UNIVERSITY OF CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

COMPUTER SCIENCE DEGREE

FINAL DEGREE PROJECT

HoloMusic XP: Gamification-based System for Teaching Music and Piano using Mixed Reality

Diego Molero Marín

July, 2018

HoloMusic XP: Gamification-based System for Teaching Music and Piano using Mixed Reality

UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

Technologies and Information Systems Department

SPECIFIC TECHNOLOGY OF
INFORMATION AND COMMUNICATION TECHNOLOGIES

FINAL DEGREE PROJECT

HoloMusic XP: Gamification-based System for Teaching Music and Piano using Mixed Reality

Author: Diego Molero Marín

Supervisor: Santiago Sánchez Sobrino

Supervisor: David Vallejo Fernández

July, 2018

Diego Molero Marín

Ciudad Real – Spain

E-mail: diego.molero@alu.uclm.es
Telephone: 606 613 709
Web site: `diegomolero.github.io`

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE	VOCAL	SECRETARIO
Fdo.:	Fdo.:	Fdo.:

Resumen

La música es considerada por muchas culturas como uno de los pilares de la educación. Aprender a interpretar música a través de un instrumento puede aportar muchos beneficios al individuo. Sin embargo, esto requiere de muchas horas de estudio y práctica.

Con el desarrollo de las tecnologías cada vez son más las soluciones que se encuentran en cualquier ámbito a problemas que antes no tenían solución. Concretamente, en el campo de la música, se ha experimentado un gran cambio desde que emergieron las tecnologías digitales. Este cambio no solo a afectado a la música que escuchamos sino también a la forma de aprenderla. Algunos sistemas surgen con el objetivo de complementar la función del profesor en el aprendizaje de la música.

Una de las tecnologías que mayor impacto va a tener en la educación es la realidad mixta. Esta tecnología permite crear nuevas formas de aprender mediante la interacción y visualización de conceptos que hasta ahora eran se consideraban abstractos como la música. A través de un dispositivo de realidad mixta, como las Microsoft HoloLens, es posible ver la música y entenderla de manera distinta. Las Microsoft HoloLens son un dispositivo similar a unas gafas capaz de proyectar imágenes en la retina produciendo así la visualización de hologramas en el entorno.

Con el objetivo de reforzar las cualidades educativas de un sistema, algunas aplicaciones implementan métodos de gamificación. Estos métodos incorporan elementos normalmente vinculados a los juegos para así captar la atención y motivar al estudiante. El uso correcto de estos métodos en un sistema educativo puede mejorar su efectividad.

Sobre este contexto, se plantea el diseño y desarrollo de un sistema de un alto grado de innovación para las Microsoft HoloLens con el objetivo de enseñar música y piano. Para ello se utilizarán técnicas de realidad mixta que, complementadas con técnicas de gamificación, guiarán al estudiante en el aprendizaje de la música y el piano.

Abstract

Music is considered by many cultures to be one of the pillars of education. Learning how to play music through an instrument can bring many benefits to the individual. However, this requires many hours of study and practice.

With the development of technologies, more and more solutions are being found in any field to problems that previously were unsolvable. Specifically, in the field of music, a big change has been experienced since the emergence of digital technologies. This change has not only affected the music we listen to but also the way we learn it. Some systems arise with the aim of complementing the role of the teacher in the learning of music.

One of the technologies that is going to have a major impact on education is the mixed reality. This technology allows the creation of new ways of learning through the interaction and visualisation of concepts that until now were considered abstract, such as music. Through a mixed reality device, such as Microsoft HoloLens, it is possible to see music and understand it in a different way. The Microsoft HoloLens is a device similar to a pair of glasses capable of projecting images onto the retina, thus producing the visualisation of holograms in the environment.

With the aim of strengthening the educational qualities of a system, some applications implement gamification methods. These methods incorporate elements normally linked to the games in order to capture the attention and motivate the student. The correct use of these methods in an education system can improve their effectiveness.

In this context, it is proposed the design and development of a highly innovative system for the Microsoft HoloLens with the aim of teaching music and piano. For this purpose, mixed reality techniques will be used which, supplemented with gamification techniques, will guide the student in the learning of music and piano.

Agradecimientos

Este proyecto es el fruto de toda la educación que he recibido. Por ello debo agradecer especialmente a mis padres. Ellos crearon la base de mi educación y me enseñaron a crecer como persona. Gran parte del mérito de este trabajo se lo debo a ellos. Gracias por haberme permitido formarme tanto en la música como en la informática, abriéndome las puertas a un futuro mejor. Quiero dedicaros este trabajo y que os sintáis orgullosos de mí.

Gracias a mi familia y a mi novia por haberme apoyado en todo momento facilitándome el camino. Agradezco entre lágrimas a todos los que ahora no están pero estarán siempre en mis recuerdos porque ellos también han sido muy importantes para mí.

Gracias a mis profesores del conservatorio de música Marcos Redondo por haberme enseñado más allá de la música y por ayudarme a desarrollar mi faceta personal.

Gracias a los que me han acompañado en las noches de estudio en la biblioteca, a mis compañeros de Furious Koalas y del laboratorio, a mis amigos, a mis profesores de las escuela superior de informática, en especial a mis directores de proyecto.

Diego

A mi madre y a mi padre.

Contents

List of Tables

List of Figures

Listing index

List of Acronyms

AR	Augmented Reality
PPP	Point-to-Point Protocol
OS	Operative System
USB	Universal Serial Bus
BLE	Bluetooth Low Energy
SSH	Secure Shell
MIDI	Musical Instrument Digital Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
OSI	Open System Interconnection
UWP	Universal Windows Platform
API	Application Programming Interface
SDK	Software Development Kit
QR	Quick Response
PC	Personal Computer
GUI	Graphical User Interface
LCD	Liquid Crystal Display
MR	Mixed Reality
VC	Virtuality Continuum Scale
VR	Virtual Reality
HMD	Head Mounted Display
OHMD	Optic Head Mounted Display
IMU	Inertial Measurement Unit
VPL	Virtual Programming Languages
VCASS	Visually Coupled Airborne System Simulator

CPU	Central Processing Unit
OPENCV	Open Source Computer Vision
BSD	Berkeley Software Distribution
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
GPS	Global Positioning System
UI	User Interface
HPU	Holographic Processing Unit
TPM	Trusted Platform Module
IPD	Interpupillary Distance
FPS	First Person Shooter
DLL	Dynamic Link Library
XAML	eXtensible Application Markup Language
D3D	Direct3D
IDE	Integrated Development Environment
MRTK	Mixed Reality ToolKit
GPU	Graphics Processing Unit
UX	User Experience
ISO	International Organization for Standardization
SH	Shell Script
UML	Unified Modeling Language
XML	Extensible Markup Language
URL	Uniform Resource Locator
BPM	Beat Per Minute

Chapter 1

Introduction

THe music has accompanied the human being since the very beginning, becoming one of the first artistic expression forms. The music has not been separated from our history since then. According to the philosopher *Nietzsche*, *"Without music, life would be a mistake"*. Music as an art has evolved in form and style over time. The media by which music is transmitted has also evolved. The musical instrument that has always accompanied music is the voice. This was followed by percussion instruments and later melodic instruments.

