer momento, el concepto de la aplicación era el siguiente: un juego serio que, a través de la tecnología MIDI, permita a los usuarios aprender a tocar el piano en los niveles más básicos.

Sin embargo, para desarrollar una aplicación con esas características sería necesario tener un conocimiento sobre el instrumento, teoría musical y enseñanza musical que el autor no posee, y le llevaría años aprender. Además, ya existen en el mercado varias herramientas que cubren esa necesidad y que, adicionalmente, cuentan con excelsos equipos de desarrolladores y profesionales de la música involucrados en su desarrollo, por lo que sería bastante complicado poder llegar a competir a su nivel.

Debido a ello, se decidió cambiar el enfoque al siguiente : un juego serio que permita a profesor y alumno conectar de forma telemática, de manera que el profesor pueda crear ejercicios para que el alumno realice en su casa. El alumno debe sentir como si estuviese jugando a un videojuego, para que de esta manera aumente su motivación por mejorar en el instrumento.

Además, esta aplicación debería enfocarse en que cualquier tipo de usuario pueda comprenderla y usarla, desde un niño a partir de 7/8 años, hasta un adulto que quiera empezar a aprender a tocar el instrumento.

Antes de comenzar los primeros diseños de la aplicación, se decidió que la aplicación debía cumplir con los siguientes enunciados:

- Como profesor, puedo crear ejercicios en formato MIDI para mis alumnos para que practiquen por su cuenta.
- Como alumno, puedo realizar los ejercicios desde mi casa y compartir mis resultados con el profesor.
- Como usuario, puedo utilizar mi controlador MIDI para usar esta aplicación.
- Como alumno, puedo importar archivos MIDI y jugarlos en la aplicación.
- Como usuario, puedo ver un historial de todas los niveles completados, y las estadísticas del mismo.
- Como usuario, debo de poder ser consciente acerca mi progreso.

A estas alturas en el diseño, no se tenía claro exactamente cómo sería la jugabilidad de la aplicación, ni qué tipo de competencias debería abarcar. Para ello primero habría que realizar la investigación acerca de las necesidades de los docentes y alumnos, así como identificar otras herramientas existentes de propósito similar.

5.1.2. Prototipo

Una vez realizada la investigación en el estado de la cuestión, se comenzó el desarrollo de el primer prototipo a presentar.

Con los enunciados anteriores en mente, el paso siguiente consistió en pensar en el tipo de jugabilidad que debía de tener la aplicación. Para ello, se decidió inspirarse en la de la aplicación de *Synthesia*, pero añadiéndole notas que habría que mantener pulsadas durante un tiempo determinado [1]. Este tipo de jugabilidad, también usado en otros juegos de ritmo anteriormente mencionados, consiste en lo siguiente:

El jugador controlará un piano virtual que se encuentra en la parte inferior de la pantalla. Mientras, irán cayendo una serie de objetos verticalmente. Estos objetos representan las distintas notas de la canción, y cada nota está alineada con su correspondiente tecla en el piano. El jugador deberá pulsar la tecla correspondiente del teclado en el momento preciso para ganar puntos. Cuanto más preciso sea el jugador, más puntuación ganará. Si presiona la tecla equivocada, o si la pulsa demasiado pronto o demasiado tarde, el

[1] Cuando se habla de notas, se hace referencia a los sprites azules rectangulares, que van bajando a velocidad constante y que hay que tratar de hacer desaparecer pulsando la tecla adecuada en el momento indicado.

jugador será penalizado. Obviamente la posición inicial y velocidad de estas teclas será la indicada para que, si el jugador juega correctamente, este se encuentre tocando la pieza musical en la que se basa el nivel al tempo correcto.

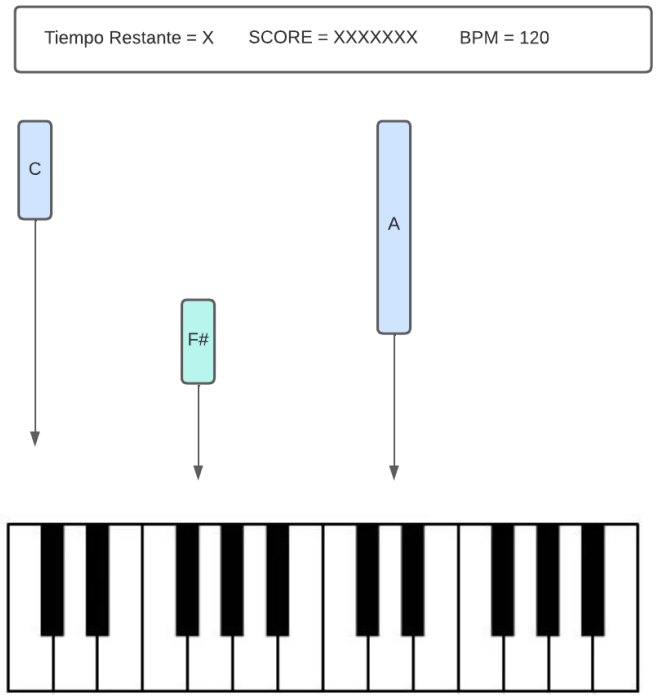


Figura 5.1: Primero borrador de diseño de la jugabilidad de la aplicación.

Una vez decidido esto, se inició el desarrollo de un prototipo de la aplicación. Este prototipo contaría con un piano virtual totalmente funcional, y un corto nivel pre-diseñado para probar la jugabilidad. Para no gastar demasiado tiempo en su desarrollo, este prototipo no podría por el momento leer o escribir archivos MIDI. Podemos observar una captura de pantalla del prototipo en *Unity* en la Figura 5.2. Se pueden visualizar las notas en azul, el sistema de puntuación y el piano virtual. Los puntos verdes encima de las teclas son el punto exacto donde deben estar las notas para conseguir la puntuación perfecta.

Una vez presentado el prototipo a el director de este proyecto y se decidió que este tipo de *gameplay* podría funcionar, entonces se pasó a la fase de diseño de la herramienta.

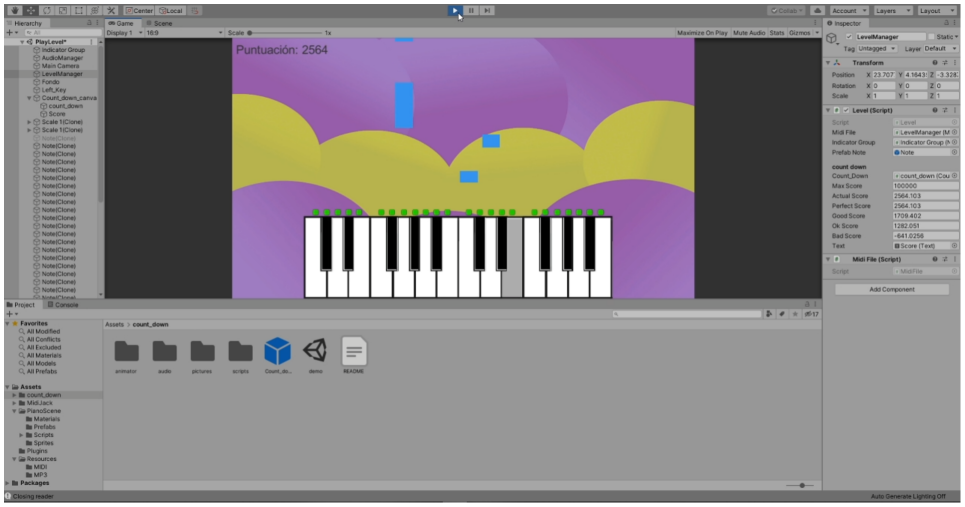


Figura 5.2: Captura del primer prototipo jugable de la herramienta en el motor *Unity*.

5.1.3. Diseño de la aplicación

En este apartado se detallará el diseño de la aplicación escena a escena [2], explicando la función de cada elemento presente en ellas.

- **Main Menu**. El menú principal de la aplicación debe de ser visualmente claro, fácil de entender y fácil de navegar. Por ello, se decidió que este contara con únicamente 4 botones, que fueran interactuables por medio del ratón y que contuvieran en ellos la mayor información visual posible para, de esta manera, poder usar la menor cantidad de texto posible.

  Como se puede observar en la Figura 5.18, tanto la primera versión como la última versión del menú son fieles a esta filosofía, aunque la segunda versión sea cromáticamente más clara.

  Como en el menú de opciones únicamente se podrá modificar el sonido tanto del piano como de los efectos especiales, no es necesario realizarlo en una escena nueva. Como es el estandard, la selección del volumen se realizará seleccionando el valor deseado en un *slider* (Figura 5.3).

  Por lo demás, las única función restante del menú será la de permitir navegar al usuario por las distintas escenas o salir de la aplicación si así lo desea.

- **Achivements**. Este es el nombre de la escena en la que el jugador puede mirar el historial de partidas. Esta escena, aunque sea sencilla,

[2] Cuando se habla de escenas, se hace referencia a las escenas de *Unity*.

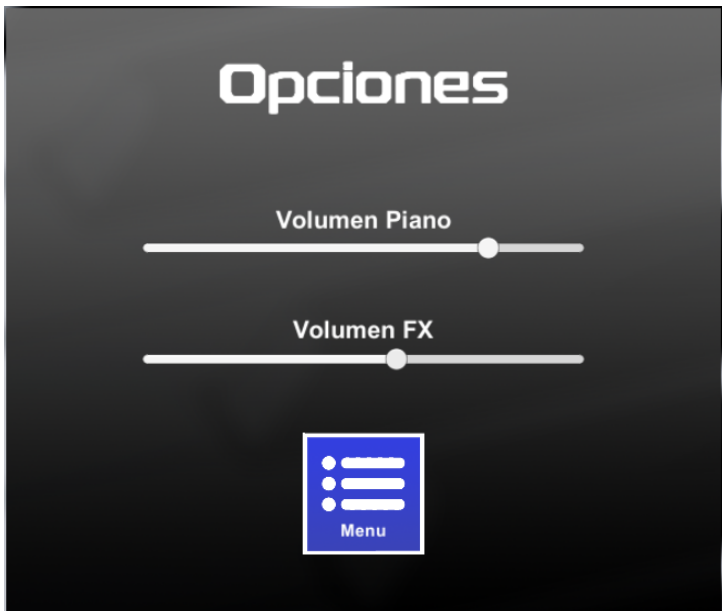


Figura 5.3: Menú de opciones de sonido.

es vital para la aplicación, ya que permitirá tanto al alumno como a los profesores analizar el rendimiento en cada uno de los ejercicios.

Para que resulte una pantalla sencilla de navegar, la escena consistirá en un panel, dónde aparecerán listadas una a una la información de cada partida. La última partida jugada aparecerá destacada y la primera en la lista, para poder acceder a ella de manera rápida. Además, los ejercicios dónde se haya conseguido una puntuación superior a 75.000 puntos se mostrarán con un tick, indicando al jugador que ese nivel ha sido completado satisfactoriamente.

El diseño final de esta pantalla se puede ver en la Figura 5.4.

- **Score Scene**. Esta escena se mostrará al finalizar una partida o, lo que es lo mismo, tras completar un ejercicio. En ella se presentará una sencilla animación en la que se revela la puntuación final obtenida, así como la experiencia adquirida. Tras mostrar esta información, en la parte inferior aparecerá un botón de volver al menú. Al pulsarse, se guardarán los datos de la partida (Figura 5.5).

  El nivel y los puntos de experiencia del jugador son una mecánica esencial para el juego, ya que permiten es dar una sensación de progreso al jugador, así como aumentar su motivación. Cada vez que complete un nivel, el jugador ganará una serie de puntos de experiencia. Si completa el mismo nivel de nuevo, ganará la mitad de puntos de experiencia que la primera vez en finalizarlo. El número de puntos obtenidos también



Figura 5.4: Pantalla de historial de partidas.

variará dependiendo de la puntuación obtenida, siguiendo la siguiente formula: $X = \frac{score*100}{maxScore}$, siendo X la experiencia ganada. El número de puntos de experiencia que se requiere para completar cada nivel es incremental y sigue la siguiente fórmula: $y = nivel * 75$, siendo y el número de puntos de experiencia que posee ese nivel. Cada vez que se sube un nivel se devuelven los puntos de experiencia a 0, para después sumar los puntos restantes.

- **Play Level**. Esta es la escena en la que el alumno podrá realizar los ejercicios mandados por el profesor o aquellos que encuentre por internet. El orden en el que el usuario experimentará cada una de las fases de esta escena es el siguiente:

  - La escena comienza con el explorador de archivos abierto. Su ubicación predeterminada sera el mismo directorio donde se encuentra el ejecutable del programa. El archivo escogido debe de ser un archivo con extensión .mid válido. De lo contrario el usuario será lanzado de vuelta al menú principal.
  - Una vez escogido un archivo válido, se pedirá al jugador que escoja la velocidad a la que quiere que se reproduzca el ejercicio. Esta funcionalidad fue añadida para que los alumnos que encuentren dificultades para tocar una pieza en particular, puedan interpretarla a menor velocidad y así poder acostumbrarse a ella. Por el contrario, si al alumno le resulta un ejercicio demasiado sencillo, puede probar a tratar de resolverlo a mayor velocidad, para así poner a prueba sus habilidades de digitación.

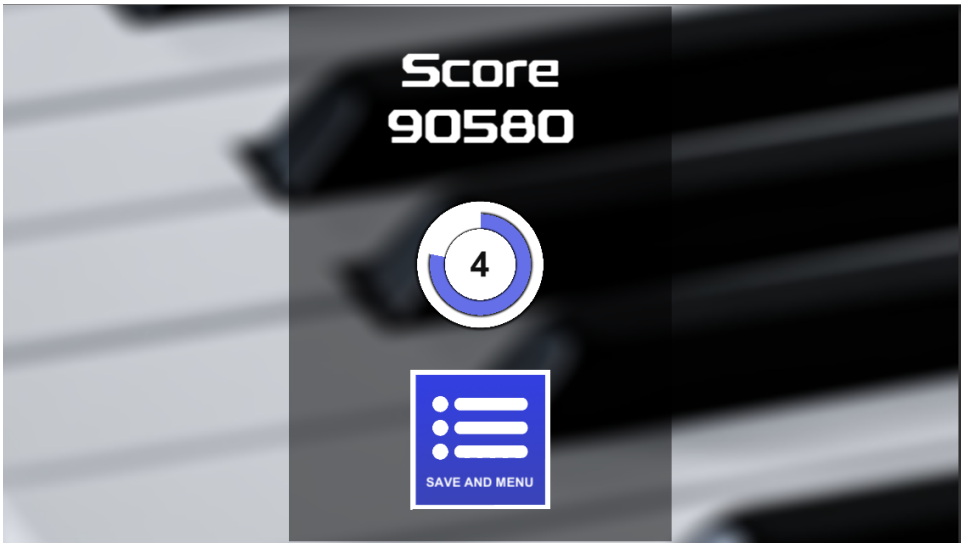


Figura 5.5: Escena de puntuación.

- Tras escoger la velocidad de reproducción el jugador deberá pulsar el botón de iniciar el nivel, el cual comenzará tras completar una cuenta atrás. El usuario podrá usar esta cuenta atrás para posicionar sus dedos en la parte correspondiente del teclado. Además, después de esta cuenta se le otorgarán un par de segundos de margen antes de que se empiecen a ver caer las primeras notas.
- Cuando haya comenzado el nivel, el jugador deberá de pulsar cada nota en el momento indicado. Irá ganando puntuación dependiendo de la exactitud con la que acierte las notas. Si pulsa una la nota que no es, o demasiado a destiempo se restarán puntos. En todos los niveles, el máximo de puntuación alcanzable será de 100.000 puntos. La puntuación máxima que se puede obtener con cada nota será igual a $\frac{100{,}000}{nmeroDeNotasTotal}$. Hay 5 tipos de resultados posibles tras pulsar una tecla:
  - **NAN**: Se pulsó la tecla cuando no había ninguna nota cerca. Se restará la mitad del valor completo de una nota.
  - **Bad**: Se pulsó la tecla cuando había una nota lejos. Se sumará una cantidad de $\frac{valorMaxNota}{4}$.
  - **Ok**:Se pulsó la tecla cuando había una nota a media distancia. Se sumará una cantidad de $\frac{valorMaxNota}{2}$.
  - **Good**:Se pulsó la tecla cuando había una nota cerca. Se sumará una cantidad de $\frac{valorMaxNota}{1{,}5}$.
  - **Perfect**:Se pulsó la tecla cuando había una nota a una distancia muy cercana. Se sumará el valor máximo de la nota.

En las notas largas, el jugador tendrá que mantener pulsada la tecla por lo menos un 75 % del tiempo que dura la misma. De lo contrario, se le penalizará.

Para poder predecir las siguientes notas que bajarán en pantalla, existirán unos *sprites* a modo de alerta en la parte superior de la pantalla, que avisarán de la cercanía de una nota.

Durante toda la duración del ejercicio el jugador podrá elegir reiniciar el mismo o volver al menú principal.

- Una vez terminada la partida, el jugador podrá o volver a empezar la pieza si piensa que puede hacerlo mejor. Si considera que ha finalizado, deberá darle al botón de continuar, el cual le enviará a la escena de puntuación o *Score Scene*.

En la Figura 5.6, se puede visualizar una captura de pantalla de cada una de las fases de esta escena.