One of the most popular musical instruments today is the piano. The first piano dates back to approximately 1700, in Italy. Although it is relatively modern, it had many predecessors that used the same mechanics based on using keys to sound strings. Thanks to the great composers of the 18th century such as *Mozart*, the piano began to gain importance as a solo instrument and its popularity increased. One of the advantages of the piano as a musical instrument is that it is easy to play as it **does not require a complex initial technique** as it can be with wind instruments. It also allows to sing and play several notes simultaneously, thus forming chords. As a result, the piano is one of the best instruments for learning music.

Both the learning of music and the instrument require many years of practice. When a student begins to learn how to play a musical instrument, it is necessary to complement his or her training with learning **how to read music**. This is because most musical methods or exercises are represented in the form of a music sheet. Normally, the student interprets these exercises individually and later a teacher corrects their mistakes. This methodology could cause that some students **lose motivation and leave their music studies** because if the student does not learn music theory he/she will not be able to properly acquire the instrument skills.

New technologies have opened a new door to the music. Whereas in the past it was necessary to attend a concert in order to listen to music, today, thanks to the advance of technology, we can enjoy music at any time. Technology has affected not only the way we listen to music but also the way we create and learn it. New applications have emerged that facilitate the composition of musical works, allowing the reproduction of compositions without the need to test them with a real orchestra. New sounds are emerging that give rise to new styles of music. And also, **new methods of learning music** appear through these technologies.

As mentioned before, classical methods of learning music consist of the repetition of exercises. Through technology, new approaches have been created to help the student in this task. In particular, many systems have been developed to **complement the role of the teacher** in learning music. An example of this is the tutorial videos about music, where a person explains how to play a song or some musical concept simulating a class. Another way for young people is to use web-based platforms to learn musical concepts through interactive games[1]. In addition, there are other more complex systems that collect the information from the musical instrument and evaluate whether it is playing the piece correctly, as a teacher would do.

What technology has allowed is to be able to represent music in a different way than traditional music sheets. In this way, a system can approach the learning in different ways depending on the target audience. In particular, one technology that will provide new ways of representing music is **Mixed Reality (MR)**. [MK94] The mixed reality permits to introduce virtual visual elements in our real environment and to be able to interact with them. This is possible thanks to devices similar to a helmet called a **Head Mounted Display (HMD)** [CFAW13] which through our sight and hearing achieve the desired immersion. Currently, there are few devices on the market that use this technology as it is still under development. One of the most advanced devices which uses this technology is the Microsoft HoloLens **Microsoft HoloLens**.

Microsoft HoloLens are glasses that allow to visualise and interact with mixed reality applications [Vro17]. How do they work? In order to visualise virtual objects, the HoloLens project the virtual images onto the retina taking into account the position of the retina and the movement of the user. As a result they create the illusion of visualising holograms. In addition to this, they are considered a *wearable device*, which means that no external computer is required to process the information - the device performs all the tasks itself. Thanks to their sensors, the HoloLens are able to analyse the space around them. As a result, it allows to create applications that interact with the floor, walls or tables. It also has a user interface with which through hand gestures or voice commands it is possible to interact with the system in an intuitive way.

[1] www.aprendomusica.com

According to a study published by the company *Perkins Coie LLP*[2] about the growth of mixed reality technologies, the second sector with the most money invested is in education, being video-games the first one. However, these sectors are sometimes linked, as in the case of serious or educational games, whose main objective is teaching. Since 1980, scientists and educators have been studying the benefits that video games provide to education [Mal80]. They concluded that a good way to capture the attention of students was to combine the use of new technologies with challenging and fantasy goals. In a research study that compared the use of games with traditional methods in students between 16 and 17 years of age in a computer application designed for learning [Pap08], it was concluded that the methods that used techniques similar to a game had better results. In addition, students who used the game were significantly more motivated and involved in learning than those who did not use the game. As a result, this places educational games as a good learning environment.

However, in order to make an educational game effective, it has to be well thought out. Otherwise, it could be boring or not achieve its main objective, which is to teach. The use of **gamification techniques** can help to solve this problem. Nevertheless, in order to make gaming techniques effective, the main system must have a solid base and a good idea to exploit. The simple fact of incorporating game mechanics does not make the system fun. The gamification is, according to the text, *Yu-Kai Chou*. [Cho15], the incorporation of fun and engaging elements typically found in games and applying them to other sectors. With the use of specific gamification techniques it is possible to **transmit positive feelings to the students while capturing their attention and interest**. The areas of gamification that achieve these objectives, according to the *Yu-Kai Chou's Octalysis gamification model*[3], are those responsible for stimulating creativity, giving feedback to the user, setting objectives and rewarding their actions.

Inspired by the idea of being able to represent music in a simple but instructive way thanks to the new possibilities offered by mixed reality, it is desired to soften the previously described problem. This problem was that students lost their initial motivation to learn music due to the complexity of the musical reading. It is also intended **to arouse interest in music** among new students. For this purpose, it has been decided to create a system that combines the learning of piano and music with mixed reality techniques, using the Microsoft HoloLens.

These techniques consist of using the **positioning of holograms to transmit relevant information** about a song in an intuitive way. Through specific mechanics, very important concepts of music can be represented, such as the **tempo and notes**. In this way, the student will become familiar with the piano and the basic musical concepts.

[2] `PerkinsCoie.com/AR/VR`
[3] `www.yukaichou.com`

To strengthen the learning of music in the system, gamification techniques have been employed. These techniques are those related to clearly defining goals, giving feedback and rewarding the user. The **objectives** have been represented in the form of levels. These levels contain a song and specific concepts for the student to learn. The levels are sorted by difficulty, in order to better guide the student. The **feedback** is represented directly on the piano. If the student has pressed the key correctly, he/she will receive positive feedback, and if he/she has made a mistake, he/she will receive negative feedback. The used **rewards** are stars that will appear at the end of the level.

In order to connect the piano to the HoloLens, a device called **Raspberry Pi 3**[4] has been used. This device is a low cost small computer with a single board. Through this device the information from the piano keyboard can reach the HoloLens. A USB MIDI converter has also been used to receive piano data on the Raspberry Pi. In the figure 1.1 a diagram representing the system architecture at a high level can be observed.