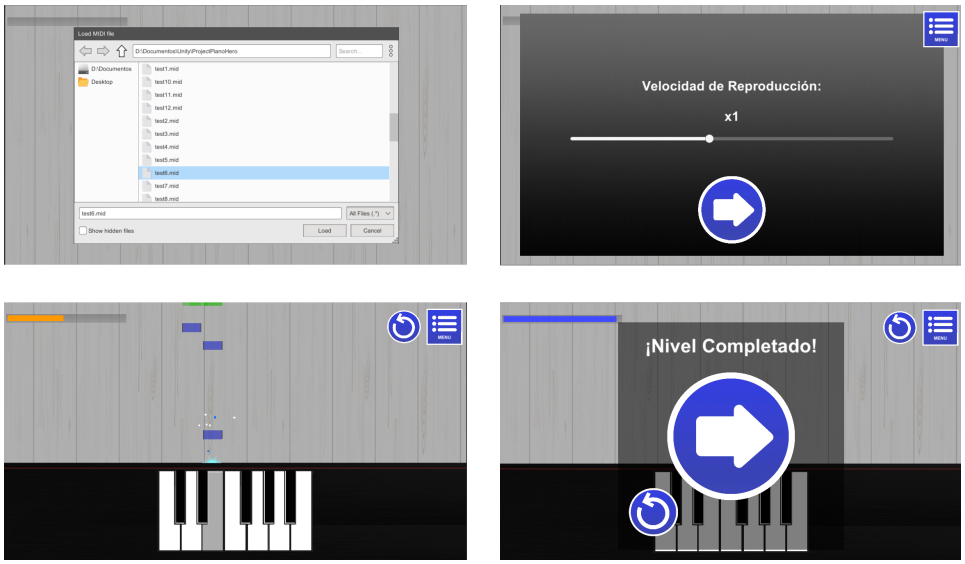


Figura 5.6: Diferentes fases de la escena de resolución de ejercicios. Orden: De izquierda a derecha y de arriba a abajo.

- **Record Level** es la escena en la que el usuario podrá grabar una pieza musical para posteriormente ser exportada a formato MIDI. El orden en el que el usuario experimentará cada una de las fases de esta escena es el siguiente:

  1. El usuario deberá elegir con qué rango de octavas va a querer tocar. Una vez seleccionada la octava más baja y la más alta, pasará a la siguiente estado.

2. En esta fase, el usuario tendrá que presionar el botón de REC para comenzar la grabación. Tras pulsarlo aparecerá una cuenta atrás. Una vez esta haya finalizado, comenzará la grabación. Es en este momento cuando todo input que el usuario haga con el controlador MIDI será guardado. Si el jugador no está contento con su interpretación, en cualquier momento puede volver a empezar inmediatamente presionando el botón de *restart*. Si considera que ha terminado, deberá pulsar el botón de *guardar*.
3. Una vez finalizada la grabación, el usuario podrá, o bien volver a empezar, o exportar la pieza ya tocada. Si desea exportar la pieza, deberá pulsar el botón de exportar. Una vez pulsado se abrirá el explorador de archivos, donde el usuario podrá elegir el directorio donde quiera que se guarde el archivo, así como su nombre. En el momento en el que pulse en el icono de guardar, se generará el archivo MIDI.
4. Una vez generado este archivo, el usuario podrá bien tratar de grabar otro, o volver al menú principal.

Se han tomado algunas decisiones a la hora de exportar el archivo MIDI. Para empezar, los bpm marcados en el archivo MIDI serán siempre de 120 pulsos por minuto. Esto no varía a la velocidad a la que se reproduce el archivo con respecto a la pieza interpretada. Otra decisión tomada fue la de eliminar el tiempo de silencio después de levantar la última tecla de la canción, ya que de esta manera podemos asegurar que no se crearán archivos con un final en silencio innecesariamente largo. De esta manera, a la hora de realizar el ejercicio el jugador no tendrá que esperar sin hacer nada a que termine la canción.

En la Figura 5.7, se puede visualizar una captura de pantalla de cada una de las fases de esta escena.

5.1.4. Interfaz de usuario

En el campo del diseño visual, el primer objetivo propuesto es que la aplicación pudiera ser visualmente sencilla, agradable y capaz de ser entendida rápidamente por cualquier tipo de usuario. Para ello, habría que limitar al mínimo toda la información innecesaria mostrada en pantalla.

Los primeros elementos a diseñar fueron las teclas del piano. En un primer momento se pensó que estas debían estar modeladas en 3D para dar un efecto de profundidad al ser pulsadas. Sin embargo, para fomentar la claridad visual,estas teclas 3D fueron remplazadas por unos *sprites* 2D lo más sencillos posible. Para dar la ilusión de que la tecla ha sido pulsada, se remplazará la imagen de la tecla en particular por una versión con los colores más oscuros.

Por otro lado, el diseño de los botones para navegar por los distintos menús deberían de ser llamativos y entenderse en un primer vistazo. Algunas

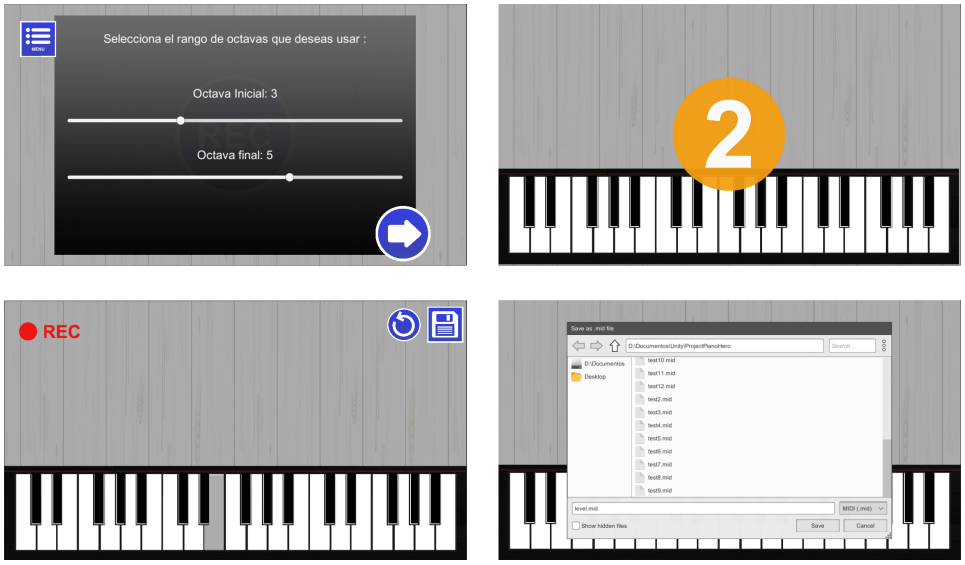


Figura 5.7: Diferentes fases de la escena de grabación. Orden: De izquierda a derecha y de arriba a abajo.

de las primeras versiones de estos botones eran de diversos colores, como rojas y verdes pero, sin embargo, fueron recoloreadas más adelante para así poder mantener una estética cromática coherente. Las versiones finales de estos iconos se encuentran en la Figura 5.8.

Sobre los fondos y su evolución, se comentará mas en detalle la sección de desarrollo (Sección 5.2). Sobre las animaciones y efectos de partículas, se comentará mas en detalle la sección de desarrollo (Sección 5.2).

5.1.5. Modificaciones finales

Aquí se comentarán los últimos cambios al diseño de la aplicación. Estos son los cambios que se decidieron aplicar tras haber casi completado el desarrollo, ya que tras probarla en su estado casi terminada es más sencillo ver que es lo que realmente funciona o no. También se incluirán las adiciones agregadas para la demo.

El primer cambio significativo fue el de agregar los mensajes de introducción y de tutorial, que sólo se enseñan la primera vez que se abre el juego, y se acceden a las respectivas escenas (Figura 5.9).

En un principio, al jugador se le enseñaba la puntuación durante el transcurso de la partida. Sin embargo, para aumentar la expectativa a la hora de mostrar la pantalla de puntuación final, así como para no distraer al jugador, se decidió cambiar esta puntación por una barra de progreso, que se va rellenando y cambiando de color a medida que el jugador va ganando puntos.

Otro cambio efectuado en los últimos días, fue el de crear dos versiones



Figura 5.8: Versiones finales de los botones que posee el juego. Se puede observar como los únicos con tonos rojos son el de salir de la aplicación, y el que se usa para indicar algun tipo de error o advertencia.

diferenciadas del juego: una para profesores y otra para alumnos. Aunque las dos versiones poseen las mismas funcionalidades, en cada una de ellas se destaca un modo de juego determinado en el menú principal, y se cambian los textos de introducción y tutorial. Esta decisión se tomó para de esta manera dejar más claro a los usuarios el funcionamiento de la misma, y como debería de ser usada idealmente.

También, se decidieron modificar el valor de puntuación máxima a 100.000 puntos por nivel (siendo anteriormente 10.000 puntos máximos por nivel). De la misma manera, se retocaron las distancias que otorgan las puntuaciones de *ok*, *good*, *bad* y *perfect*, para resultar más retante.

Tras las pruebas de usuario, y como último cambio, se decidió añadir algo más de *feedback* a la escena de grabación de ejercicios, haciendo que tras tocar una tecla se genere una nota que suba ascendentemente, representando que se ha registrado correctamente.

5.2. Desarrollo

En esta sección se detallará, paso a paso, cómo ha sido el desarrollo de cada una de las funcionalidades del proyecto. En esta sección no se va comentar en detalle el código, puesto que ya se encuentra comentado y explicado paso a paso en el Github público.

5.2.1. Esquema de recursos

En esta sub-sección se detallará la estructura de todos los elementos pertenecientes al proyecto de *Unity* de *Piano Unite*. En la Figura 5.10 se puede

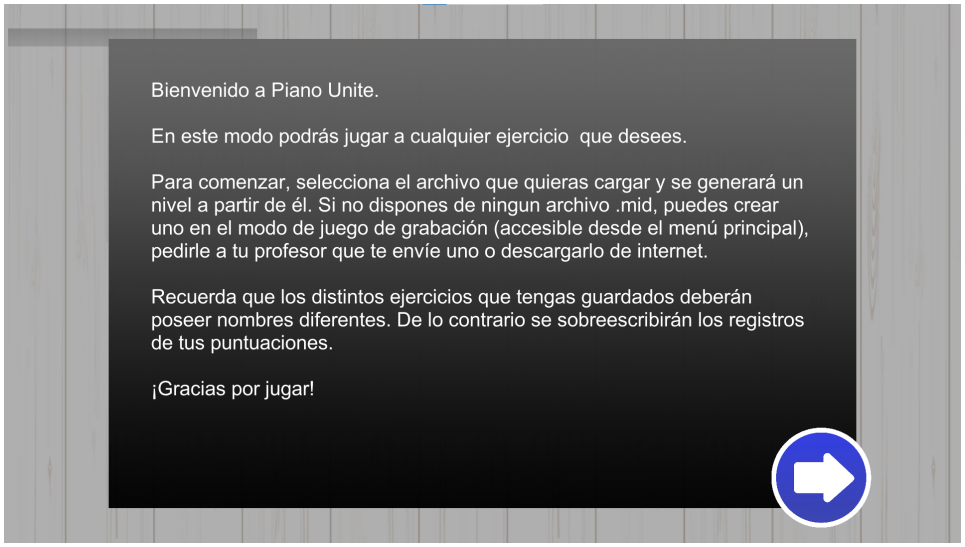


Figura 5.9: Mensaje de tutorial que se muestra en la versión para alumnos, al iniciar la escena de *PlayLevel* por primera vez.

observar una captura de pantalla de la jerarquía de ficheros del proyecto en *Unity*. En la carpeta de *resources* se encuentran los archivos usados para *debuggear*. En la carpeta de *PianoScene* se contiene todo lo desarrollado específicamente para este proyecto, mientras que las otras carpetas contienen los plugins externos usados.

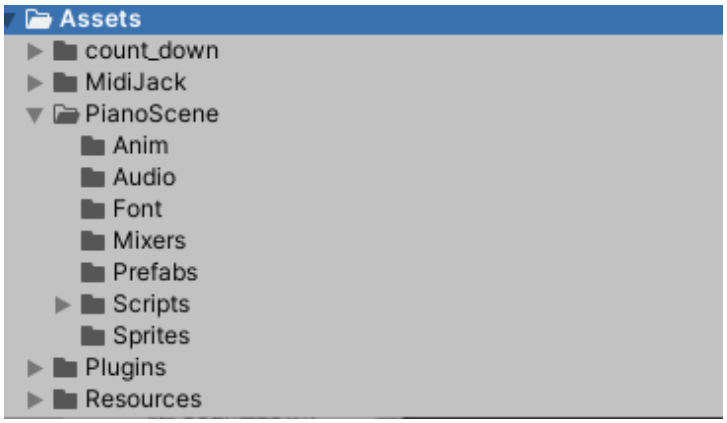


Figura 5.10: Captura de la jerarquía de carpetas del proyecto en *Uniy*.

Ahora, se listarán todos los *assets* creados para el proyecto, junto a una muy breve descripción de cada uno de ellos.

- Animaciones:
  - **Star**: Animación que aparece encima de las teclas al ser pulsadas

en el momento correcto.

- Audios:
  - **Select.wav**: Audio de seleccionar.
  - **Tick.wav**: Audio de cancelar o volver atrás.
- Fuentes:
  - **Arial**: Fuente usada en la mayoría del texto.
  - **Rexlia rg**: Fuente para el título del juego.
- Mixers:
  - **Piano.mixer**: Mixer principal, que contiene las pistas de Master y Piano.
- Prefabs:
  - **Aviso**:Prefab Aviso de nota aproximandose
  - **Note**: Prefab de nota.
  - **Scale**: Prefab de cada una octava, con sus respectivas notas y detectores de notas.
  - **Element**: Prefab de elemento de UI del historial de partidas.
  - **Particle**: Prefab del efecto de partículas.
  - **XP**: Prefab de la rueda que indica el nivel y experiencia del jugador.
- Escenas:
  - **MainMenu**: El menú principal.
  - **PlayLevel**: La escena consistente en importar un archivo MIDI y "jugarloçomo un nivel.
  - **RecordLevel**: La escena que permite grabar una pieza musical y exportarlo en formato MIDI.
  - **ScoreScene**: La escena que aparece tras completar un nivel, dónde se revela la puntuación obtenida y la experiencia ganada.
  - **Achivements**: La escena para poder ver el historial de todas las partidas y sus puntuaciones.
- Scripts:
  - **AchivementManager.cs**: Administra la pantalla de historial.
  - **Data.cs**: Los datos que se guardan al guardar la partida.

- **Detector.cs**: Detecta las notas que se aproximan a una tecla.
- **FileManager.cs**: Funciones del explorador de archivos.
- **Level.cs**: Administra la escena de *PlayLevel*.
- **MainMenuUI.cs**: Las funciones de la interfaz de usuario del menú principal.
- **ManageScenes.cs**: Para cambiar de escenas.
- **MidiFile.cs**: Capaz de leer archivos MIDI y traducirlos a objetos MidiEvent.
- **MidiRecorder.cs**: Exportar la lista de eventos grabada a formato MIDI.
- **Note.cs**: Movimiento y funciones de cada una de las notas que caen en *PlayLevel*.
- **NoteAlert.cs**: Control del sprite que avisa de las notas que se aproximan.
- **NoteIndicator.cs**: Cada una de las teclas del piano.
- **NoteIndicatorGroup.cs**: Administra e instancia cada una de las teclas del piano.
- **ObjectPool.cs**: Para crear *pools* de objetos de un *prefab* determinado.
- **RecordLevel.cs**: Administra la escena de *RecordLevel*.
- **SaveController.cs**: Permite guardar y cargar la partida.
- **ScoreCoreo.cs**: Administra la escena de *ScoreScene*.

- **Sprites** (Figura 5.11):

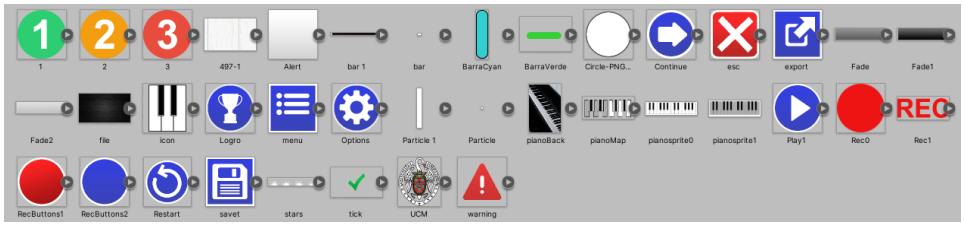


Figura 5.11: Captura de todos los sprites usados en el juego.

5.2.2. Desarrollo del piano virtual

El primer paso en el desarrollo del proyecto *Piano Unite* es el desarrollo en *Unity* de un piano virtual que pueda ser controlado por medio de un teclado MIDI.

Para comenzar se creó el proyecto de *Unity*, implementando en él la librería de *MidiJack*. Esta librería se puede importar como plug-in de *Unity* y consta de 4 scripts y una serie de escenas de prueba para comprobar y testear su funcionamiento. En este trabajo, no interesa entrar al funcionamiento de este plug-in internamente. Lo que sí cabe comentar el único método que usaremos en este proyecto, el método "MidiJack.MidiMaster.GetKey(int noteNumber)". Este método devolverá un valor *float* con posibles valores [0,1] en cada *frame*. Este valor indicará la velocidad de pulsación de la tecla introducida como parámetro de entrada, siendo 1 el valor máximo posible y 0 el valor que indica que la tecla no ha sido presionada. En este caso, el rango total de notas que se pueden tocar van desde la nota A0 (siendo su número de nota el 9) hasta Db8 (siendo su número de nota el 109)[3].