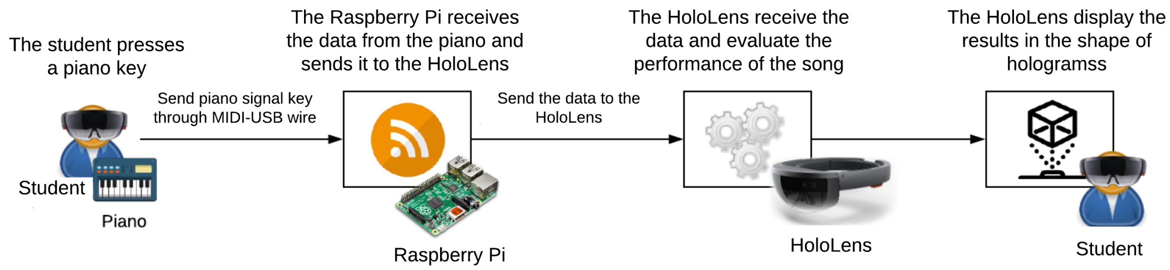


Figure 1.1: Overview diagram of the high-level system architecture

[4]www.raspberrypi.org

Chapter 2

Introducción

LA música ha acompañado al ser humano desde los inicios, siendo una de las primeras formas de expresión artística. Desde entonces, la música no se ha separado de nuestra historia. Según el filósofo *Nietzsche*, *"Sin música, la vida sería un error"*. Como arte, la música ha evolucionado en forma y estilo a lo largo del tiempo. También han evolucionado los medios por los que se transmite la música. El instrumento musical que siempre ha acompañado a la música es la voz. A esta se le fueron uniendo instrumentos de percusión y más tarde instrumentos melódicos.

Uno de los instrumentos musicales de mayor popularidad hoy en día es el piano. El primer piano tiene origen en el año 1700 aproximadamente, en Italia. Aunque es relativamente moderno, tuvo muchos antecesores que usaban la misma mecánica basada en emplear teclas para hacer sonar las cuerdas. Gracias a los grandes compositores del siglo XVIII como *Mozart*, el piano empezó a ganar importancia como instrumento solista y su popularidad creció. Una de las ventajas del **piano** como instrumento musical es que es **fácil de tocar** ya que no necesita de una técnica inicial compleja como puede ser con los instrumentos de viento. Además, permite **cantar y tocar varias notas simultáneamente** formando así acordes. Como resultado, el piano es uno de los mejores instrumentos para aprender música.

Tanto el aprendizaje de la música como el del instrumento, requieren muchos años de práctica. Cuando un alumno empieza a aprender a tocar un instrumento musical es necesario complementar su formación con el **aprendizaje de la lectura musical**. Esto es debido a que la mayoría de métodos o ejercicios musicales están representados en forma de partitura. Normalmente, el alumno interpreta de forma individual estos ejercicios y más tarde un profesor corrige sus fallos. Esta metodología podría causar que algunos alumnos **pierdan la motivación y abandonen sus estudios de música** ya que si el alumno no aprende solfeo no podrá desarrollar sus habilidades con el instrumento correctamente.

Las nuevas tecnologías han abierto una nueva puerta a la música. Mientras que antiguamente era necesario asistir a un concierto para escuchar música, hoy en día gracias al avance de las tecnologías podemos disfrutar de la música en cualquier momento. La tecnología no solo ha afectado a la manera de escuchar la música sino también como crearla y aprenderla. Han surgido aplicaciones que facilitan la composición de obras, permitiendo reproducir las partituras sin necesidad de probarlas en la orquesta con músicos reales. Surgen nuevos sonidos que dan lugar a nuevos estilos de música. Y también, surgen **nuevos métodos de aprendizaje de la música** a través de estas tecnologías.

Como se ha mencionado anteriormente, los métodos clásicos de aprendizaje de música consisten en la repetición de ejercicios. Mediante la tecnología, se han intentado crear nuevos enfoques que ayuden al estudiante en esta tarea. Concretamente se han desarrollado muchos sistemas orientados a **complementar la función del profesor en el aprendizaje de la música**. Un ejemplo de esto son los vídeos explicativos sobre música, donde una persona explica cómo tocar una canción o algún concepto musical simulando una clase. Otra forma orientada a los más jóvenes son las plataformas web para aprender conceptos musicales a través de juegos interactivos[1]. También, contamos con otros sistemas más complejos que recogen la información del instrumento musical y evalúan si está ejecutando correctamente la pieza, como haría un profesor.

Lo que ha permitido la tecnología es **poder representar la música de manera distinta a la partitura tradicional**. De esta manera, un sistema se puede enfocar el aprendizaje de distintas formas dependiendo del público al que se quiera centrar. En concreto, una tecnología que permitirá nuevas formas de representar la música es la **MR**. [MK94] La realidad mixta permite introducir elementos visuales virtuales en nuestro entorno real y poder interactuar con ellos. Esto es posible gracias a unos dispositivos similares a unos cascos llamados **HMD** [CFAW13] los cuales a través de nuestra vista y oído consiguen la inmersión deseada. Actualmente, existen pocos dispositivos en el mercado que usen esta tecnología ya que aun esta en desarrollo. Uno de los dispositivos más avanzados que usan esta tecnología son las **Microsoft HoloLens**.

Las Microsoft HoloLens son unas gafas que permiten visualizar e interactuar con aplicaciones de realidad mixta [Vro17]. ¿Cómo funcionan? Para visualizar los objetos virtuales, las HoloLens proyectan en la retina las imágenes virtuales teniendo en cuenta la posición de la retina y el movimiento del usuario. Como resultado crean la ilusión de visualizar hologramas. Además de esto, se consideran un *wearable device*, lo que significa que no se requiere de un ordenador externo que procese la información sino que el propio dispositivo realiza todas las tareas por sí mismo. Gracias a sus sensores, las HoloLens permiten analizar el espacio que les rodea. Como consecuencia, permite crear aplicaciones que interacciones con el suelo, paredes o mesas. También posee una interfaz de usuario con la que a través de gestos con la mano u órdenes de voz se puede interactuar con el sistema de manera intuitiva.

De acuerdo con un estudio de la empresa *Perkins Coie LLP*[2] sobre el crecimiento de las tecnologías de realidad mixta, el segundo sector donde más dinero se ha invertido es en la educación siendo los videojuegos el primero. Sin embargo, estos sectores a veces

[1] www.aprendomusica.com

[2] PerkinsCoie.com/AR/VR

están unidos, como es el caso de los juegos serios o educativos, cuyo objetivo principal es enseñar. Desde 1980, científicos y pedagogos han estado estudiando los beneficios que los videojuegos proporcionan a la educación [Mal80]. Estos, llegaron a la conclusión de que una buena manera de lograr captar la atención de los alumnos era combinar el uso de las nuevas tecnologías con objetivos desafiantes y de fantasía. De acuerdo con una investigación que comparaba el uso de juegos con los métodos tradicionales en estudiantes entre 16-17 años de edad en una aplicación informática diseñada para aprendizaje [Pap08], se llegó a la conclusión de que los métodos que utilizaban técnicas similares a un juego obtenían mejores resultados. Además, los estudiantes que usaron el juego, estaban significativamente más motivados y comprometidos a aprender que los que no lo hicieron. Como resultado, esto sitúa a los juegos educativos como un buen escenario para el aprendizaje.