Para iniciar la implementación, se crean dos *scripts*:

- **NoteIndicator.cs**. Cada tecla del piano virtual poseerá una instancia de este *script*. Este se encargará de detectar si la tecla ha sido pulsada y, en caso afirmativo, disparar el sonido correspondiente a la nota que representa al volumen correspondiente. Más adelante se le agregarán funcionalidades como reproducir la animación correspondiente en el *sprite* de la tecla, o instanciar el detector de las notas que caen en el modo de jugar un nivel.

- **NoteIndicatorGroup.cs**. Existirá una única estancia de este *script* en cada escena que haga uso del piano virtual. Este se encarga de instanciar y administrar todos las instancias de NoteIndicator.cs. Les pasará parámetros como su número de nota, su archivo de sonido correspondiente, su posición o el modo de juego en el que están siendo instanciadas. Además, el número de escalas que se utilizarán puede ser modificado al principio de cada escena, por lo que se adaptará a ello.

En un principio, cada una de las teclas eran instanciadas por separado, pero esta manera resultaba ser incómoda e ineficiente a la hora de proporcionarles una posición y tamaño correspondientes. Por ello, se hará uso de un *prefab*[4] llamado *scale*, el cual a su vez contiene los *prefabs* de cada una de las 12 teclas que corresponden a los 12 tonos de una octava, con sus *sprites*, fuentes de sonido, animaciones...etc.

Para crear la ilusión de pulsación de cada nota del piano, se han realizado dos *sprites*[5] por cada nota, uno de la tecla presionada y otro de la nota

[3]En este proyecto se utilizará la notación musical anglosajona. En este caso, el A0 equivale a un LA0 y el Db8 a un DOb 8.

[4]En *Unity*, un *prefab* hace referencia a un objeto pre-creado, al cual se le pueden agregar una serie de componentes y parámetros ya establecidos antes de la compilación del proyecto. De esta manera se facilita el código y se ahorra tiempo.

[5]El concepto de *sprite* se usa en el mundo de los videojuegos, y hace referencia a una imagen 2D que representa a un personaje, objeto, edificio o cualquier otro tipo de elemento que exista dentro del juego.

liberada. Cada vez que se pulse o libera una tecla, se alternará entre estos dos *sprites*.

Una vez implementados todos los elementos mencionados anteriormente en esta subsección, obtenemos un teclado virtual completamente funcional. Más adelante, en la subscección de funcionalidades sonoras, se comentará más en detalle todo lo relacionado con el sonido de las notas del piano.

5.2.3. Desarrollo de la lectura MIDI

La lectura de archivos MIDI ha sido una de las partes más intrincadas de desarrollar del proyecto. El objetivo de esta funcionalidad es poder traducir cada uno de los eventos MIDI de un archivo a valores dentro del juego que se puedan usar de manera más sencilla.

Como se ha comentado en capítulos anteriores, los archivos MIDI están escritos en binario y en big-endian. Además, existe una gran cantidad de posibles eventos distintos y mensajes de sistema. Sin embargo, se necesita recoger todos los tipos de eventos, ya que en este caso sólo se necesita emular archivos que contengan una o varias pistas de piano. Por ello, eventos como *polyphonic aftertouch* o *pitch wheel range*, entre otros, no serán recogidos. Los eventos que si serán reunidos y utilizados para generar las notas en el juego son los siguientes:

- **VoiceNoteOn**: Indica el tick del comienzo de una nota, su velocidad de pulsación, el canal y su ID.
- **VoiceNoteOff**: Indica el tick del final de una nota, su velocidad, el canal y su ID.
- **SystemExclusiveEvent**: Estos eventos pueden indicar un gran número de variables. Algunos de ellos son:
  - **MetaText**: Usado para añadir texto. (Opcional)
  - **MetaCopyright**: Usado para indicar el *copyright* de la pieza. (Opcional)
  - **MetaTrackName**: Indica el nombre de la pista que comienza a continuación. (Opcional)
  - **MetaInstrumentName**: Indica el nombre del instrumento usado en la pista a continuación. (Opcional)
  - **MetaLyrics**: Indica la letra de la canción. (Opcional)
  - **MetaEndOfTrack**: Indica la finalización de una pista.
  - **MetaSetTempo**: Indica el tempo de la canción, en microsegundos por corchea.
  - **MetaTimeSignature**: Indica el compás de la canción, en diferentes formatos.

- **MetaKeySignature**: Indica la armadura de la pieza. (Opcional)
- **SystemExclusiveBegin / SystemExclusiveEnd**: Indican el comienzo y final de un evento SystemExclusive especial.

Hay otro tipo de eventos que, aunque debamos tener en consideración para leer un archivo MIDI, no serán utilizados para nada en nuestra herramienta:

- **VoiceAfterTouch**. Utilizado para medir la profundidad de la pulsación de una tecla del controlador MIDI. Usado especialmente en PADS.
- **VoiceControlChange**.Indica un cambio en ciertos ajustes del controlador MIDI.
- **VoiceProgramChange**. Indica un cambio en los modos del controlador MIDI.
- **VoiceChannelPressure**. Utilizado para medir la profundidad de la pulsación del canal.
- **VoicePitchBend**. Utilizado detectar cambios de tono utilizando las ruedas de cambio de tono del controlador.

El código relacionado con la lectura de archivos MIDI se ha desarrollado en el script **MidiFile.cs**[6].

Para guardar de forma organizada todos los eventos MIDI, haremos uso de una lista la cual la estructura representada en la Figura 5.12.

5.2.4. Desarrollo de la escritura MIDI

La escritura en archivo MIDI ha sido un trabajo de análisis y comprensión de cada uno de los distintos elementos, cabeceras y estándares utilizados en este formato. Además, ha requerido la realización de operaciones en binario para ser capaz de representar algunos valores ya que, como se explica en la sección de tecnologías MIDI en el capítulo de estado de la cuestión (Capítulo 2), este formato tiene una particular forma de representar valores altos, para así agilizar la transmisión de información.

Para realizar la funcionalidad de grabar un archivo MIDI, se creó una nueva escena llamada **RecordLevel**. Esta escena contiene los siguientes elementos: Cámara, el canvas para los objetos de la UI y un objeto *level* que

[6]La lectura secuenciada de MIDI y su traducción a una lista de eventos más fácilmente usable está inspirado en el código en C++ desarrollado por el usuario *OneLoneCoder*, que podemos encontrar en su GitHub, al cual se puede acceder a través del siguiente enlace: `https://github.com/OneLoneCoder/olcPixelGameEngine/blob/master/Videos/OneLoneCoder_PGE_MIDI.cpp`.

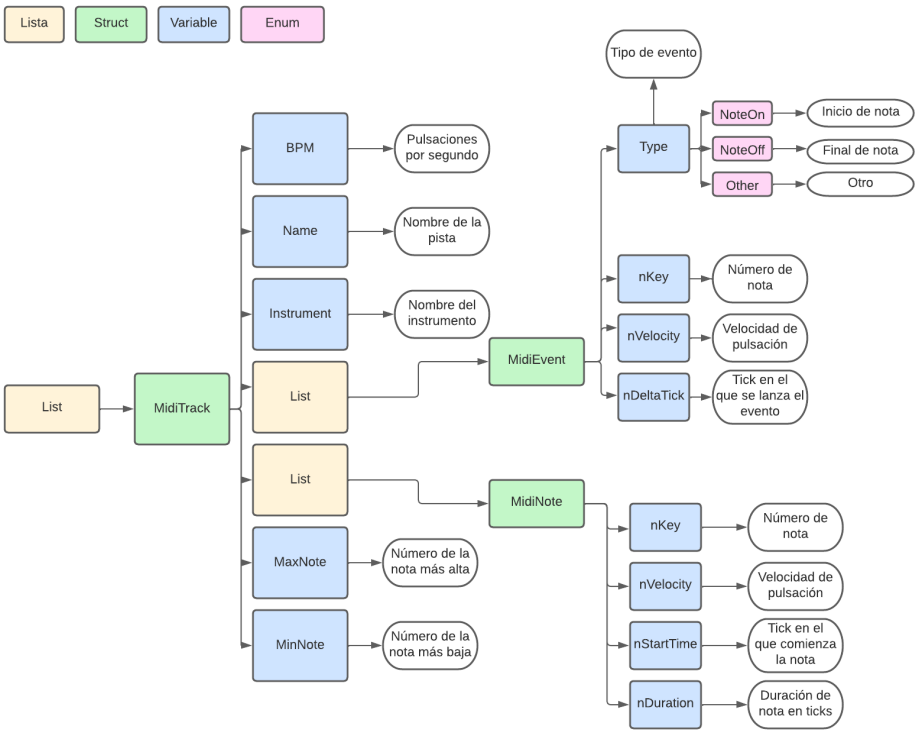


Figura 5.12: Diagrama de organización de los eventos MIDI recopilados.

posee el *script* que se encarga de controlar el funcionamiento de la escena así como el guardado de archivos MIDI.

Como el funcionamiento de la escena ya ha sido descrito en la sección de diseño de este mismo capítulo, aquí nos limitaremos a hablar sobre los dos *scripts* principales de la escena:

- **RecordLevel.cs** se encarga de administrar la escena. Esto incluye los botones y elementos de UI, así como las distintas fases de la escena. Además, instanciará el piano virtual dependiendo del número de octavas que se haya decidido anteriormente y se le indicará que está en el modo grabación. También albergará la instancia de la lista que que va guardando cada uno de los eventos recogidos del controlador MIDI entre la finalización de la cuenta atrás y la pulsación del botón de *stop*. Los eventos o *MidiEvents* que se añaden a la lista son los mismos que se pueden ver en la Figura 5.12. Cada vez que la nota sea pulsada, el mismo script **NoteIndicator.cs** que controla su comportamiento enviará un nuevo evento para guardarlo en la lista. Al finalizar la grabación, esta lista de elementos sera enviadoa a la instancia del *script* que genera el archivo MIDI, y será vaciada en caso de que el usuario

decida reiniciar la grabación.

- **MidiRecorder.cs** se encarga de recibir la lista de los eventos y exportarla en el formato MIDI. Para ello, se abre un *stream* y secuencialmente se añade la información necesaria, ya sea en forma de cabeceras, eventos u otro tipo de mensajes. Una vez terminado el proceso, se guardará el archivo en la ruta indicada. Al principio, este script podía grabar todos los distintos tipos de eventos mencionados en el apartado de lectura de MIDI. Sin embargo, incluir todas estas opciones resultó ser innecesario, aparte de que podía acarrear algunos errores dependiendo del modelo del controlador MIDI que se estuviese usando. Por ello, se decidió sólo guardar los eventos de *NoteOn*, *NoteOff*, *MetaEndOfTrack*, *MetaTimeSignature*, *MetaKeySignature* y *MetaTrackName*. Con estos eventos, se puede crear un archivo MIDI de una única pista de piano, que es todo lo que se necesita para este juego.

5.2.5. Desarrollo de la jugabilidad

El concepto de desarrollo de jugabilidad se refiere a la parte de la herramienta consistente en la realización de los ejercicios o, llamado de otra manera, la escena en la que es posible jugar a los distintos niveles. Esta se llama *PlayLevel*, y funcionamiento ha sido detallado en la sección de diseño. Similarmente a la escena de *RecordLevel*, esta consta de los siguientes elementos: Cámara, canvas para los objetos de UI y un objeto *level* que posee los scripts que controlan el funcionamiento de la partida, así como la lectura de archivos MIDI.

Los principales scripts de *PlayLevel* son los siguientes:

- **Level.cs**. El trabajo de este *script* es encargarse de la lógica del nivel. Eso incluye: Administrar la puntuación, las distintas fases de la escena, llevar la UI, determinar si se ha llegado al final del nivel e instanciar las distintas notas a pulsar en sus respectivas posiciones y velocidades. Para optimizar el rendimiento de la aplicación durante el nivel y que pueda funcionar a 60 FPS en una gran variedad de dispositivos, se crean todas las notas antes de empezar el nivel, en lugar de ir instanciándolas a medida que transcurre el mismo.

- **Note.cs**.Script que controla cada una de las distintas notas que posee la pieza a tocar. Hay dos tipos distintos de notas, las largas y las cortas. Las cortas sólo requieren de una pulsación rápida de la tecla, mientras que las largas necesitan una pulsación prolongada en el tiempo. El tiempo de pulsación dependerá de las dimensiones de la nota. La velocidad de las notas dependerá del número de BPM *(Beats Per Second)* del archivo MIDI. Todas ellas son instanciadas al principio del nivel, y se van destruyendo a si misma a medida se encuentren en una

posición lo suficientemente baja en el eje Y. También es la propia nota la que le manda la puntuación ganada correspondiente a Level.cs. Para que la herramienta no se vea saturada en caso de que el nivel posea una gran cantidad de notas, los componentes de estas quedarán inactivos hasta que sea su momento de entrar en la pantalla. Una vez hayan salido, se destruirán a sí mismas.

- **Detector.cs** es la clase que se encarga de detectar las notas que caen cuando se encuentran a cierta distancia. Para ello, se hace uso de el sistema de *RayCasy*, el cual consiste en lanzar un rayo en una dirección determinada y comprobar si algún objeto colisiona con él.

- **NoteAlert.cs**. Este *script* se encarga de controlar los pequeños avisos de que se aproxima una nota, haciendo visible o invisible el *sprite*, dependiendo de a la distancia que se encuentre la nota más cercana por encima del objeto con la instancia.

El resto del trabajo en cuanto a la jugabilidad ha ido dirigido a cambiar valores de parámetros para equilibrar lo más posible el juego. Parámetros como la distancia a la que se puede detectar las notas, las puntuaciones obtenidas o el tamaño de los objetos.

5.2.6. Menús y HUD

Tras el desarrollo de las dos principales funcionalidades de la aplicación, se debe comenzar a realizar el resto de escenas: *MainMenu*, *ScoreScene* y *Achivements*.

Para navegar por los distintos menús, se usarán objetos de tipo *Button* proporcionados por *Unity*. Estos objetos forman parte del canvas, y será necesario añadir un objeto de tipo *EventSystem* para que puedan ser detectados al clicar el ratón, haciendo uso de un *RayCast*.

Para el control de la navegación por los distintos menús y escenas, se han desarrollado los siguientes *scripts*:

- **MainMenuUI.cs**. Su trabajo es administrar todos los aspectos de la navegación por la escena de *MainMenu*. Además incluye la pestaña de opciones, así como aplicar los cambios producidos en esta por el usuario. Por último, también se ocupa de mantener el nivel del jugador actualizado.

- **ManageScenes.cs** tan sólo contiene un único método, que sirve para cambiar de escena. Se llamará a este método a través de los botones.

- **ScoreCoreo.cs**. *Script* que controla la coreografía de la escena de *ScoreScene*. Esto incluye el incremento de la puntuación y la animación

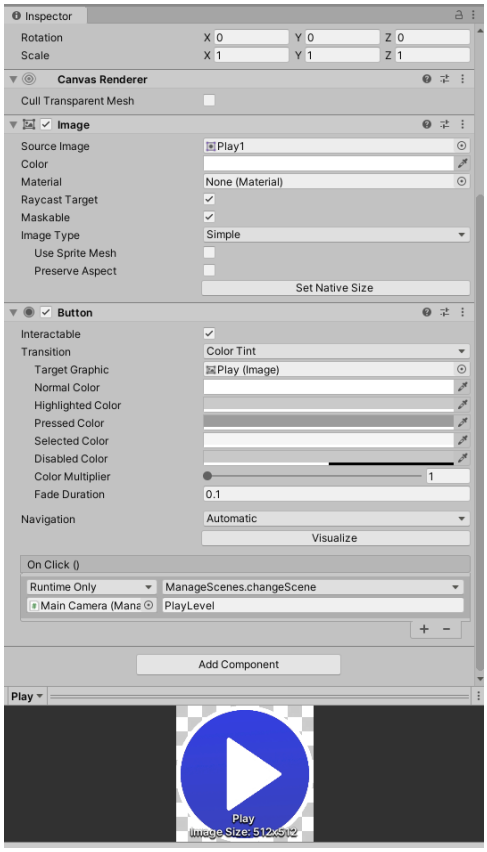


Figura 5.13: Captura de objeto *Button* de *Unity*. Se puede ver las distintas opciones de personalización del botón que ofrece. Este componente ahorra mucho tiempo, facilitando la implementación de botones en la aplicación.

de subir de nivel. Estas animaciones están realizadas por código, por lo que es necesario hacer uso de las corrutinas de *Unity* para su desarrollo[7].

- **AchivementManager.cs** administra la escena de *Achivements*. Esto incluye el panel desplazable y sus elementos, el marcador de nivel y los botones. Cada uno de los elementos en el panel desplazable corresponde a los datos guardados de cada nivel completado. Para poder guardar y enseñar estos elementos, así como mostrar el nivel, se tuvo que desarrollar el sistema de guardado del juego, el cual se comentará en la siguiente sección.

El desarrollo tanto del HUD y la UI, así como de los *sprites* fue reali-

[7]Según al documentación de *Unity*, una corrutina es una función que tiene la habilidad de pausar su ejecución y devolver el control a *Unity* para luego continuar donde lo dejó en el siguiente frame. Así, se puede ejecutar un bucle a lo largo de varios frames.



Figura 5.14: Versión final del menú (en la versión para profesores). Empezando por la izquierda, encontramos las siguientes opciones: Grabar ejercicio, jugar un ejercicio, observar las mejores puntuaciones, la pestaña de opciones y el botón de cerrar aplicación.

zado en Adobe Photoshop. Algunas imágenes e iconos fueron creados desde cero, mientras que otras se crearon modificando en forma, tamaño y color imágenes vectoriales sin copyright. La imagen de fondo en el menú principal fue extraída del siguiente banco de imágenes gratuito `https://unsplash.com/es/fotos/y4rvvuftDfk`, y posteriormente modificada en tamaño, color, rotación y añadiendo efectos de desenfoque.

5.2.7. Implementación del audio

El concepto de desarrollo sonoro hace referencia tanto al sonido de cada una de las teclas, como a los efectos de sonido presentes a lo largo de los distintos menús.

Al comienzo del desarrollo, se decidió usar FMOD (Firelight Technologies, 1995) y su implementación con *Unity* para poder implementar el sonido de cada una de las teclas del piano de forma más sencilla y compactada en un sólo banco de sonido. Sin embargo, esta opción resultó ser ineficiente y traer algunos problemas como fallos al responder, o mucho tiempo de latencia que arruinaba la experiencia del piano virtual. Por ello, se determinó en cierto punto del desarrollo retirar FMOD del proyecto, y usar los sistemas de *AudioSource* que posee el motor de forma nativa.

El proyecto cuenta con 102 archivos de audio en .wav distintos: 100 para cada una de las notas de piano, y 2 para los efectos de sonido de los botones

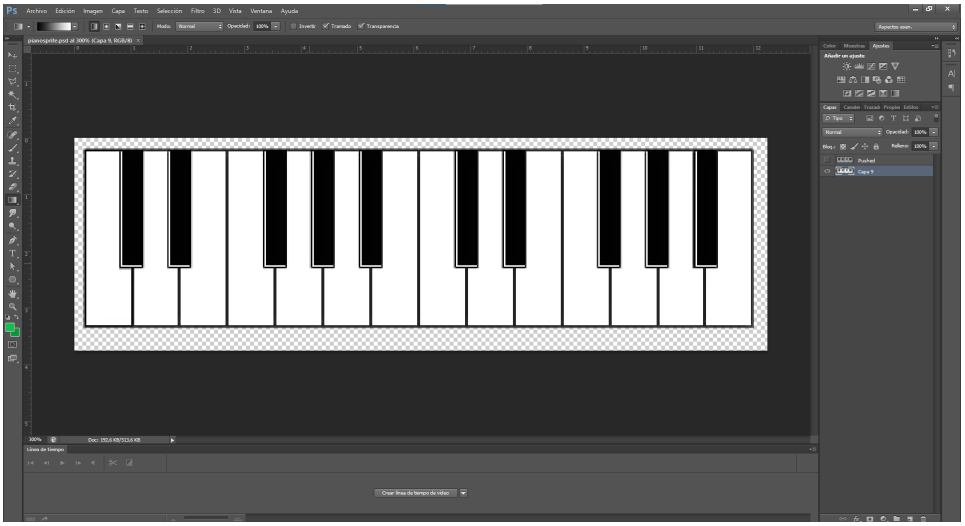


Figura 5.15: Proyecto de Photoshop de los sprites de las distintas teclas de una octava en un piano. Posteriormente, serán separadas en notas individuales.

de la interfaz de usuario. El efecto de sonido 1 (*select1*) será usado para los botones que te permiten avanzar en el menú, guardar o cambiar de escena, mientras que el sonido 2 (*select2*), más sutil que el primero, es para volver atrás, cancelar y cerrar pestañas. Estos dos audios han sido extraídos de la página `https://freesound.org/`, web que ofrece una gran cantidad de efectos de sonido gratuitos sin copyright.

Para poder diferenciar el volumen de las teclas del piano de los efectos de sonido de los botones y así poder modificar sus volúmenes de forma independiente desde la pestaña de opciones, se creó un *Audio Mixer* de *Unity*, con dos canales diferenciados. Ambos canales se encuentran conectados a su vez al canal *Master* (Figura 5.16).

5.2.8. Demo jugable

Una vez implementado todo lo anterior, se comenzó a preparar la demo para ser usada en las pruebas de usuario. Para ello, se realizaron unos cambios de última hora para mejorar la aplicación de cara al usuario:

- El primer cambio, y el más significativo consistió en **dividir la herramienta en dos versiones** distintas: Una para profesores y otra para alumnos. Las razones de este cambio ya han sido explicadas en el apartado de diseño. La diferencia entre las dos versiones consistió en realizar dos versiones del menú principal, con los iconos intercambiados. En la versión para profesores se enfatiza el modo de juego de grabación, mientras que en la de alumnos el de resolución de ejercicios.

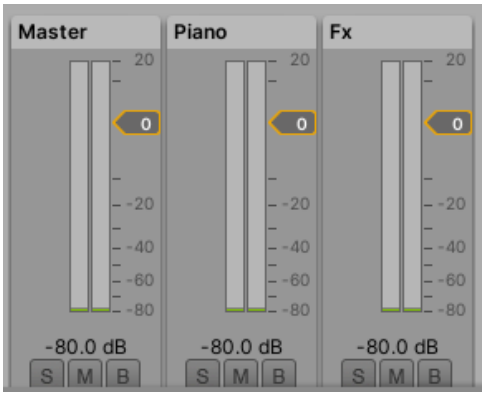


Figura 5.16: Mezclador de sonido integrado en *Unity*.

- Además, para cada una de las dos versiones se realizaron unos **mensajes a modo de tutorial**. Estos mensajes solo aparecerán la primera vez que se inicia el juego. El primero de ellos aparecerá nada más abrir la aplicación, mientras que los otros dos al seleccionar uno de los modos de juego por primera vez.

- Cambios en la **paleta de colores**. En un principio, existía una variedad en los colores de la UI y de los fondos escogidos sin sentido estético ni teoría del color. Por ello, se decidió cambiarlos para usar únicamente el color azul (en diferentes tonos), el blanco, el negro y diferentes tonos de gris. La única excepción a esta regla son los indicadores de nota que son verdes, el logo de la UCM y el botón de cerrar la aplicación. Podemos observar las diferencias en los dos diseños visuales en la Figura 5.18.

- Por último, se agregó la posibilidad de **usar el ratón para tocar el teclado virtual**. Aunque lo ideal es que las pruebas de usuario se realicen con un controlador MIDI, se decidió agregar esta funcionalidad para la gente que deseara probarlo y no contara con uno.

La demo jugable se subió en el siguiente enlace `https://drive.google.com/file/d/18NpsaNnztUIUc6CP36juCP8mJ8FRv5bw/view?usp=sharing`.

La herramienta, tras los cambios realizados después del testeo, será publicada en la plataforma de itch.io (`https://narratech.itch.io/piano-unite`).

5.2.9. Funcionalidades auxiliares

Para finalizar este capítulo, se detallarán las funcionalidades adicionales que se han implementado en el proyecto. Aunque no posean su propia sección en este capítulo, son necesarias para el correcto funcionamiento de la herramienta, tal y como ha sido diseñada. Las funcionalidades son las siguientes:



Figura 5.17: Capturas de pantalla de los textos de introducción. Sólo se enseñan la primera vez que se inicia la aplicación.

5.2.9.1. Sistema de guardado

Para el sistema de guardado, se ha utilizado el sistema de serialización de clase automática que proporciona *Unity* en C# ([System.Serializable]). De esta manera, sólo es necesario crear la clases con los atributos que queramos serializar, y crear dos métodos públicos: uno que escriba todos los valores de una instancia de esta clase con los datos; y otra que abra el archivo indicado, deserialice la información y la escriba en el mismo.

Se usan dos métodos para el guardado:

- **Data.cs** es el *script* que contiene la clase serializable que usaremos para guardar toda la información del jugador y de sus partidas. Los contenidos de esta han ido evolucionando con el tiempo, a medida que



Figura 5.18: A la izquierda, la pantalla de menú principal y la de grabación del juego antes de los cambios visuales. A la derecha, las mismas pantallas después de los cambios visuales.

se han añadido posibles variables, valores y preajustes. La estructura de esta clase *Data* es la que se puede observar en la Figura 5.19

- **SaveController.cs** tiene dos métodos, uno llamado LoadData() que, en caso de que exista un archivo de guardado, devuelve una instancia de Data; y SaveData(Data data) que se encarga de serializar la información contenida en data y la escribe en un archivo en binario llamado "data.fun". Tratandose de una aplicación gratuita y educativa, no se vió necesario encriptar los datos de guardado.

5.2.9.2. Explorador de archivos

El objetivo era desarrollar la funcionalidad de poder acceder al explorador de archivos, de manera que el usuario pueda cargar y guardar archivos desde cualquier directorio en su propio equipo. Dentro de la librería de *UnityUtilites* existe una forma sencilla de acceder al explorador de archivos del sistema operativo en tiempo de ejecución. Sin embargo, esta forma sólo funciona para acceder cuando se compila el juego dentro del editor, por lo que hubo que acceder a un asset gratuito de la *asset store* (Runtime File Browser).

El código relacionado con esta funcionalidad se encuentra en el archivo **FileManager.cs**. Este *script* se encarga de permitir cargar y guardar archivos, utilizando este explorador de archivos personalizado como interfaz. Para su correcto funcionamiento, es necesario, una vez más, usar el sistema de corrutinas de *Unity*. Está API también permite personalizar ligeramente

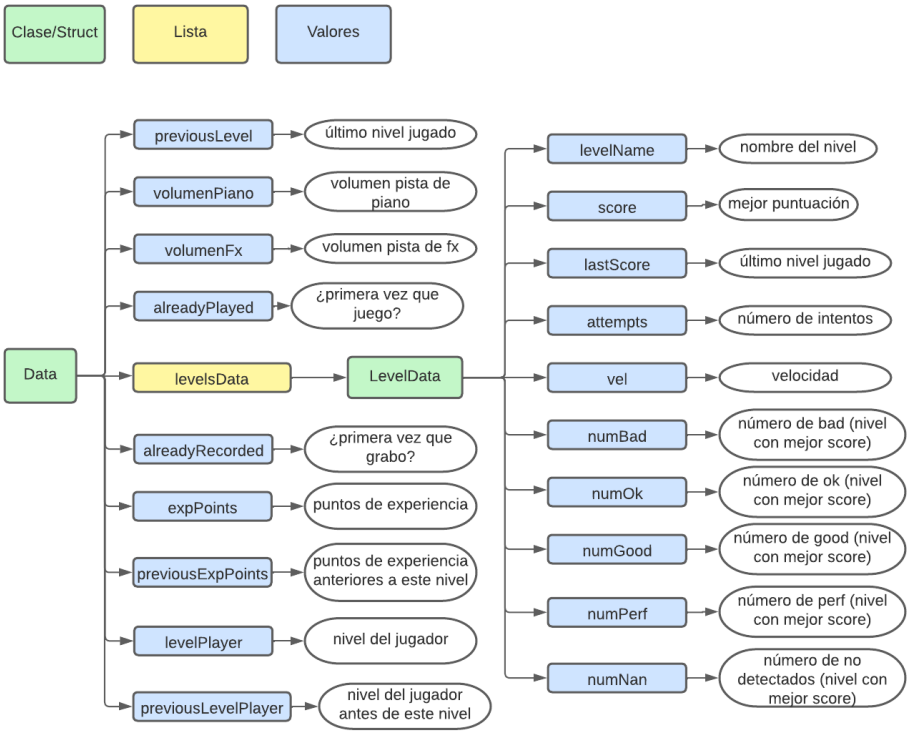


Figura 5.19: Estructura de los datos de guardado.

el explorador de archivos, y las opciones que nos brinda.

5.2.9.3. Animaciones y efectos de partículas

Adicionalmente, se creó una animación 2D en *Photoshop* para cuando se pulsa una tecla en el momento en el que hay una nota pasando por encima del detector. Esta animación es importante para dar una mejor sensación de feedback por parte del juego.

Por último, se desarrolló un efecto de partículas que simula una pequeña explosión de confeti blanco y azul que sale cada vez que desaparece una nota de la pantalla, debido a que ha sido pulsada y procesada ya. Este efecto fue creado usando el componente de *Particle System* de *Unity*. Este efecto aumenta significativamente la sensación de satisfacción al procesar una nota con éxito.

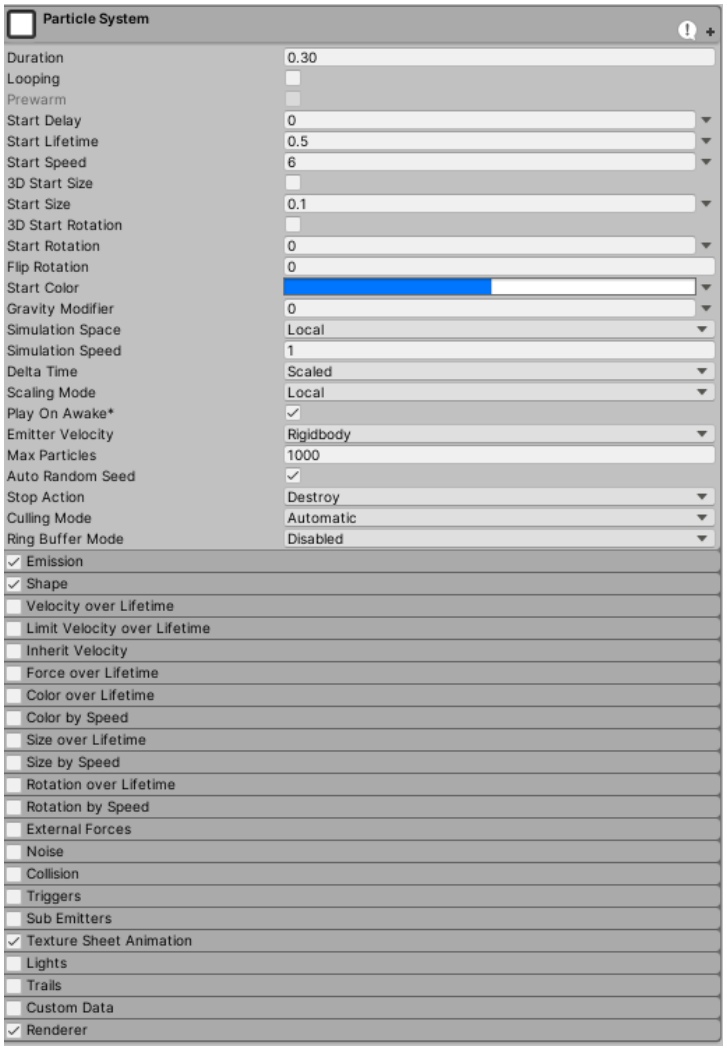


Figura 5.20: Sistema de partículas de Unity.

Capítulo 6

Pruebas y resultados

En este apartado se abordarán las cuestiones referentes a las pruebas de usuario, consistentes en el testeo de una demo y la cumplimentación de un formulario.

Debido a los requisitos necesarios para su testeo como es poseer un ordenador con sistema operativo *Windows*, con conocimientos básicos de piano y, sobre todo, poseer un teclado o controlador MIDI, resulta bastante complicado encontrar un número significativo de *testers* para realizar las pruebas.

6.1. Pruebas

El objetivo de estas pruebas de usuario es dar respuesta a las siguientes cuestiones:

- ¿Existe potencial en el concepto de la herramienta?
- ¿Podría cumplir la aplicación con su objetivo educacional de servir de apoyo docente al profesor?
- ¿Podría cumplir la aplicación con su objetivo de motivar a los alumnos?
- ¿Está la aplicación diseñada adecuadamente?
- ¿Existen errores que deban ser corregidos?
- ¿Es la herramienta funcional desde un punto de vista de usabilidad?
- ¿Que cambios y mejoras se podrían hacer a la demo de testeo para aumentar su atractivo de cara al usuario?
- ¿Que cambios y mejoras se podrían hacer a la demo de testeo para aumentar su atractivo de cara aumentar su utilidad?

6.1.1. Guión de las pruebas

Para encontrar las respuestas, se ha desarrollado un formulario cuyo guión es el siguiente:

"Hola a todos,

Soy Guillermo Sánchez de Lamadrid, estudiante del grado de Desarrollo de Videojuegos en la Facultad de Informática de la UCM, y este es un cuestionario cuyo fin es recoger todas las opiniones, sugerencias e informes de errores acerca de Piano Unite, la aplicación creada para mi Trabajo Fin de Grado (TFG).

Piano Unite es una aplicación que permite a los profesores de piano crear ejercicios y grabar piezas musicales en tiempo real para que, tras enviárselos a sus alumnos, estos puedan practicar piano desde su casa, a la vez que se divierten jugando al estilo de juegos populares como *Guitar Hero* o *Synthesia*. Tras completar cada uno de los distintos niveles, los alumnos podrán compartir sus resultados con el profesor para que este pueda llevar un seguimiento de su rendimiento y progresión en el manejo del instrumento. Gracias al uso de la tecnología MIDI, también se podrá interpretar cualquier canción de piano en ese formato (con extensión .mid) que encuentres en Internet.

Existen dos versiones de la aplicación, una para los profesores y otra para los alumnos, aunque ambos pueden acceder a las mismas funcionalidades desde las dos. Aunque se pueda usar la aplicación únicamente con el ratón, está pensada para utilizarse con un piano MIDI compatible con Windows (¡conéctalo si lo tienes, por favor!), siendo esto lo ideal pues el objetivo es practicar la digitación en un instrumento real.