A pesar de esto, para que un juego educativo sea efectivo tiene que estar bien planteado. De lo contrario, puede ser aburrido o no lograr su objetivo principal que es enseñar. El uso de **técnicas de gamificación** puede ayudar a resolver este problema. Sin embargo, para que las técnicas de gamificación resulten efectivas, el sistema principal debe tener una base sólida y una idea buena que explotar. El simple hecho de incorporar mecánicas de juego no consigue hacer un sistema divertido. La gamificación es, según *Yu-Kai Chou* [Cho15], la incorporación de elementos de diversión y de captación típicamente encontrados en juegos y aplicarlos a otros sectores. Con el uso de técnicas de gamificación especificas es posible **transmitir sensaciones positivas a los estudiantes mientras se capta su atención y su interés**. Las áreas de la gamificación que consiguen estos objetivos, según el modelo de gamificación *Octalysis de Yu-Kai Chou*[3], son las encargadas de estimular la creatividad, ofrecer retroalimentación al usuario, establecer objetivos y ofrecer recompensas por sus acciones.

Inspirándose en la idea de poder representar la música de forma sencilla pero instructiva gracias a las nuevas posibilidades que ofrece la realidad mixta, se desea suavizar el problema previamente descrito. Este problema consistía en que los alumnos perdían la motivación inicial en aprender música debido a la complejidad de la lectura musical. También se desea **despertar el interés sobre la música** en nuevos estudiantes. Para esto, se ha decido realizar un sistema que combine el aprendizaje del piano y la música con técnicas de realidad mixta, utilizando las Microsoft HoloLens.

Estas técnicas consisten en utilizar el **posicionamiento de los hologramas para transmitir información** relevante sobre una canción de forma intuitiva. A través de ciertas mecánicas se pueden representar conceptos muy importantes de la música, como **tempo y las notas**. De esta manera, se intenta conseguir que el estudiante se familiarice con el piano y con los conceptos básicos musicales.

[3] `www.yukaichou.com`

Para reforzar el aprendizaje de la música, se han utilizado técnicas de gamificación. Estas técnicas son las relacionadas con definir de forma clara objetivos, dar feedback y recompensar al usuario. Los **objetivos** han sido representados en forma de niveles. Estos niveles contienen una canción y unos conceptos específicos que el estudiante deberá aprender. Los niveles están ordenados por dificultad, para así poder guiar mejor al estudiante. El **feedback** es representado de manera directa en el piano. Si el estudiante ha pulsado la tecla correctamente recibirá feedback positivo, y si se ha equivocado recibirá feedback negativo. Las **recompensas** utilizadas son unas estrellas que aparecerán al final del nivel.

Para poder conectar el piano digital a las HoloLens se ha utilizado un dispositivo llamado **Raspberry Pi 3**[4]. Este dispositivo es un pequeño ordenador de una única placa y de bajo coste. A través de este dispositivo la información del teclado consigue llegar a la HoloLens. También se ha utilizado un convertidor MIDI USB que permite recibir los datos del piano en la Raspberry Pi. En la figura 2.1 se puede observar un esquema que representa la arquitectura del sistema a alto nivel.

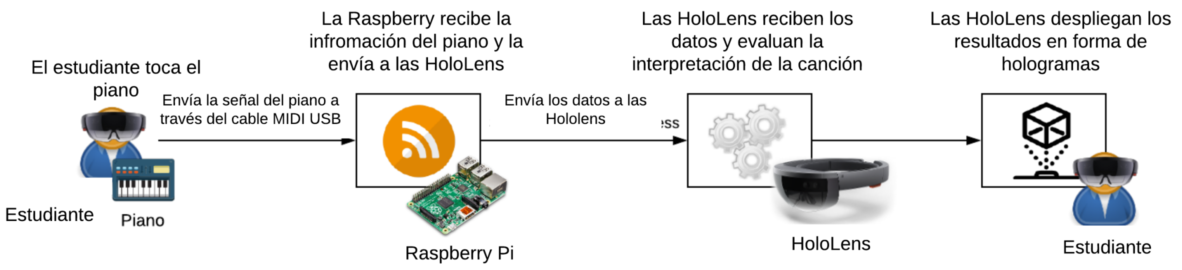


Figure 2.1: Esquema resumen de la arquitectura del sistema

[4]`www.raspberrypi.org`

Chapter 3

Objetives

In this chapter, the defined objectives to be achieved in this project will be explained. First, it will be introduced the general objective and then the specific objectives that complement the general objective.

3.1 General objective

The main objective of the project is to **facilitate the learning of music and piano** through the design, creation and development of a system based on **mixed reality** and **gamification** techniques for the **Microsoft Hololens** device. The use of the digital piano, will allow to evaluate the knowledge of music obtained and will be an additional incentive for the user. Through HoloLens and mixed reality techniques, new ways of teaching music could be created to arouse interest in learning music.

3.2 Specific objectives

For the development of this system, a number of specific objectives have been taken into account:

- **Intuitive learning.** In order to achieve the final system to be more accessible to students without previous knowledge of music, new methods of representing music notation combined with mixed reality techniques will be studied. As a result, it will simplify the way the music is read and played.
- **Immediate feedback.** With the aim of evaluating and correcting the user in the learning of music, information about how the piece is being performed at the moment will be provided. Through mixed reality, this information can be visualised in a more effective way. Feedback is very important because it is the medium through which the user can realise their mistakes and then learn.
- **Incremental learning** The system will follow a guided line of learning combined with the gamification techniques. In order to achieve this objective, the learning will be divided into levels. Each level will approach and evaluate a specific knowledge of music that will progressively increased in difficulty in the next levels. These levels guide the user and will contain small, specific learning goals.

- **Student's motivation.** With the aim of encouraging positive interest in the student who uses the system will apply gamification techniques that create challenges and objectives to achieve. These techniques are based on rewarding the user with prizes through the performance of these challenges. Specifically, at the end of each level, the users will be rewarded according to how well they perform.
- **User-friendly.** In order to have a pleasant and non frustrating user playing experience, it will be intended to create an intuitive and familiar user interface and to design the system with the purpose of facilitating the installation and configuration processes.
- **Scalability.** In order to increase easily the contents and take the most out of the system, the design and development of the project will be focused on making it scalable.

Chapter 4

State of the Art

IN this chapter it will be presented the main areas that constitute the background of this project.