Por favor, realiza los siguientes pasos:

1. Descarga el archivo PianoUnite.zip desde este link y descomprímelo en tu escritorio: (Link de descarga)

2. Una vez descomprimido, te encontrarás con dos carpetas: una con la versión para alumnos, otra con la versión para profesores.

3. Primero, abre la carpeta con la versión para profesores. Para lanzar la aplicación ejecuta el archivo "Piano Unite (prof).exe"de la carpeta PianoUniteProf. Si tu antivirus considera una amenaza el intentar ejecutar un fichero .exe, dale permisos para hacerlo. Una vez abierta la aplicación, prueba a grabar y exportar una pieza musical breve, que toques desde el teclado, usando la opción REC en el menú principal (si quieres, puedes hacer otras pruebas con la aplicación, pero esta es la básica). Recuerda guardar el ejercicio que has tocado en una carpeta donde tengas permito de acceso, como por ejemplo una carpeta llamada "Niveles"dentro de la carpeta que ha aparecido al descomprimir el ZIP, PianoUniteDemo. Crea la carpeta Niveles con el botón derecho del ratón, entra dentro y escribe el nombre del fichero, asegurándote que se graba con extensión .mid.

4. Una vez que hayas grabado uno o más ejercicios, ya puedes cerrar del

todo la versión para profesores.

5. Después, abre la versión para alumnos. Para ello ejecuta el archivo PianoUnite(est).exe en la carpeta de PianoUniteAlumno. Una vez abierta la aplicación, prueba a jugar al nivel que hayas creado anteriormente. Para ello tienes que usar el navegador de archivos y dirigirte a la carpeta Niveles donde los guardaste anteriormente. También podrás probar una serie de niveles ya creados que corresponden a varias piezas populares a la hora de empezar a tocar el piano.

6. Puedes ir al menú y darle al icono de trofeo para ver el historial de ejercicios realizados. Verás en color azul el último ejercicio y si has conseguido una puntuación suficiente, aparecerá una marca que indica que has SUPERADO ese ejercicio. ¡La idea es practicar hasta conseguirlo!

7. Cuando hayas decidido terminar de explorar las dos versiones de la aplicación y sus diversas funcionalidades, por favor regresa a este cuestionario y contesta las preguntas que vienen a continuación. El cuestionario consta de dos secciones, una donde se realizarán una serie de preguntas acerca de ti, y otra dónde se realizan preguntas acerca de tu experiencia con la aplicación. La encuesta es completamente anónima, ya que las preguntas sobre datos demográficos sólo se usarán para distinguir por tipos de usuario.

¡Muchísimas gracias a todos por participar! Vuestras impresiones y comentarios serán de gran utilidad para poder seguir mejorando el proyecto.

Guillermo Sánchez de Lamadrid : gusanc03@ucm.es "

6.1.2. Preguntas realizadas

La primera sección del cuestionario se basa en preguntas que tienen relación con el perfil del usuario, para así poder clasificar mejor las respuestas dependiendo de sus características como usuario.

Otra sección de las preguntas del cuestionario están basadas en el modelo de cuestionario SUS. El cuestionario SUS o cuestionario de *Sistemas de Escala de Usabilidad* es un sistema que, de forma rápida y eficaz, permite medir la usabilidad de una gran variedad de tipos de productos y servicios, incluyendo casi cualquier tipo de hardware y software. Creado originalmente por John Brooke, el cuestionario consta de 10 preguntas, las cuales deben de ser respondidas seleccionando una de entre 5 opciones. Opciones que existen en un rango que comprende desde totalmente en desacuerdo a totalmente de acuerdo.

Las partes del cuestionario relacionadas con el cuestionario SUS están basadas en la plantilla visible en la figura 6.1. Sin embargo, se han omitido algunas preguntas que podrían resultar redundantes o que no tienen tanta razón de ser en una aplicación con estas características. Esta medida se tomó para amenizar el cuestionario a los *testers*.

Las preguntas omitidas son las siguientes:

Participant ID: _____ Site: _____ Date: ___/___/___

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

		Strongly Disagree				Strongly Agree
1.	I think that I would like to use this website frequently.	☐	☐	☐	☐	☐
2.	I found this website unnecessarily complex.	☐	☐	☐	☐	☐
3.	I thought this website was easy to use.	☐	☐	☐	☐	☐
4.	I think that I would need assistance to be able to use this website.	☐	☐	☐	☐	☐
5.	I found the various functions in this website were well integrated.	☐	☐	☐	☐	☐
6.	I thought there was too much inconsistency in this website.	☐	☐	☐	☐	☐
7.	I would imagine that most people would learn to use this website very quickly.	☐	☐	☐	☐	☐
8.	I found this website very cumbersome/awkward to use.	☐	☐	☐	☐	☐
9.	I felt very confident using this website.	☐	☐	☐	☐	☐
10.	I needed to learn a lot of things before I could get going with this website.	☐	☐	☐	☐	☐

Please provide any comments about this website:

Figura 6.1: Plantilla de cuestionario SUS para aplicaciones web.

- I found this website very cumbersome/akward to use (Encuentro esta web incómoda/rara de usar).
- I felt very confident using this website (Me sentí muy confiado usando esta web).
- I needed to learn a lot of things before I could get going with this website (Necesité aprender un montón de cosas antes de poder empezar a manejarme en esta web).

Por último, se realizan preguntas relacionadas más estrechamente con *Piano Unite* y sus funciones. Además, se proponen una serie de preguntas abiertas para que los probadores pueden explayarse en sus opiniones acerca de la aplicación, así como reportar errores y proponer posibles mejoras y cambios.

Las preguntas, listadas en el mismo orden que en el cuestionario enviado, son las siguiente:

1. ¿Cuál es tu edad?

2. ¿Cuál es tu nivel de experiencia con el piano? (Opciones: Nunca he tocado el piano / Principiante / Intermedio / Avanzado / Profesional).
3. ¿Cuál es tu formación en el mundo del piano? (Opciones: Ninguna formación / Alumno autodidacta / Alumno de profesor particular / Alumno de academia / Alumno de conservatorio / Alumno de estudios avanzados de piano).
4. ¿Cuál es tu relación con la enseñanza de piano? (Opciones: Ninguna relación / Profesor amateur / Profesor particular / Profesor de academia / Profesor de conservatorio / Profesor de otro instrumento o materia relacionada con el mundo de la música).
5. ¿Tienes alguna experiencia con aplicaciones o videojuegos de enseñanza musical (Rocksmith, Synthesia, Simply Piano...)? Si la respuesta es sí, indica cuál/cuales.
6. ¿Qué modelo de teclado MIDI has usado con la aplicación?
7. He comprendido el concepto de Piano Unite. (Rango : 1 = Totalmente en desacuerdo / 5 = Totalmente de acuerdo).
8. Me agradaría utilizar esta aplicación frecuentemente, ya sea con el rol de alumno o profesor.(Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
9. Encuentro esta aplicación innecesariamente compleja. (Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
10. Encuentro esta aplicación fácil de usar.(Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
11. Pienso que las distintas funciones de esta aplicación están bien implementadas.(Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
12. Pienso que la mayoría de gente sería capaz de aprender a usar esta aplicación rápidamente.(Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
13. Pienso que hay demasiada inconsistencia en esta aplicación. (Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo).
14. Creo que necesitaría ayuda para poder usar esta aplicación. (Rango : 1 = Totalmente en desacuerdo - 5 = Totalmente de acuerdo)
15. Finalmente, puedes usar este espacio para hacer los todos comentarios adicionales que creas convenientes. Por ejemplo: ¿qué cambiarías o mejorarías de la actual versión de Piano Unite?, ¿qué eliminarías?, ¿qué opinas acerca de que existan dos versiones?...)

16. Si durante la prueba has experimentado algún fallo o problema con la aplicación, por favor, descríbelo a continuación.

6.2. Resultados y discusión



Figura 6.2: Foto de persona jugando a Piano Unite.

Obtener un número elevado de participaciones ha sido una tarea complicada, puesto que los requisitos para ser un potencial usuario de Piano Unite han sido más complicados de reunir de lo que se esperaba en un primer momento. Para recordar al lector, estos requisitos son poseer un teclado MIDI, poseer un sistema operativo Windows y tener mínimos conocimientos de piano, o por lo menos tener interés en aprender el instrumento.

Por ello, se decidió cambiar el enfoque del análisis de los resultados, de una evaluación cuantitativa a una evaluación cualitativa.

El mayor problema para reunir a profesionales en la enseñanza de piano con el fin de probar esta aplicación fue que todos los preguntados en academias y conservatorios, hacían uso de ordenadores con sistema operativo

MacOs. Las razones de este fenómeno escapan a la intención del trabajo pero, aún así ,nos sugiere que si se quiere hacer una aplicación dirigida a ser utilizada por músicos, se debe de hacer compatible con este sistema.

Este conocimiento llegó en un momento en el que, por razones de tiempo y falta de medios, se dificultaba enormemente el hacer una versión para MacOs. Aún así, esta versión se realizó y se adjuntó al cuestionario. Sin embargo, debido a la imposibilidad por parte del autor de probar el funcionamiento de esta versión debido a la falta de medios, no cayó en la cuenta de que, para poder ser ejecutada en Mac, se le debían conceder una serie de permisos a la aplicación, ya que de lo contrario no sería ejecutada en absoluto. Esto provocó que la mayoría de los sujetos que intentaron probarla en Mac reportaran que el programa no funcionaba, por lo que no realizaron el cuestionario. Únicamente un usuario fue capaz de resolver este inconveniente y realizar el formulario, pero cuando informó de los pasos que había que seguir para ejecutar el juego ya había finalizado el plazo de pruebas. Este párrafo tiene como objetivo servir como advertencia para posibles proyectos que se puedan llegar a encontrar con la misma sucesión de inconveniencias en el futuro.

6.2.1. Características de los participantes

El tiempo que estuvo abierta la participación en el cuestionario fue desde el 3 de mayo al 17 de mayo. Participaron un total de 5 personas, las cuales cuentan con los perfiles respecto al mundo del piano que se pueden observar en las figuras 6.3, 6.4 y 6.5.

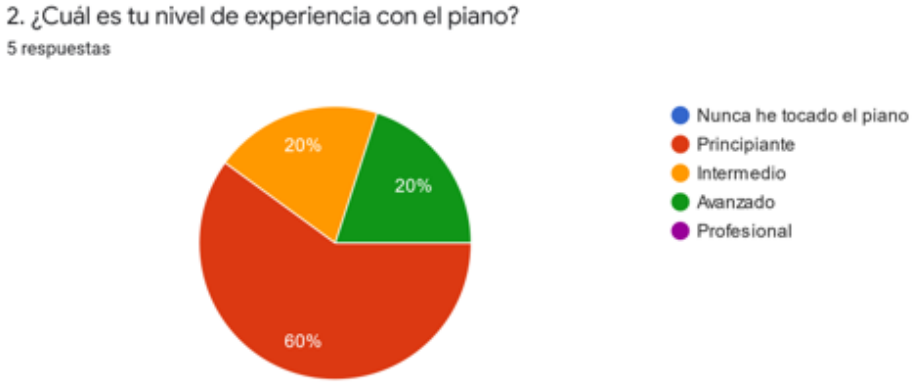


Figura 6.3: Gráfica que recopila el nivel de experiencia en piano de los usuarios de prueba.

Como se puede deducir tras analizar estas gráficas, la variedad de los perfiles participantes no ha sido la deseada .Se cuenta con un solo pianista avanzado. Aún así, ya que esta herramienta va dirigida a principiantes, no

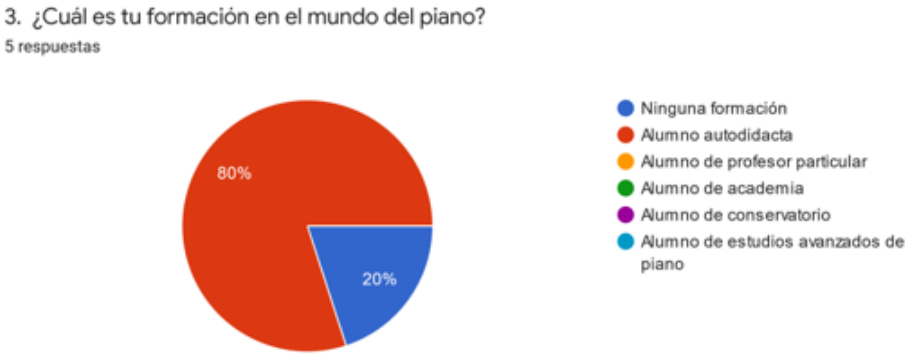


Figura 6.4: Gráfica que recopila el nivel de formación en piano de los usuarios de prueba.

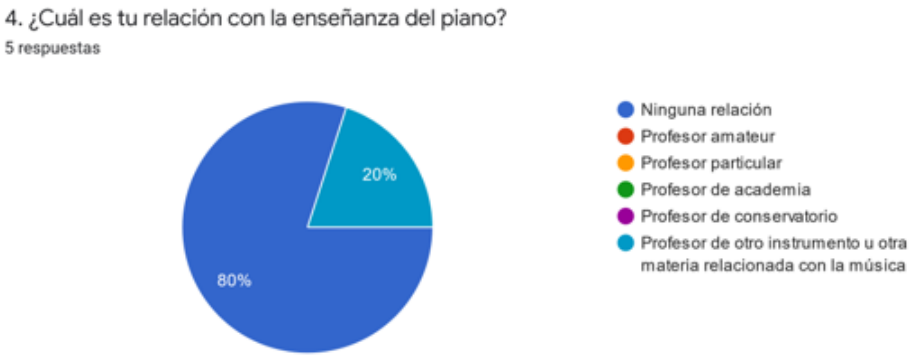


Figura 6.5: Gráfica que recopila la relación de los usuarios con la enseñanza de piano.

supone un mayor problema. Sin embargo, sólo se cuenta con la participación de un docente relacionado con el mundo de la música el cuál, aunque no resulta de ser docente de piano, puede aportar información útil desde la perspectiva de profesor. El resto de participantes son usuarios de pianos MIDI que poseen conocimientos de principiante o intermedios en el instrumento. Ninguno de los encuestados ha realizado estudios de piano en una academia o conservatorio, lo que nos deja sin poder conocer sus opiniones acerca de la aplicación desde ese punto de vista.

Otra información que merece la pena mencionar es la referente a las edades (Figura 6.6), previa experiencia con aplicaciones similares (Figura 6.7), y modelos de los controladores MIDI (Figura 6.8).

Respecto a la experiencia con otros programas de objetivos parecidos, todos lo participantes menos uno habían usado previamente alguna, por lo

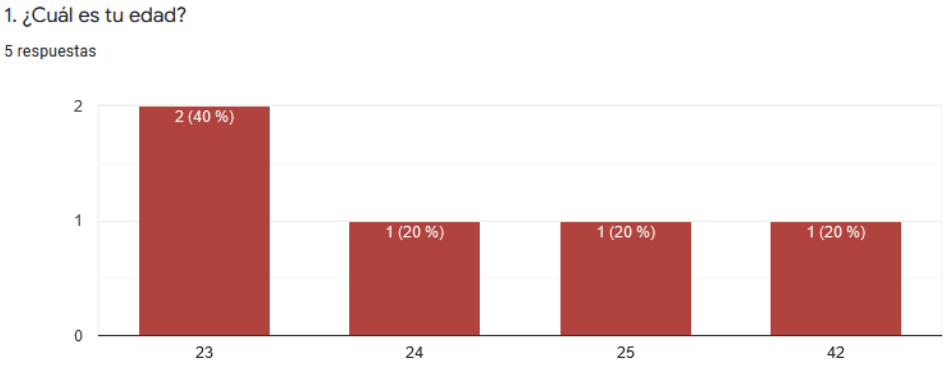


Figura 6.6: Gráfica que recopila las edades de los participantes.

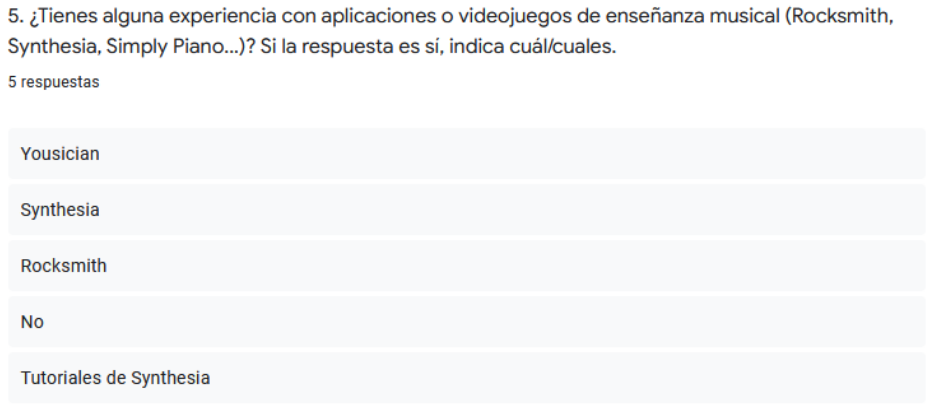


Figura 6.7: Respuestas de las distintas experiencias de los participantes con aplicaciones similares.

que les podría resultar más sencillo de entender el concepto de la aplicación, así como ser capaces de compararlas con Piano Unite.

Por otro lado, podemos encontrar una gran variedad en cuanto a los controladores MIDI utilizados por los usuarios. Ninguno de los participantes afirma haber tenido algún problema con su uso, por lo que se puede concluir que Piano Unite es compatible con, por lo menos, estos 5 modelos. Esto nos sugiere que es bastante probable que sea compatible con otro gran número de ellos.

6.2.2. Resultados del cuestionario SUS

Los resultados de las preguntas correspondientes al cuestionario SUS no pueden tenerse en demasiada consideración debido al bajo numero de parti-



Figura 6.8: Respuestas de los distintos controladores MIDI usados en las pruebas.

cipaciones. Aún así, se comentarán brevemente las respuestas obtenidas, que se encuentran representadas en las figuras 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16.

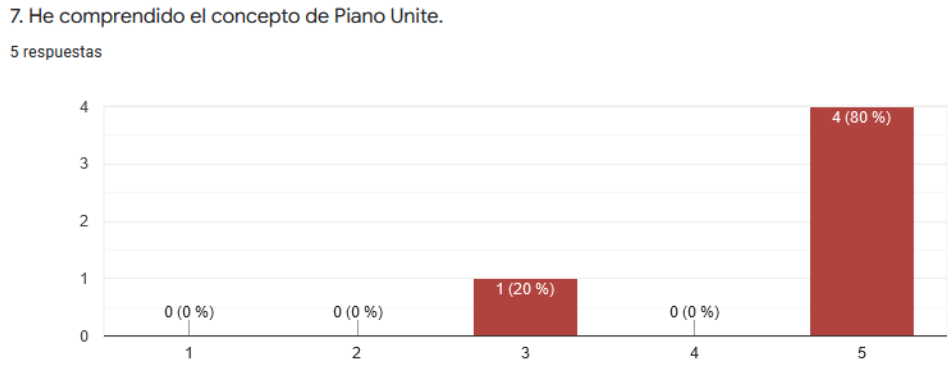


Figura 6.9: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 7 del cuestionario.

Lo que se puede sacar en claro de estas respuestas es que 4 de los 5 usuarios han tenido una experiencia de usuario agradable, comprensible y accesible en general; mientras que el usuario restante tuvo una opinión completamente opuesta. Por suerte, este mismo expondrá en detalle los motivos de su disgusto en las preguntas posteriores, en las cuales se brindaba a los sujetos la posibilidad de desarrollar su opinión. En el siguiente apartado se discutirán en detalle estas respuestas.

Para conseguir la puntuación SUS de cada uno de los participantes, se sumarán las puntuaciones de cada encuestado en cada una de las preguntas

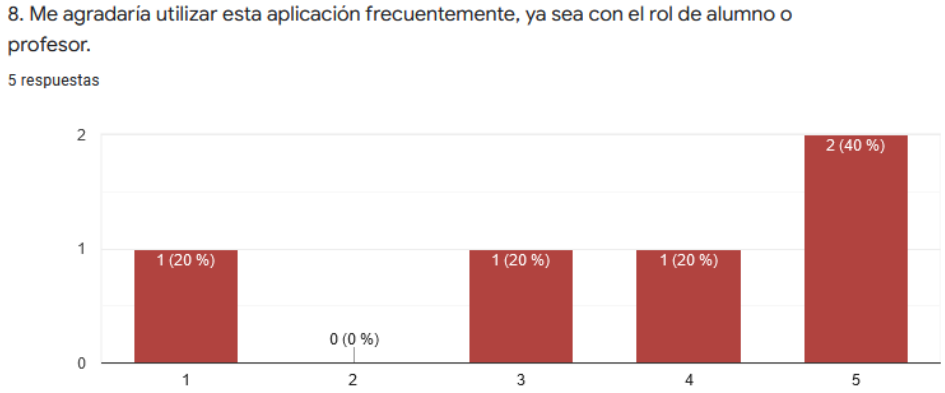


Figura 6.10: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 8 del cuestionario.

(en caso de que la pregunta esté formulada de forma que la respuesta más positiva sea 1, se suma 6 menos la puntuación). Esto nos dejará con una puntuación sobre 40, por lo que podremos multiplicar el resultado por 2,5 para poder tenerlo sobre 100. Como la encuesta es anónima, no se puede saber la puntuación exacta de cada encuestado, por lo que sumaremos la puntuación total de todos los encuestados, y la dividiremos entre 5 para hacer la media. La puntuación total de todos los valores de la gráfica es de 140.

$$X = \frac{140*2,5}{5} = 70.$$

Según las directrices de el sistema SUS, una puntuación entre 68 y 80,3 indica que la aplicación es correcta en cuanto a usabilidad se refiere, aunque puede mejorar. Menos de 51 sería un suspenso, y más de 80,3 sería una puntuación excelente.

De todas formas, cabe reiterar que con la cantidad reducida de sujetos de prueba obtenidos, estos valores no deben de ser tomados en demasiado en cuenta.

6.2.3. Resultados de las preguntas de desarrollo

En esta sección, se repasarán las respuestas de los encuestados a la pregunta número 15. Esta pregunta dice así: "Finalmente, puedes usar este espacio para hacer los todos comentarios adicionales que creas convenientes. Por ejemplo: ¿qué cambiarías o mejorarías de la actual versión de Piano Unite?, ¿qué eliminarías?, ¿qué opinas acerca de que existan dos versiones?...) ".



Figura 6.11: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 9 del cuestionario.

Las dos primeras respuestas que merece la pena comentar es la mostrada en la Figura 6.17.

Lo que tienen en común estas dos respuestas es que hacen referencia a la necesidad de invertir más esfuerzos en el feedback de la aplicación, tanto en la parte de juego y de grabación como en otras funcionalidades como la del explorador de archivos que puede dar la sensación de que no se está guardando correctamente. Este problema al identificar si se graba un archivo o no muy probable se deba al hecho de que, si se decide guardar con una extensión diferente a .mid, no se guardará nada. Sin embargo, no se lanza ningún mensaje de error, sino que la aplicación continúa normalmente. Debido a varios problemas que han tenido algunos usuarios entendiendo el explorador de archivos, se ha pensado el eliminar esta funcionalidad y hacer que el guardado y cargado de archivos fuera automático. Sin embargo, el hecho de poder acceder al explorador de archivos hace más accesible el poder importar niveles

Invertir en mejorar este aspecto en cualquier aplicación o juego es invertir en mejorar significativamente la experiencia de usuario, por lo que de cara a la versión final se mejorarán estos aspectos.

Hay otra respuesta que hace referencia a que la aplicación debería dejar hacer un *setup* previo donde se puedan ajustar las opciones de teclado, tarjeta de sonido... etc. Esto puede ser de gran utilidad si el usuario dispone de varias tarjetas o controladores conectadas a su ordenador. Sin embargo, esta no debería ser una prioridad en principio, ya que todos estos ajustes se pueden modificar desde el propio sistema operativo y, aunque sería más cómodo que se pudiesen también realizar desde la propia aplicación, no es una funcionalidad necesaria.

El último comentario de esta pregunta, y el que más merece la pena

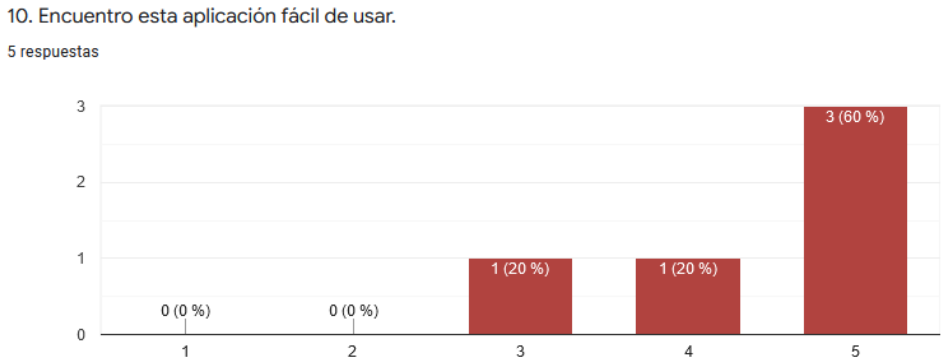


Figura 6.12: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 10 del cuestionario.

comentar en profundidad es el de la Figura 6.18.

La primera cuestión que hace referencia a la necesidad de tener un teclado de dos octavas es falsa, puesto que con un teclado de una sola es más que necesario a la hora de grabar y jugar ejercicios, ya que pregunta al principio el número de octavas que se desea usar. Aún así, también nos sugiere que se podría añadir una advertencia del número de octavas que se necesitan para tocar cada una de los ejercicios antes de comenzar. Otro punto que hace es el de que la aplicación podría ampliarse para incluir partituras. Sin embargo, como se ha explicado desde el principio, esta aplicación sirve para practicar digitación y técnicas de piano, no aprender partituras ni teoría musical. Aún así, esa podría ser una funcionalidad que incluir en una ampliación futura.

En cuanto a la pregunta que trata de recoger reportes de posibles errores o bugs en la herramienta, mientras que algunos encuestados han señalado que no se encontraron con fallos, otros llegaron a reportar cosas como los permisos necesarios que hacen falta para su ejecución en MacOs, que al principio no se escuchaba por problemas con su tarjeta de sonido o algún problema visual con la UI o la resolución. Todos los problemas de UI y resolución fueron solucionados en el momento de leer las respuestas, mientras que los otros se dejaron ya por falta de más información o falta de capacidad de solucionarlos.

6.2.4. Discusión de los resultados

En esta sección se tratará de sacar conclusiones acerca de los resultados obtenidos en las pruebas.

Comenzaremos respondiendo a cada una de las preguntas planteadas al principio de este capítulo. Estas preguntas son:

1. ¿Existe potencial en el concepto de la herramienta?

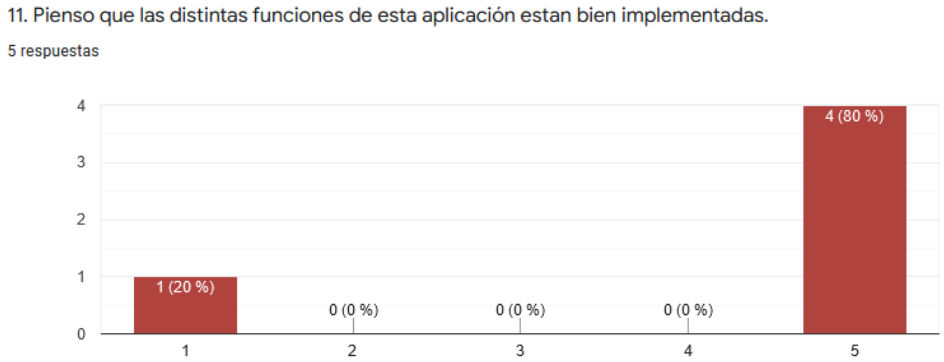


Figura 6.13: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 11 del cuestionario.

2. ¿Podría cumplir la aplicación con su objetivo educacional de servir de apoyo docente al profesor?
3. ¿Podría cumplir la aplicación con su objetivo de motivar a los alumnos?
4. ¿Está la aplicación diseñada adecuadamente?
5. ¿Existen errores que deban ser corregidos?
6. ¿Es la herramienta funcional desde un punto de vista de usabilidad?
7. ¿Que cambios y mejoras se podrían hacer a la demo de testeo para aumentar su atractivo de cara al usuario?
8. ¿Que cambios y mejoras se podrían hacer a la demo de testeo para aumentar su atractivo de cara aumentar su utilidad?

Y una breve conclusión a cada una de ellas:

1. Según las opiniones generales de los usuarios de prueba, si.
2. Según las opiniones generales de los usuarios de prueba, si.
3. Según las opiniones generales de los usuarios de prueba, si.
4. Hasta cierto punto. Hay algunos problemas con la inclusión de algunos elementos como el explorador de archivos a la hora de importar y exportar MIDI. Además, el diseño visual podría ser mejorado.
5. Existen algunos con los menús, así como a la hora de ejecutar la versión de MacOs.

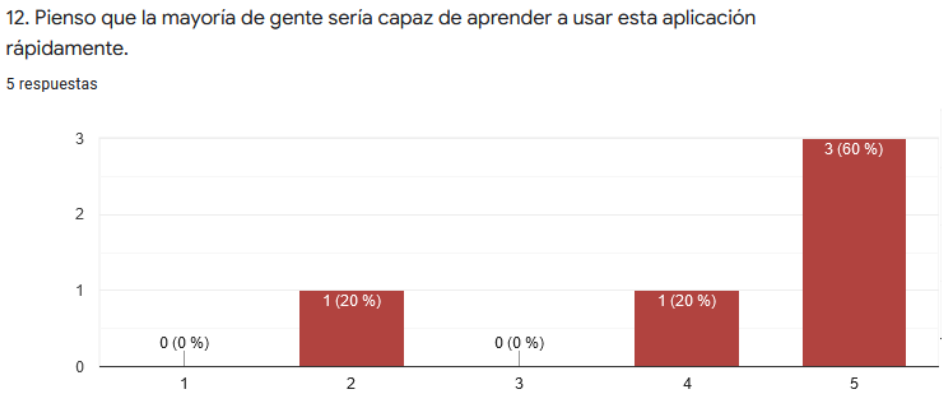


Figura 6.14: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 12 del cuestionario.

6. Gracias a la nota obtenida usando la puntuación SUS, podemos concluir de que es funcional, aunque sigue existiendo un gran espacio para la mejora.

7. Como se ha mencionado en el apartado anterior, cambios en el diseño visual y presentación, mejoras en el feedback, más opciones de resolución y más mensajes de aviso y alertas que orienten al usuario.

8. Los usuarios más expertos en el instrumento sugieren añadir una funcionalidad que permita trabajar los ejercicios con partituras, para así poder enseñar adicionalmente la lectura de estas.

Empezando por las conclusiones generales positivas, se puede sacar en claro el hecho de que Piano Unite es, para la mayoría del público, una aplicación con un concepto prometedor y sencilla de usar.

En ese aspecto, podemos concluir con que Piano Unite es una aplicación con un mayor enfoque a la posibilidad de trabajar en conjunción con un profesor de piano que otras alternativas similares como puede ser *Synthesia*. Esto se debe sobretodo a su posibilidad de grabar y reproducir los distintos ejercicios desde una misma aplicación de forma rápida y sencilla. Otro elemento que lo diferencia de esta aplicación, es la inclusión de sistemas que registran el progreso y rendimiento del usuario.

Sin embargo y como ya se ha sugerido anteriormente, la aplicación necesita unas mejoras en cuanto a lo que interfaz de usuario, animaciones, diseño visual y feedback se refiere.

Para mantener al público actualizado acerca de novedades, cambios y mejoras en la herramienta, se ha creado una entrada de Piano Unite en la web oficial de Narratech (`https://narratech.com/es/game/piano-unite/`).

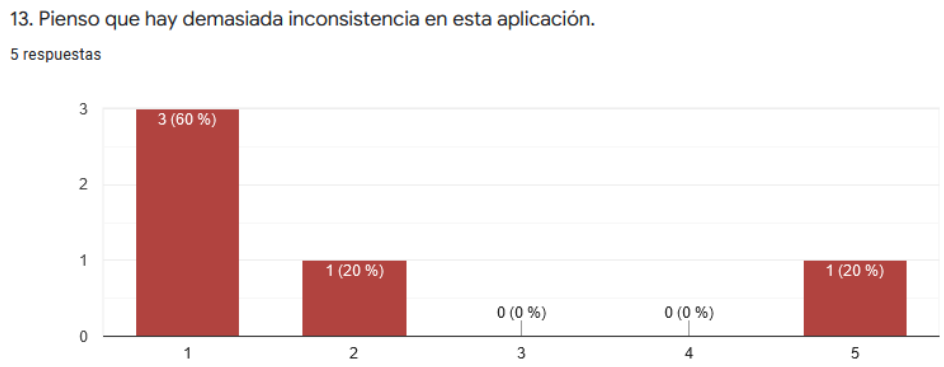


Figura 6.15: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 13 del cuestionario.

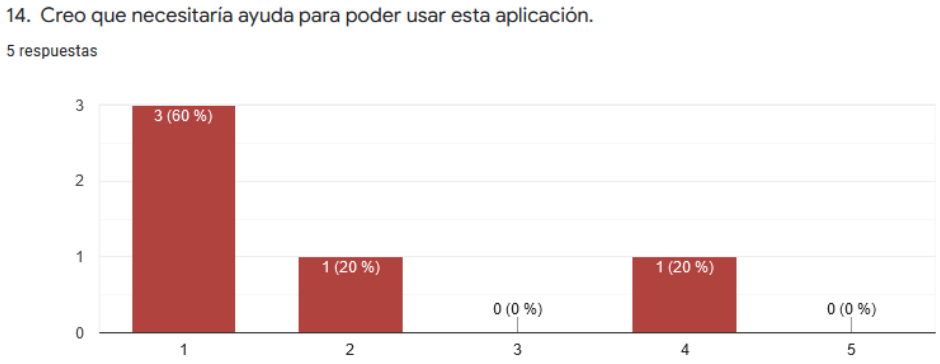


Figura 6.16: Gráfica que representa las opiniones de los sujetos de prueba ante la pregunta número 14 del cuestionario.

En el próximo y último capítulo de conclusiones (Capítulo 7), se explicarán más en detalle todos los elementos a tener en cuenta para continuar el desarrollo de esta herramienta, para que así pueda llegar a ser su mejor versión respecto a su utilidad para los usuarios y posible viabilidad comercial.

Le faltan opciones para ajustar la resolución de la aplicación y añadiría algo de feedback al grabar con el piano, al igual que al tocar un midi ves como se aproximan las distintas notas se podrían ver al grabarlas también.