As a principal point of reference, the concept of mixed reality will be explained. The reader will be guided through topics such as the evolution of the first systems that used technology based on mixed reality up to the present. It will be also provided some examples of the use of mixed reality in different industries as well as some of the main development tools of mixed reality.

As an important topic, the HoloLens device will be introduced, describing its characteristics and capabilities. From the point of view of the developer the tools for developing applications to the HoloLens will be presented.

Finally, it will be explained the most important and basic concepts about the musical representation system. It will also be discussed the piano, which is one of the main elements of this project. And finally, some of the systems that approach music education, how they do it and how they have been progressing are going to be described.

4.1 Mixed Reality

4.1.1 Introduction

The concept of **MR** was introduced by Paul Milgram and Fumio Kishino in 1994 [MK94]. In spite of being more than 20 years old, it is still not a popularly distinguished concept.

In the context of technology, MR includes the set of three dimensional taxonomies related with the grade of immersion of the observer in the world (environment where the application is displayed). These differ depending on many subjective and objective features like, for example, **reproduction fidelity** (how realistically the world is displayed), **extent of presence metaphor** (the extend of the illusion that the observer is present within that world), or the **extend of the world knowledge** (how much it is known about the world displayed) among others [MK94].

In the figure 4.1 it can be observed the representation of the Virtuality Continuum Scale (VC). It is a line which represents all the possible grades of immersion where the MR spec-

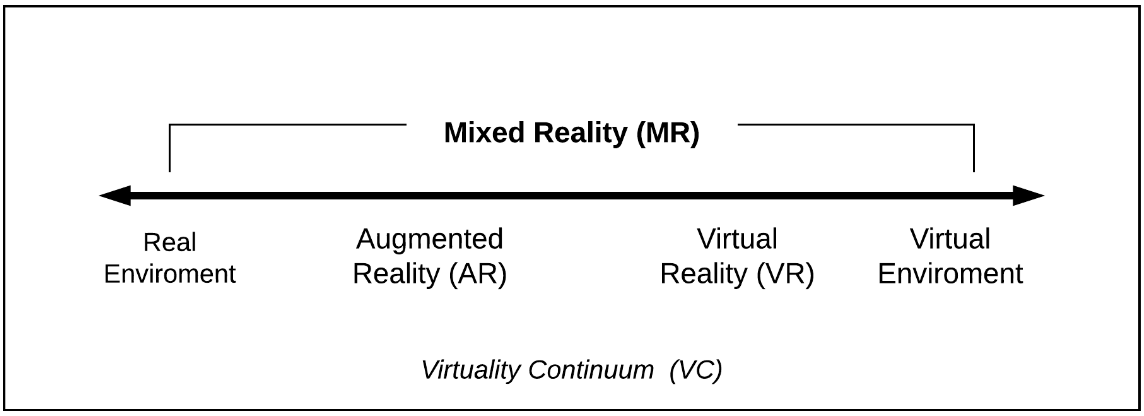


Figure 4.1: Virtuality Continuum Scale

trum is located. The left side represents the real world, without any influence of computers. In the meantime, on the other side, *virtual environment* represents a world simulated entirely by the computers. The space among both covers all the possibles combinations between real and virtual environments which belong to the set of MR.

Within the set of technologies that use MR (see figure 4.1) two concepts that are increasingly trendy can be found. Both are typically defined in terms of technological hardware. However, today, there are so many different types of systems that it would be a mistake to define them in relation to a specific hardware.

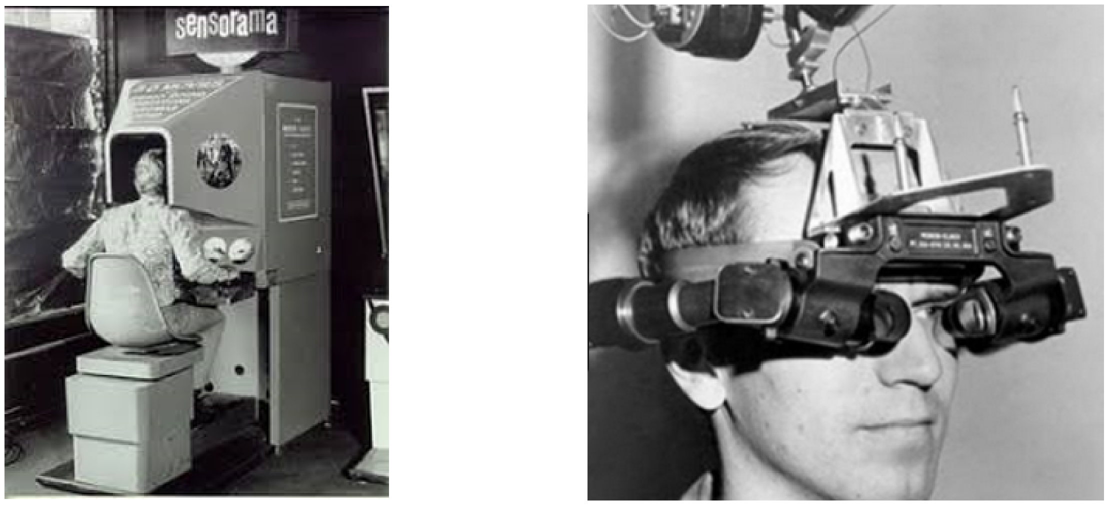
On the one hand, a Virtual Reality (VR) system would be defined as the interaction of an observer through a virtual world completely simulated by computers, whose objective is to achieve the most realistic possible immersion on it, without the intervention of any real-world visual element [Ste92].

On the other hand, Augmented Reality (AR) uses computer vision techniques to analyse the real environment [ABB^{+}01]. As a result, the system mixes the elements of reality with virtual elements generated by the computer. As can be appreciated, the main difference between AR and VR is the degree of interactivity with real-world elements, mainly visual.

Even though these concepts were defined a long time ago, there is still no standard that could objectively classify the mixed reality systems. Therefore, it could be said that a system can belong to both AR or VR sets depending on the point of view. In this situation, the best option is to classify it as a MR system, since it includes the previous ones.