Mejoraría el ataque y el desvanecimiento de los sonidos en las notas. El sistema de archivos me dio la impresión de no funcionar del todo bien. El hecho de que exista una versión para alumno y otra para profesor acentúa la capacidad didáctica de la aplicación. Como primera versión opino que tiene verdadero potencial para facilitar la enseñanza y la interacción profesor-alumno, especialmente de forma telemática y mediante ejercicios que se pueden dictar y exportar fácilmente.

Figura 6.17: Capturas de dos de las respuestas a la pregunta 15 del cuestionario.

Recomendaría avisar que es necesario un teclado o controlador midi de más de dos octavas para poder utilizarlo, siento que la idea es buena, y podría ser buena forma para enseñar a los alumnos a distancia o mandar tarea. Sin embargo, se pierde un factor muy importante si quieres aplicar esto en clases como tal: No hay partitura, no se aprende a tocar música ni a leerla, simplemente se enseña a replicar canciones con el piano como el que aprende con la guitarra por su cuenta. En rocksmith tiene algo de sentido porque en el juego se enseñan técnicas de guitarra y hasta en el próximo juego añadirán tablaturas. Si realmente se intenta que este juego se utilice para clases, habría que implementar eso. Las partituras en sibelius se pueden importar con midi, no se cómo de complejo puede ser hacer algo similar aquí, pero así tendría más sentido que haya una parte para el profe y otra para el alumno, ya que sí que se enseñaría algo más que simplemente tocar teclas de un piano.

Figura 6.18: Captura de pantalla de una de las respuestas a la pregunta 15 del cuestionario.

Capítulo 7

Conclusiones

En este último capítulo se va a repasar el concepto y los objetivos previamente propuestos para este proyecto y comentar, uno a uno, hasta qué punto se han conseguido cumplir. Se comentará, además, cuales han sido las nuevas competencias adquiridas durante el desarrollo del proyecto por parte del autor. Por último, se detallarán posibles mejoras y extensiones que se podrían hacer en el trabajo de cara al futuro, para que de esta manera *Piano Unite* pueda llegar a ser una herramienta útil para los usuarios y competente comercialmente hablando.

Los objetivos propuestos anteriormente son:

1. Identificar las oportunidades que ofrecen las nuevas tecnologías y la ludificación para mejorar la experiencia docente tanto de los profesores como de los alumnos en enseñanza elemental de piano.

2. Diseñar un sistema que transforme cualquier pieza musical en contenido audiovisual interactivo, disfrutable de manera intuitiva por un amplio espectro de jugadores realizando la digitación sobre un teclado de piano.

3. Desarrollar el prototipo de un videojuego educativo para ordenadores personales que aproveche el sistema anterior de manera que los profesores, por un lado, puedan crear ejercicios personalizados y los alumnos, por otro, practicar repetidamente con ellos y ser evaluados de manera automática.

4. Validar la utilidad y la usabilidad de este prototipo con usuarios reales, para poder plantear un proceso completo de producción. La aplicación debería ser conveniente e intuitiva para profesores y motivadora y entretenida para los alumnos. Pudiendo, de esta manera, llegar a ser un producto viable de cara al futuro.

7.1. Objetivos alcanzados

Ahora se comentará punto por punto el resultado de cada uno de los objetivos, tras haber completado el desarrollo de la herramienta:

1. Tras la investigación realizada en el capítulo de estado de la cuestión (Capítulo 2), opino que se han podido identificar una necesidad en cuanto a la integración de las nuevas tecnologías en el mundo de la enseñanza musical a fecha de publicación de este trabajo. Quizás la mayor conclusión que se puede sacar es la referente a la necesidad de crear aplicaciones y herramientas que permitan trabajar de forma telemática en este campo.

2. Este sistema ha sido desarrollado exitosamente. Funciona con una gran cantidad de tipos de archivos MIDI, y es capaz de detectar los 23 principales tipos de eventos a la hora de la lectura. A la hora de grabar MIDI, resulta ser preciso y eficiente, y puede ser fácilmente extendido para que pueda detectar aún más tipos de eventos en el futuro. Este sistema además, puede ser reciclado para el desarrollo de cualquier proyecto de *Unity* que trabaje con MIDI, lo cual puede ser una funcionalidad de gran utilidad para juegos de ritmo y juegos musicales. Por ello, estos dos scripts capaces de leer y escribir MIDI podrían incluso llegar a lanzarse como herramienta en la *Unity Asset Store* para facilitar ese trabajo a los desarrolladores.

3. La herramienta ha sido desarrollada, y ha resultado ser totalmente funcional como demo. Sin embargo, existen algunos campos en los que se podría ampliar para ofrecer un mejor producto, como puede ser realizar un mejor diseño visual y de efectos especiales; compatibilidad con más tipos de plataformas y sistemas operativos; o capacidad de conectividad online y sistemas de cuentas de usuario.

4. Este ha sido el objetivo que ha sido más complicado de cumplir exitosamente debido, sobretodo a la dificultad para poder acceder a un número y una variedad significativa de usuarios de prueba. Aunque el consenso general es que la aplicación puede llegar a ser muy útil, también se coincide en que posee algunos problemas en su diseño. Especialmente estos problemas hacen referencia la accesibilidad, el feedback y la comodidad por parte del usuario a la hora de usarla.

Aunque en diseño Piano Unite pueda resultar similar a otras aplicaciones como Synthesia, existe una diferencia clave con respecto al resto de la competencia, y es la capacidad de poder grabar piezas y ejercicios desde la propia aplicación. De esta manera, se obtiene una herramienta que no necesita del uso de programas externos para su uso, lo que puede llegar a resultar mucho

más cómodo para los usuarios. Además, la visualización de los resultados y de los errores cometidos a lo largo del ejercicio es más precisa que en otras.

El proyecto ha sido publicado en la plataforma de Itch.io (`https://narratech.itch.io/piano-unite`), por lo que queda completamente disponible para ser usada de forma gratuita por cualquiera.

7.2. Trabajo futuro

Este proyecto puede seguir ampliándose por varios campos distintos:

- Añadiendo **Funcionalidades on-line**. Se podría ampliar *Piano Unite* para convertirse en una herramienta on-line, donde alumnos y profesores puedan loggear y poseer una cuenta personal. Al igual que en aplicaciones como *Duolingo*, el profesor podría administrar a todos sus alumnos a través de ella, así como su progreso y los ejercicios que deben realizar. Los alumnos por otro lado, podrían compartir sus logros y puntuaciones con el resto, dando lugar a una sana competitividad.

- Complementando con lecciones de **teoría musical y lectura de partituras**. Se podrían añadir modos de juego que, en vez de ir enfocados en al mera digitación, puedan servir para adquirir otras competencias. Desde leer partituras a tiempo real, hasta entender otros conceptos de teoría musical, como por ejemplo la armonía.

- Mejorando su **diseño artístico y visual**. Mejorando el ya existente, y añadiendo animaciones, efectos de partículas, filtros o incluso algún tipo de avatar para los usuarios.

- Poniendo a prueba su efectividad en **entornos educativos reales**.Esto consistiría en llevar la aplicación a academias, conservatorios y clases particulares durante un periodo de tiempo, para así estudiar sus repercusiones. Lo ideal sería que esta facilitara el trabajo de los docentes a la vez que motiva y divierte a los alumnos.

7.3. Reflexión final

A partir de este punto comenzaré a desarrollar las opiniones personales sobre el desarrollo de este proyecto, por lo que me permitiré hablar en primera persona.

Como único autor de este proyecto he quedado bastante contento con los resultados y como se han abordado los distintos objetivos. Pienso que con un grupo de desarrollo más grande se podría haber conseguido una herramienta más completa y lista para su uso comercial, ya que se podrían haber podido

dividir mejor los esfuerzos entre desarrollo, diseño visual y contacto más estrecho y amplio con el público objetivo.

Aún así, opino que el proceso de desarrollo de este proyecto ha sido enormemente enriquecedor. He tenido que pasar por todas las fases: La de concepción, el prototipo, realizar y redactar toda la investigación realizada, diseñar y desarrollar una aplicación con fines educativos, diseñar y gestionar las pruebas de usuario, corregir errores y plasmar todo ese trabajo en este escrito. El hecho de haber tenido que abordar sólo todas ellas me ha aportado una experiencia increíble de cara a comprender lo que supone desarrollar un juego serio de tales características. Sin duda me ha permitido poner en práctica muchísimos de los conocimientos y competencias adquiridos a lo largo de la carrera. Además, como la mayoría de proyectos grandes a lo largo del grado han sido realizados en grupos, el tener que desarrollar este por mi cuenta pienso que ha sido muy enriquecedor, y me ha aportado mucha autonomía y disciplina.

Appendix A

Conclusions

This last chapter will review the concept and objectives previously proposed for this project and comment, one by one, to what extent they have been achieved. It will also comment on the new competences acquired by the author during the development of the project. Finally, possible improvements and extensions that could be made to the work in the future will be detailed, so that Piano Unite can become a useful tool for users and commercially competent.

The objectives were:

1. To identify the opportunities offered by new technologies and gamification to improve the teaching experience of both teachers and students in elementary piano teaching.

2. To design a system that transforms any piece of music into interactive audiovisual content, intuitively enjoyed by a wide range of players performing fingering on a piano keyboard.

3. To develop the prototype of an educational video game for personal computers that takes advantage of the above system so that teachers, on the one hand, can create personalised exercises and students, on the other hand, can repeatedly practice them and be automatically evaluated.

4. Validate the usefulness and usability of this prototype with real users, in order to consider a complete production process. The application should be convenient and intuitive for teachers and motivating and entertaining for students. In this way, it could become a viable product for the future.

A.1. Achieved objectives

The outcome of each of the objectives will now be discussed point by point, after having completed the development of the tool:

1. After the research carried out in the chapter on the state of the question (chapter 2), I believe that it has been possible to identify a need in terms of the integration of new technologies in the world of music teaching at the date of publication of this work. Perhaps the main conclusion that can be drawn is the need to create applications and tools that allow telematic work in this field.

2. This system has been successfully developed. It works with a large number of MIDI file types, and is able to detect the 23 main event types when reading. When recording MIDI, it proves to be accurate and efficient, and can be easily extended to detect even more event types in the future. This system can also be recycled for the development of any project in *Unity* that works with MIDI, which can be a very useful functionality for rhythm games and music games. Therefore, these two scripts capable of reading and writing MIDI could even be released as a tool in the *Unity Asset Store* to make it easier for developers to do so.

3. The tool has been developed, and has proven to be fully functional as a demo. However, there are some areas where it could be expanded to offer a better product, such as better visual design and special effects; compatibility with more types of platforms and operating systems; or online connectivity and user account systems.

4. This has been the most difficult objective to achieve successfully, mainly due to the difficulty of accessing a significant number and variety of test users. Although the general consensus is that the application can be very useful, it is also agreed that it has some problems in its design. In particular, these problems relate to accessibility, feedback and user comfort.

Although in design Piano Unite may be similar to other applications such as Synthesia, there is a key difference with respect to the rest of the competition, and that is the ability to record pieces and exercises from within the application itself. In this way, you get a tool that does not require the use of external programs for its use, which can be much more convenient for users. In addition, the visualisation of the results and the errors made throughout the exercise is more precise than in others.

The project has been published on the Itch.io (`https://narratech.itch.io/piano-unite`) website, making it fully available for free use by anyone.

A.2. Future work

This project can be further expanded in a number of different fields:

- Adding **Online functionalities**. *Piano Unite* could be extended to become an on-line tool, where students and teachers could log in and have a personal account. As in applications such as *Duolingo*, the teacher could manage all his students through it, as well as their progress and the exercises they have to do. The students, on the other hand, could share their achievements and scores with the rest, giving rise to a healthy competition.

- Complementing this with lessons in **music theory and score reading**. Play modes could be added which, instead of focusing on mere fingering, could be used to acquire other skills. From reading scores in real time, to understanding other music theory concepts, such as harmony.

- Improving its **artistic and visual design**. Improving the existing one, and adding animations, particle effects, filters or even some kind of avatar for users.

- Testing its effectiveness in **real educational environments**. This would consist of taking the application to academies, conservatories and private classes for a period of time, in order to study its repercussions. Ideally, it would facilitate the work of teachers while motivating and entertaining students.

A.3. Final thoughts

From this point I will start to develop personal opinions on the development of this project, so I will allow myself to speak in the first person.

As the sole author of this project I have been quite happy with the results and how the different objectives have been addressed. I think that with a larger development group a more complete and commercially ready tool could have been achieved, as the efforts could have been better divided between development, visual design and closer and wider contact with the target audience.

Still, I think the development process of this project has been enormously enriching. I have had to go through all the phases: The conception phase, the prototyping phase, conducting and writing up all the research done, designing and developing an application for educational purposes, designing and managing the user tests, correcting errors and translating all that work into this paper. The fact that I had to deal with all of them on my own has

given me an incredible experience in terms of understanding what it means to develop a serious game of such characteristics. It has certainly allowed me to put into practice a lot of the knowledge and skills I have acquired throughout my studies. In addition, as most of the big projects throughout the degree have been carried out in groups, the fact of having to develop this one on my own has been very enriching, and has given me a lot of autonomy and discipline.

Apéndice B

Manual de uso

En este apéndice se explicará cómo descargar, ejecutar y poder hacer uso de la aplicación Piano Unite. Esta será una sección breve ya que, en principio, se trata de una aplicación sencilla de usar.

Los requisitos para poder usar la aplicación son los siguientes: Tener Windows (7, 10, 11), un controlador MIDI (preferiblemente un teclado) y altavoces o auriculares.

Para descargar la última versión de Piano Unite, hay que acceder al siguiente enlace:

Luego, habrá que pinchar en el botón de descargar de la versión que prefieras (para profesores o para alumnos). Una vez descargado, se podrá encontrar un archivo llamado PianoUnite.zip. Habrá que descomprimir este archivo. Una vez descomprimido, ejecuta el archivo PianoUnite(prof).exe o PianoUnite(est).exe, dependiendo de la versión que hayas escogido. Recuerda que debes conectar el teclado MIDI. Es posible que necesites dar permisos especiales a la aplicación para poder ejecutarla.

Una vez estés dentro de la aplicación, puedes hacerla uso tal y como prefieras. Se puede empezar creando algún ejercicio presionando el botón de REC. También se puede empezar pulsando el icono de Play y realizando alguno de los ejercicios que vienen ya cargados, o descargando algún archivo MIDI de piano y seleccionándolo desde el propio explorador de archivos.

A continuación, se incluirán una serie de capturas de pantalla de Piano Unite.



Figura B.1: Menú principal.



Figura B.2: Diferentes fases de la escena de grabación. Orden: De izquierda a derecha y de arriba a abajo.

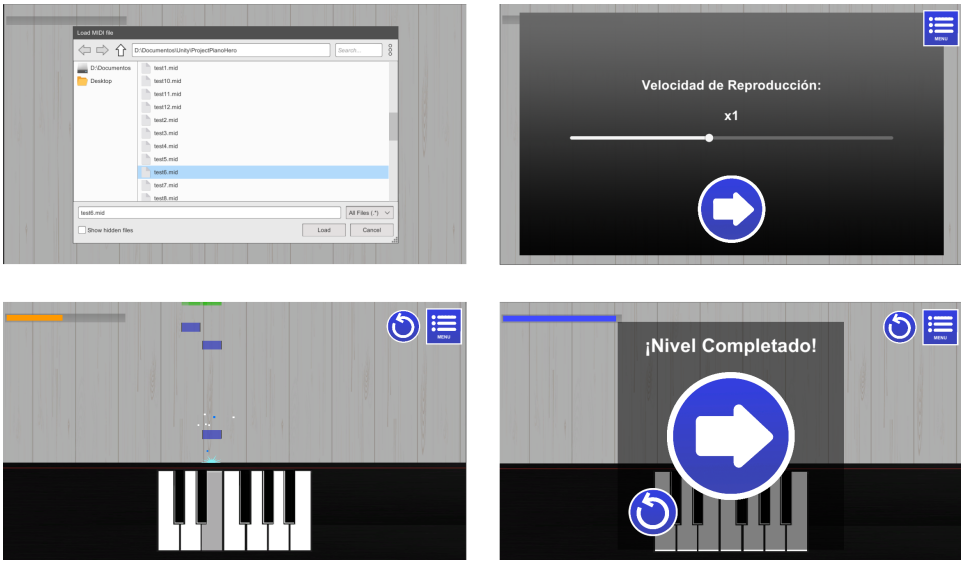


Figura B.3: Diferentes fases de la escena de resolución de ejercicios. Orden: De izquierda a derecha y de arriba a abajo.

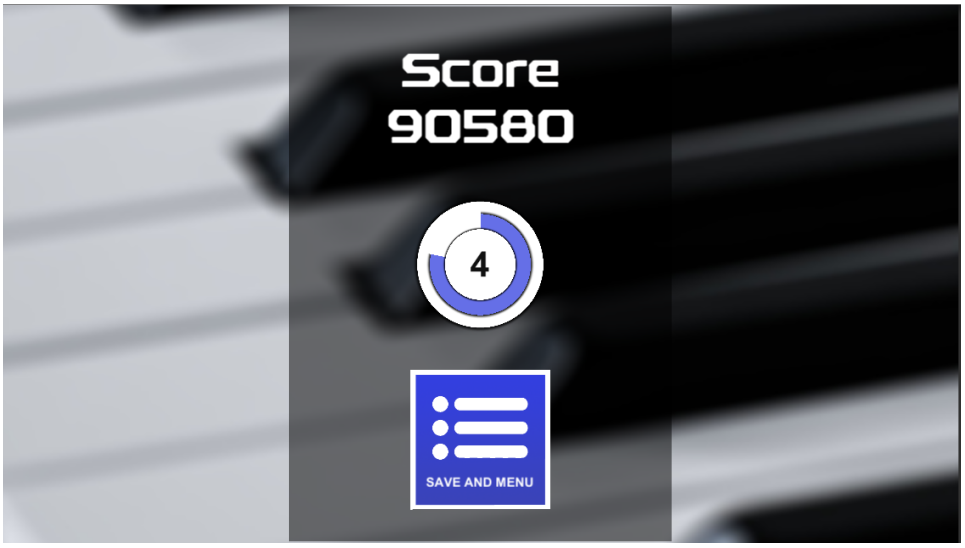


Figura B.4: Escena de puntuación.



Figura B.5: Pantalla de historial de partidas.



Figura B.6: Menú de opciones de sonido.

Apéndice C

Tabla Completa de Bytes de estado MIDI

Table 2: Expanded Status Bytes List			
STATUS BYTE		DATA BYTES	
1st Byte Value Binary |Hex| Dec	Function	2nd Byte	3rd Byte
10000000= 80= 128	Chan 1 Note off	Note Number (0-127)	Note Velocity (0-127)
10000001= 81= 129	Chan 2 Note off	Note Number (0-127)	Note Velocity (0-127)
10000010= 82= 130	Chan 3 Note off	Note Number (0-127)	Note Velocity (0-127)
10000011= 83= 131	Chan 4 Note off	Note Number (0-127)	Note Velocity (0-127)
10000100= 84= 132	Chan 5 Note off	Note Number (0-127)	Note Velocity (0-127)
10000101= 85= 133	Chan 6 Note off	Note Number (0-127)	Note Velocity (0-127)
10000110= 86= 134	Chan 7 Note off	Note Number (0-127)	Note Velocity (0-127)
10000111= 87= 135	Chan 8 Note off	Note Number (0-127)	Note Velocity (0-127)
10001000= 88= 136	Chan 9 Note off	Note Number (0-127)	Note Velocity (0-127)
10001001= 89= 137	Chan 10 Note off	Note Number (0-127)	Note Velocity (0-127)
10001010= 8A= 138	Chan 11 Note off	Note Number (0-127)	Note Velocity (0-127)
10001011= 8B= 139	Chan 12 Note off	Note Number (0-127)	Note Velocity (0-127)
10001100= 8C= 140	Chan 13 Note off	Note Number (0-127)	Note Velocity (0-127)
10001101= 8D= 141	Chan 14 Note off	Note Number (0-127)	Note Velocity (0-127)
10001110= 8E= 142	Chan 15 Note off	Note Number (0-127)	Note Velocity (0-127)
10001111= 8F= 143	Chan 16 Note off	Note Number (0-127)	Note Velocity (0-127)
10010000= 90= 144	Chan 1 Note on	Note Number (0-127)	Note Velocity (0-127)
10010001= 91= 145	Chan 2 Note on	Note Number (0-127)	Note Velocity (0-127)
10010010= 92= 146	Chan 3 Note on	Note Number (0-127)	Note Velocity (0-127)

10010011= 93= 147	Chan 4 Note on	Note Number (0-127)	Note Velocity (0-127)
10010100= 94= 148	Chan 5 Note on	Note Number (0-127)	Note Velocity (0-127)
10010101= 95= 149	Chan 6 Note on	Note Number (0-127)	Note Velocity (0-127)
10010110= 96= 150	Chan 7 Note on	Note Number (0-127)	Note Velocity (0-127)
10010111= 97= 151	Chan 8 Note on	Note Number (0-127)	Note Velocity (0-127)
10011000= 98= 152	Chan 9 Note on	Note Number (0-127)	Note Velocity (0-127)
10011001= 99= 153	Chan 10 Note on	Note Number (0-127)	Note Velocity (0-127)
10011010= 9A= 154	Chan 11 Note on	Note Number (0-127)	Note Velocity (0-127)
10011011= 9B= 155	Chan 12 Note on	Note Number (0-127)	Note Velocity (0-127)
10011100= 9C= 156	Chan 13 Note on	Note Number (0-127)	Note Velocity (0-127)
10011101= 9D= 157	Chan 14 Note on	Note Number (0-127)	Note Velocity (0-127)
10011110= 9E= 158	Chan 15 Note on	Note Number (0-127)	Note Velocity (0-127)
10011111= 9F= 159	Chan 16 Note on	Note Number (0-127)	Note Velocity (0-127)
10100000= A0= 160	Chan 1 Polyphonic Aftertouch	Note Number (0-127)	Pressure (0-127)
10100001= A1= 161	Chan 2 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100010= A2= 162	Chan 3 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100011= A3= 163	Chan 4 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100100= A4= 164	Chan 5 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100101= A5= 165	Chan 6 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100110= A6= 166	Chan 7 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10100111= A7= 167	Chan 8 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)

10101000= A8= 168	Chan 9 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101001= A9= 169	Chan 10 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101010= AA= 170	Chan 11 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101011= AB= 171	Chan 12 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101100= AC= 172	Chan 13 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101101= AD= 173	Chan 14 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101110= AE= 174	Chan 15 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10101111= AF= 175	Chan 16 Polyphonic Aftertouch	Note Number (0-127	Pressure (0-127)
10110000= B0= 176	Chan 1 Control/Mode Change	see Table 3	see Table 3
10110001= B1= 177	Chan 2 Control/Mode Change	see Table 3	see Table 3
10110010= B2= 178	Chan 3 Control/Mode Change	see Table 3	see Table 3
10110011= B3= 179	Chan 4 Control/Mode Change	see Table 3	see Table 3
10110100= B4= 180	Chan 5 Control/Mode Change	see Table 3	see Table 3
10110101= B5= 181	Chan 6 Control/Mode Change	see Table 3	see Table 3
10110110= B6= 182	Chan 7 Control/Mode Change	see Table 3	see Table 3
10110111= B7= 183	Chan 8 Control/Mode Change	see Table 3	see Table 3
10111000= B8= 184	Chan 9 Control/Mode Change	see Table 3	see Table 3
10111001= B9= 185	Chan 10 Control/Mode Change	see Table 3	see Table 3
10111010= BA= 186	Chan 11 Control/Mode Change	see Table 3	see Table 3
10111011= BB= 187	Chan 12 Control/Mode Change	see Table 3	see Table 3
10111100= BC= 188	Chan 13 Control/Mode Change	see Table 3	see Table 3

10111101= BD= 189	Chan 14 Control/Mode Change	see Table 3	see Table 3
10111110= BE= 190	Chan 15 Control/Mode Change	see Table 3	see Table 3
10111111= BF= 191	Chan 16 Control/Mode Change	see Table 3	see Table 3
11000000= C0= 192	Chan 1 Program Change	Program # (0-127)	none
11000001= C1= 193	Chan 2 Program Change	Program # (0-127)	none
11000010= C2= 194	Chan 3 Program Change	Program # (0-127)	none
11000011= C3= 195	Chan 4 Program Change	Program # (0-127)	none
11000100= C4= 196	Chan 5 Program Change	Program # (0-127)	none
11000101= C5= 197	Chan 6 Program Change	Program # (0-127)	none
11000110= C6= 198	Chan 7 Program Change	Program # (0-127)	none
11000111= C7= 199	Chan 8 Program Change	Program # (0-127)	none
11001000= C8= 200	Chan 9 Program Change	Program # (0-127)	none
11001001= C9= 201	Chan 10 Program Change	Program # (0-127)	none
11001010= CA= 202	Chan 11 Program Change	Program # (0-127)	none
11001011= CB= 203	Chan 12 Program Change	Program # (0-127)	none
11001100= CC= 204	Chan 13 Program Change	Program # (0-127)	none
11001101= CD= 205	Chan 14 Program Change	Program # (0-127)	none
11001110= CE= 206	Chan 15 Program Change	Program # (0-127)	none
11001111= CF= 207	Chan 16 Program Change	Program # (0-127)	none
11010000= D0= 208	Chan 1 Channel Aftertouch	Pressure (0-127)	none
11010001= D1= 209	Chan 2 Channel Aftertouch	Pressure (0-127)	none

11010010= D2= 210	Chan 3 Channel Aftertouch	Pressure (0-127)	none
11010011= D3= 211	Chan 4 Channel Aftertouch	Pressure (0-127)	none
11010100= D4= 212	Chan 5 Channel Aftertouch	Pressure (0-127)	none
11010101= D5= 213	Chan 6 Channel Aftertouch	Pressure (0-127)	none
11010110= D6= 214	Chan 7 Channel Aftertouch	Pressure (0-127)	none
11010111= D7= 215	Chan 8 Channel Aftertouch	Pressure (0-127)	none
11011000= D8= 216	Chan 9 Channel Aftertouch	Pressure (0-127)	none
11011001= D9= 217	Chan 10 Channel Aftertouch	Pressure (0-127)	none
11011010= DA= 218	Chan 11 Channel Aftertouch	Pressure (0-127)	none
11011011= DB= 219	Chan 12 Channel Aftertouch	Pressure (0-127)	none
11011100= DC= 220	Chan 13 Channel Aftertouch	Pressure (0-127)	none
11011101= DD= 221	Chan 14 Channel Aftertouch	Pressure (0-127)	none
11011110= DE= 222	Chan 15 Channel Aftertouch	Pressure (0-127)	none
11011111= DF= 223	Chan 16 Channel Aftertouch	Pressure (0-127)	none
11100000= E0= 224	Chan 1 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100001= E1= 225	Chan 2 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100010= E2= 226	Chan 3 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100011= E3= 227	Chan 4 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100100= E4= 228	Chan 5 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100101= E5= 229	Chan 6 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100110= E6= 230	Chan 7 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)

11100111= E7= 231	Chan 8 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101000= E8= 232	Chan 9 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101001= E9= 233	Chan 10 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101010= EA= 234	Chan 11 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101011= EB= 235	Chan 12 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101100= EC= 236	Chan 13 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101101= ED= 237	Chan 14 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101110= EE= 238	Chan 15 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11101111= EF= 239	Chan 16 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11110000= F0= 240	System Exclusive	**	**
11110001= F1= 241	MIDI Time Code Qtr. Frame	-see spec-	-see spec-
11110010= F2= 242	Song Position Pointer	LSB	MSB
11110011= F3= 243	Song Select (Song #)	(0-127)	none
11110100= F4= 244	Undefined (Reserved)	---	---
11110101= F5= 245	Undefined (Reserved)	---	---
11110110= F6= 246	Tune request	none	none
11110111= F7= 247	End of SysEx (EOX)	none	none
11111000= F8= 248	Timing clock	none	none
11111001= F9= 249	Undefined (Reserved)	---	---
11111010= FA= 250	Start	none	none
11111011= FB= 251	Continue	none	none

11111100= FC= 252	Stop	none	none
11111101= FD= 253	Undefined (Reserved)	---	---
11111110= FE= 254	Active Sensing	none	none
11111111= FF= 255	System Reset	none	none
** Note: System Exclusive (data dump) 2nd byte= Vendor ID (or Universal Exclusive) followed by more data bytes and ending with EOX.			

Bibliografía

Ableton AG. Ableton. 1999.

Abt, C. *Serious Games*. Viking Press, 1970.

Adobe Systems Incorporated. Adobe photoshop. Disponible en https://www.adobe.com/es/products/photoshop.

AEVI. *La Industria del Videojuego en España Anuario 2019*. LLYC, Disponible en http://www.aevi.org.es/web/wp-content/uploads/2020/04/AEVI-ANUARIO-2019.pdf.

Altozano, J. Musihacks curso de piano básico. Disponible en https://www.musihacks.com/aprende-a-tocar-el-piano.

Aslim. Aslim fnpt alx. Disponible en https://www.alsim.com/simulators/alx-fnpt-ii-mcc-mpl-flight-training-device/.

Bloom, B. y Krathwohl, D. R. Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners.handbook i: Cognitive domain. 1956.

Bloom, B. y Krathwohl, D. R. A taxonomy for learning, teaching and assessing: a revision of bloom's taxonomy of educational objectives. 2001.

Calderón, C. Enseñanza presencial y virtual en la educación musical. Disponible en https://www.researchgate.net/publication/39208520_Ensenanza_presencial_y_virtual_en_la_educacion_musical.

Chen, J. Flow in games(and everything else). *Communications of the ACM*, Disponible en http://jenovachen.com/flowingames/p31-chen.pdf.

CodeCombat Inc. Code combat. Disponible en https://codecombat.com/.

Collins, K. An introduction to procedural music in video games. *Contemporary Music Review*, Disponible en https://www.researchgate.net/publication/232925458_An_Introduction_to_Procedural_Music_in_Video_Games.

Dannenberg, R. y Mazzoni, D. Audacity. Disponible en https://audacity.es/.

De Aguilera M and Mendiz A. Videogames and education: (education in the face of a "parallel school"). 2003.

Duolingo Inc. English test duolingo. Disponible en https://englishtest.duolingo.com/.

Duolingo Inc. Duolingo. Microsoft Windows, macOS, IOS, Android, Windows Phone, 2011.

DVD FORUM. Hddvd. Disponible en http://www.dvdforum.org/forum.shtml.

Dörner, R., Göbel, S., Effelsberg, W. y Wiemeyer, J. *Serious Games: Foundations, Concepts and Practice*. Springer International Publishing, 2016.

Easy Game Studio. count down. Disponible en https://assetstore.unity.com/packages/tools/gui/count-down-127390.

Firelight Technologies. fmod. 1995.

FlowKey. Disponible en https://www.flowkey.com.

Free Pats Project. Free pats project. Disponible en https://freepats.zenvoid.org/Piano/acoustic-grand-piano.html.

Gee, J. P. *Good Video Games and Good Learning*. University of Wisconsin-Madison, Disponible en https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.585.3385&rep=rep1&type=pdf.

Google Drive. Google meet. Disponible en https://drive.google.com.

Google Meet. Google meet. Disponible en https://meet.google.com/.

Ibáñez, M. L. *Producción musical y diseño de sonido para videojuegos*. Editorial Síntesis, 2021.

Image-Line Software. Fl studio 20.0.9. Disponible en https://www.image-line.com/.

JoyTunes. Disponible en https://www.joytunes.com/piano/song.php.

JoyTunes. Disponible en https://www.joytunes.com/simply-piano.

Keijiro Takahashi. Midijack. PC, macOS, Disponible en https://github.com/keijiro/MidiJack.

Klopfer, E., Haas, J., Osterweil, S. y Louisa Rosenheck. *Design Principles for Learning Games that Connect Hearts, Minds and the Everyday*. MIT Press, Disponible en `https://www.resonant.games/`.

Markus Schwenk. Midieditor 3.3.0. Disponible en `https://midieditor.org`.

Martín-Laborda, R. *Las nuevas tecnologías en la educación*. Fundación Auna, primera edición, 2005.

Maxis. Spore. Microsoft Windows, MacOS, Android, IOS, Nintendo DS, 2008.

Microsoft. Github desktop 3.0.0. Disponible en `https://github.com/`.

Microsoft. Visual studio 2019 version 4.8.04084. Disponible en `https://visualstudio.microsoft.com/`.

MIDI Association. Disponible en `https://www.midi.org/specifications-old/item/table-2-expanded-messages-list-status-bytes`.

Narratech Laboratories. Narratech laboratories. Disponible en `https://narratech.com`.

Native Instruments. Guitar rig 6 demo. Disponible en `https://www.native-instruments.com/es/products/komplete/guitar/guitar-rig-6-pro/`.

Neversoft. Guitar hero 3: Legends of rock. 2007.

Nikolaev, A. *Escuela rusa de piano 1º*. European Music Center (EMC), 2007.

OBS Project. Open broadcaster software. Disponible en `https://obsproject.com/es/download`.

Pérez, M. G. Herramienta para el aprendizaje de técnicas de mezcla en producción musical. Disponible en `https://eprints.ucm.es/id/eprint/56575/`.

Roldán Alcázar, M. Análisis de los métodos para la enseñanza del piano en los primeros niveles. *Revista Hekademos*, Disponible en `https://dialnet.unirioja.es/servlet/articulo?codigo=3745814`.

Serious Games Society. Serious games society. Disponible en `https://seriousgamessociety.org`.

Shell, J. *The Art of Game Design: A book of lenses*. CRC Press, 2008.

Slack Technologies. Slack 4.2. Disponible en `https://slack.com/`.

Sony, Philips, TDK, Microsoft, BDA. Blue ray disc. 2005.

Steinberg Technologies. Steinberg. Disponible en `https://www.steinberg.net/technology/`.

Synthesia LLC. Rocksmith 2014. Microsoft Windows, macOS, Android, IOS, 2006.

Tchokv/Gemiu. *El Piano. Iniciación a la música*. Real Musical - Carisch España, 1994.

The Music Commission. *Retuning Our Ambition for Music Learning. Every child taking music further*. The Music Commission, Disponible en `http://www.musiccommission.org.uk/`.

Ubisoft. Rocksmith 2014. Microsoft Windows, macOS, PS4, PS3, Xbox One, Xbox 360, 2014.

Unity Technologies. Unity. PC, Disponible en `https://unity.com/`.

Valbuena, C. Aprender música online. Disponible en `https://www.aprendermusicaonline.com/cursos-de-piano`.

yasirkula. Runtime file browser. Disponible en `https://assetstore.unity.com/packages/tools/gui/runtime-file-browser-113006#releases`.

Zichermann, G. y Cunningham, C. *Gamification by Design*. O'Reilly Media, INC., 2011.