4.1.2 History of Mixed Reality

The history of the first mixed reality systems dates back until 1960, before the release of the first Personal Computer (PC) (in 1981). Between 1960 and 1962, Morton Heilig created a multi-sensory simulator called *Sensorama* (see figure 4.2a). It was a colour and stereo film augmented by binaural sound, scent, wind and vibration experiences. It was the first



a) Sensorama b) The Sword of Democles VR system

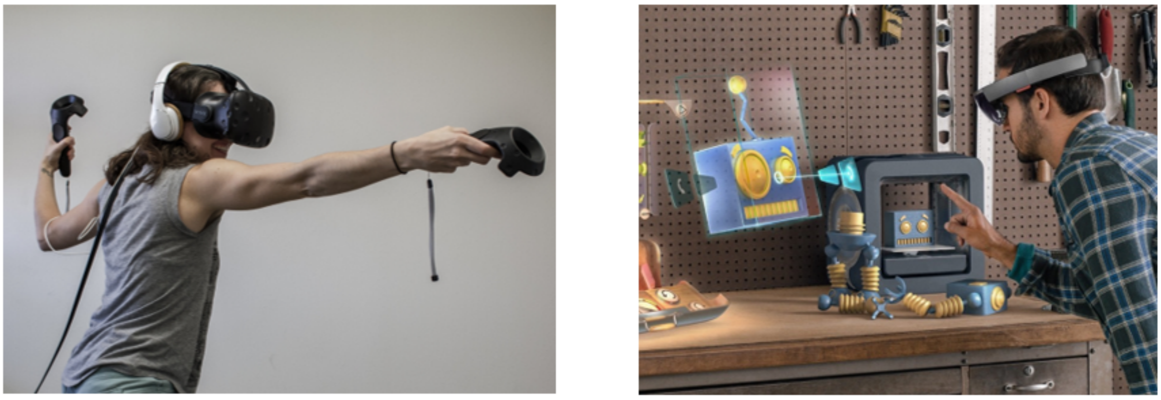
Figure 4.2: Some of the first mixed reality systems

approach to create a virtual reality system but it was not interactive. Later, in 1965, a system called *The Sword of Damocles* (see figure 4.2b) was born, which developed in hardware the concept of the first HMD. It supported a stereo view that was updated correctly according to the user's head position and orientation. In 1985 and 1988, the Virtual Programming Languages (VPL) company manufactures the popular *DataGlove* and the *EyephoneHMD* – the first commercially available VR devices [MG96].

At the same time but slowlier, AR began to improve thanks to the launch of the *Visually Coupled Airborne System Simulator (*VCASS*)*, an advanced flight simulator using a HMD. It was developed by Thomas Furness at the US Air Force's Armstrong Medical Research Laboratories in 1982. This was achieved through a see-through HMD that superimposed virtual three-dimensional objects on real ones. As a result of these projects as well as many others, from the 90's onwards, the research of computer vision algorithms and the production of mixed reality systems have grown considerably until the present day.

4.1.3 Display technologies

So, what is a Head Mounted Display (HMD)? A HMD is a display device similar to a helmet or glasses, which makes it possible to reproduce images created by computer on a screen very close to the eyes or projecting the image directly onto the retina of the eyes [MM01]. In the latter case, the device is called Optic Head Mounted Display (OHMD). HMDs are usually associated with virtual reality. They are able to fool the senses by emitting different images in both eyes giving a 3D sensation, achieving visual immersion of the virtual world (see figure 4.3a). Some models also allow to see the entire surrounding environment and introduce virtual objects (called holograms) or information into it, thus producing AR or MR (see figure 4.3b). Currently, there are also opaque HMD devices, like **Acer Windows**



a) HTCVive VR b) Microsoft HoloLens

Figure 4.3: Example of a VR HMD and a AR OHMD

Mixed Reality [1] ,that through front cameras allow the visualisation of both VR and AR, mixing the images of the real world with the virtual world and displaying them on the screens of the HMD.

The high impact of mixed reality technologies on the market has resulted in an expansion in the development of these technologies for different platforms and devices. Hence, companies such as Apple, Google, Microsoft, or Facebook, among others invest large amounts of money in research for mixed reality. In particular, Facebook leads the ranking as the company that has invested the most after buying Oculus [2] for $2 billion and investing at least $500 million more in 2017 [Mer17]. According to a recent estimate by Goldman Sachs [3] [HT17] , MR is expected to grow into a $95 billion market by 2025.

Thanks to the improvement of the hardware in the mobile devices, it is now possible to run AR or VR applications with only the components of the mobile phone itself. As explained above, HMDs were devices that depended exclusively on an external Central Processing Unit (CPU) to move the entire system. Today, HMDs have been adapted for mobile phones, and through magnifying lenses the user can see a virtual 3D world just using the display of the smartphone. In addition, by removing the large majority of the HMD electronic components, it has been possible to reduce their cost enormously. In 2014, Google launched its Google Cardboard. It is similar than the HMD described above (see figure 4.4). It is made of cardboard and were initially sold at a price of $20. Today they can be found for $5 or less. The result has been that VR has reached to many more people and its use has been extended to all types of users and industries.

Unlike virtual reality, which requires an additional complement, augmented reality has

[1] More models of opaque MR headsets www.microsoft.com/en-us/store/collections/vrandmixedrealityheadsets

[2] It is an American technology company specialises in virtual reality hardware and software products.

[3] One of the world's largest investment bank and financial services company.

Figure 4.4: Google Cardboard Glasses

all the components needed to work on today's smart-phones. The main part for this is the camera. Although, today, more advanced cameras are being developed which detect depth, through a simple camera and computer vision algorithms, it is possible to process the image and superimpose an object on the screen giving rise to AR. The oldest and most popular technology that allows this is known as Marker-based tracking [KB99]. Through some physical markers known by the system, it is able to map the position, scale and orientation of the marker. Thus it is able to place a virtual object on the marker in a more realistic way and in real time (see figure 4.5).

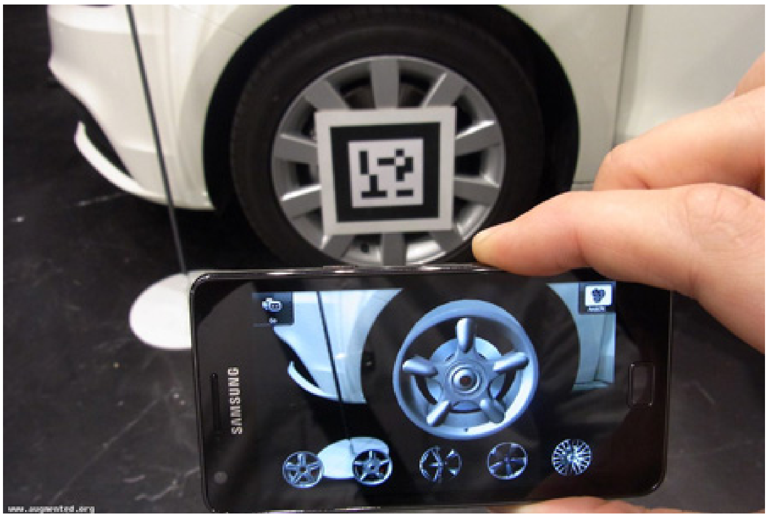


Figure 4.5: Augmented reality marker recognition system

A good marker is such that a computer vision system can robustly and reliably detect it and identify it unambiguously [VKP10]. Due to the factors of illumination and colour, a typical problem would be that, depending on the lighting, the colours of the marker could change for the camera and spoil the tracking. Hence, black and white markers are better suited. However, if the colours contrast with each other, using machine vision techniques, a result similar to black and white markers could be achieved.

Many types of markers can be found. The most common are binary markers. These are interpreted by giving a colour the value 1 and its opposite value 0, for example white, 1, and black, 0. In this way, information can be encoded in the markers and make them dynamic. There are two basic types of markers depending on the how represent the information: one-dimensional, or linear, and two-dimensional.

The first coded markers were the bar codes. These were read from left to right (1D). Their evolution was the stacked barcodes. Due to the need of representing more information, matrix codes (2D) emerged which allow to be read in horizontal and vertical [Inc11]. Thanks to the improvement of the cameras and processors, standardised marks have been developed that are becoming increasingly smaller and are able to encode more information, such as Quick Response (QR) code, DataMatrix or PDF417, among others. Today, both are being used, the traditional barcodes and the previous ones (see figure 4.6).

(a) BarCode-128

(b) DataMatrix

(c) PDF717

(d) QR Code

Figure 4.6: "hello world" message with different code markers standards

4.1.4 SDKs

As introduced above, hardware features are those which give developers the possibility of building mixed reality systems. Afterwards, Software Development Kit (SDK)s help them to adapt software applications to these architectures in order to achieve the desired world immersion. A SDK is a set of software tools that allow developers to build applications for certain systems. In the field of AR many different SDKs which have also improved over the time can be found. Some of them are discussed below [AG15].

- **OpenCV.** Probably the most famous SDK, and the core of other higher level augmented reality libraries, is Open Source Computer Vision (OPENCV). It began to be developed in 1999 by the company Intel. However, it was not until 2006 when the version 1.0 was released. One fact that made it so much improved and important is that it is cross-platform and open source library under the Berkeley Software Distribution (BSD) license. OPENCV was designed for computational efficiency and with a strong focus on realtime applications. It is written in optimised C and can take advantage of multicore processors [BK08]. Its functions cover most fields of computer vision and have been increasing in every new version of the SDK. It covers areas like image manipulation, facial recognition, factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, robotics and many more [4].

- **ARToolKit[5].** It was originally developed by *Hirokazu Kato* of Nara Institute of Science and Technology in 1999. It is also an open source free library whose main function is to recognise markers and project 3D objects onto them. The markers it recognises must be square and in black and white. Its advantage is that it facilitates the registration of markers and objects for the programmer. It is supported by many operating systems such as Microsoft Windows, Mac OS X, Linux, iOS, and Android.
- **Wikitude[6].** It was launched in 2008 by *Mobilizy GmbH*. Wikitude includes image recognition and tracking. It supports 3D model rendering with overlaying of video and provides location based AR. One of its advantages is that it uses a hybrid image recognition system, combining the user's geographic position by Global Positioning System (GPS) or Wi-Fi with the data from the sensors such as the accelerometer. It is also cross-platform because it is developed for web technologies (through HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript) which allows it to be used for most devices. Its main commercial target are tourism applications.
- **Vuforia[7].** It was originally launched in 2011. Vuforia is a SDK that provides advanced image recognition techniques at a high level, which means it is easy to implement. It supports both native development for iOS and Android. It could be cross-platform due to is its supported in Unity, that provides easily portable solutions to more platforms like Windows. What is remarkable about Vuforia is the simplicity of adding new markers. Through its web portal, Vuforia analyses 2D images or 3D models to be detected. This creates some key points for its later recognition algorithm over the image or model. Once this process is done, it is possible to obtain the markers from the cloud through the same application or insert them manually as a local database. It provides very efficient tracking of up to 5 markers simultaneously. The disadvantage is that it does not incorporate standardised mark detection such as QRCode.

[4] Here it can be found all Opencv fuctions `www.docs.opencv.org/master/`.
[5] `www.hitl.washington.edu/artoolkit`
[6] `www.wikitude.com`
[7] `www.vuforia.com`

a) Wikitude

b) Junaio

Figure 4.7: Examples of MR Guide-Based Applications

4.2 Applications based on Mixed Reality

As mentioned above, the expansion of mixed reality technologies in the market has resulted in a great immersion in all types of industries. In this section we will present some of the most characteristic applications of mixed reality in each industry and the impact that the it have had on them.

4.2.1 Tourism

In the field of tourism, the mixed reality has had a strong acceptance and many companies have decided to incorporate mixed quality systems to improve the experience of tourists [8].

Tourism-oriented applications can be classified into two categories. These are related to two different problems of tourism. One is the location of the landmarks, which is directly linked to the action of guiding the tourist through their destination in the most efficient way so that they can see and enjoy as much as possible. And the other is the reconstruction of a monument based on its ruins so that they can visualise the past and feel more immersed in the history [Oli14].

- **Guide-Based Applications.** The main targets of these applications are mobile devices such as tablets, smartphones or wearable devices. They mainly make use of the camera and the GPS coordinates of the device to orient themselves on the map. Among the applications oriented to this market, Wikitude, which uses its own SDK aforementioned, can be found. This is one of the pioneering applications in its field. *Wikitude* is based on showing on the mobile camera view relevant information for the user such as hotels, monuments, etc. (see figure 4.7a). Based on this guide concept we can also find other applications such as: *Toozla*, *ARO*, *Yelp*, *GUIMO*, *Junaio* (see figure 4.7b), or *Layar* among others.

[8] www.targetmarketingmag.com/article/tourism-future-augmented-reality

a) Archeoguide b) HoloTour

Figure 4.8: Examples of mixed reality tourism-based applications

- **Historic Heritage-Based Applications.** The main objective of these applications is the reconstruction, animation or virtual visualisation of monuments and emblematic buildings. The scope of these applications is more extensive. It can be found from applications that introduce us directly into the ancient world through HMD like *HoloTour* (see figure 4.8b), others that use augmented reality to generate the 3D model on the real ruins to visualise how it was before and how it is now as *Archeoguide* (see figure 4.8a) or for example *Google Goggles* which gives a description of a monument that the camera analyses.

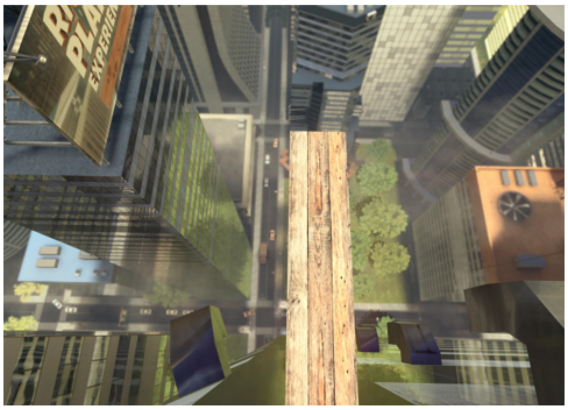
4.2.2 Simulation

Since the development of the first simulator in 1982 and the multi-sensory simulator, Sensorama in 1960, it could be said that the pioneering sector is simulation. The very fact of creating a virtual world and achieving a complete immersion of the user implies a simulation and that is why this field was the first to include mixed reality. The area that most includes the simulation is virtual reality since it is possible to have a completely controlled and safe environment.

A fields where simulation is very effective is in the treatment of psychological phobias. Through a virtual reality HMD, the patient is confronted with situations of fear. The patient is progressively overcoming more and more dangerous situations. In this way, it has been possible to treat phobias such as the fear of flying, through a flight simulation, the fear of heights, walking through higher and higher cornices like in Richie's Plank Experience system (see figure 4.9b), claustrophobia, emulating a closed room, the fear of driving, by driving a virtual car, or the fear of spiders, among others. [AGPB07].

Another of the uses of simulation is to train in some discipline before its realisation. This is widely used in the military field. There is a submarine simulator that recreates through a real periscope the experience of piloting the submarine. This is called *SIPER* and was developed

a) View through the periscope of the SIPER submarine simulator



b) Richie's Plank Experience game

Figure 4.9: Examples of mixed reality simulation-based applications

in 2002 by the Argentine Navy (see figure 4.9a). The aim of these systems is to significantly reduce the costs of normal training through a simulation of real machines. It is also possible to find radar simulators such as *ATLAS 2*, which is capable of recreating the behaviour of a real radar [YASV11]. Another kind of systems also popular are the pilot simulation systems such as the *Saitek* system which consists of a real cabin with a set of complex controllers of a real plane that are able to simulate a flight.

4.2.3 Video-games

Video-games are the most profitable industry in the interactive entertainment market. "The 2017 Year In Review Report by *SuperData* shows that mobile games remain the biggest sector, generating $59.2bn, followed by PC at $33bn and consoles at $8.3bn" [9]. As a result, this turns video-games industry into the biggest investor and promoter of mixed reality.

The advantage that video-games have in mixed reality is that they can exploit this technology to the maximum because it only has the limitations of the developers' imagination and the hardware capabilities of the device being used. However, this technology is not yet achieving the expected benefits. This could happen due to two important factors. One, the high cost of mixed reality devices, and two, the power limitations in relation to current video games that do not use mixed reality.

In the gaming industry the line that separates augmented reality from virtual reality is greatest. This is why people usually confuse the terms of mixed reality with those of previous ones.

- **Virtual Reality.** Virtual reality video-games are the evolution of VR simulation

[9] James Batchelor's article from *gamesindustry* digital magazine www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017

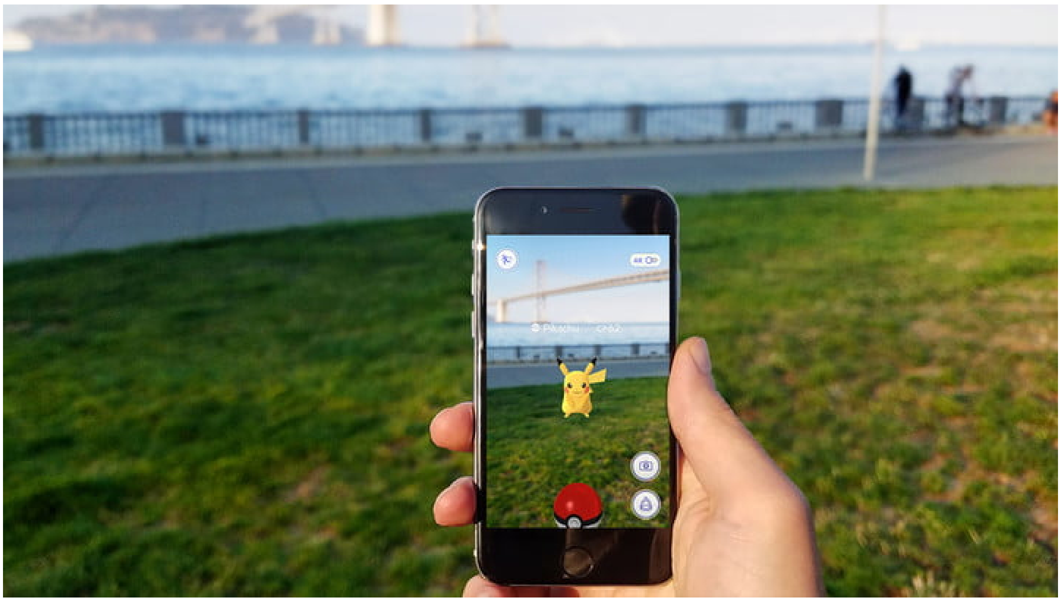


Figure 4.10: Pokemon GO

systems into a more commercial and entertaining field. Virtual reality video-games are the evolution of simulation systems into a more commercial and entertaining field. Nowadays, it has already been possible to adapt virtual reality to any genre of video-games looking for an even bigger immersion than through a screen. These games depend on powerful computers to be played because they require a lot of resources to run. Also, games already created are being adapted to virtual reality such as *Skyrim VR* [10] or *Fallout 4 VR* from *Bethesda Game Studios*.

- **Augmented Reality.** On the other hand, augmented reality is more typical of mobile devices such as smartphones. These games open up new mechanics to use. One of the most commonly used examples today is *Pokemon GO* (see figure 4.10). It was launched by the company *Niantic Inc.* in July 2016 for Android and iOS devices. In August 2017, it reached a record of 100 million downloads only on Google play store with a total profit of $268 million [11]. Although it was not the first to do so, this game allows the player to interact with street elements, such as points of interest located throughout the city, obliging the player to take to the streets and explore their city. It also promotes exercise as it is the distance travelled which influences when it comes to catching Pokemon. Or sociability as well, as players have to join together to have a better chance of defeating a Pokemon gym.

In conclusion, the mixed reality has already shown that it will have a big impact in the field of video-games and will allow the creation of new ways of playing and entertainment.

[10] www.wowamazing.com/trending/new-tech/can-facebook-really-do-teleportation-see-what-the-future-holds-for-2025

[11] www.businessofapps.com/data/pokemon-go-statistics/

4.2.4 Marketing

The application of mixed reality in the world of marketing can be very powerful and will change the way companies advertise their products. This change would allow us to take the leap to be able to observe the object in front of us to see how it would be physically and not see it through a TV or magazine. Or for example, to introduce yourself in the advertisement and be the protagonist of a short adventure. All these approaches would attract more the consumer's attention and would improve the marketing industry.

Augmented reality is already being used in advertising through the use of QR markers or similar. These markers offer the reader of the advertisement the possibility to access to more information by scanning them with their mobile devices [RDPS12]. It could be said that the marker is a door to the Internet where such advertising can come to life. Some examples of using the QR code in advertising would be: downloading an application, subscribing to a newsletter, entering a raffle (see figure 4.11b), a link to the product purchase page, additional information about the advertisement and many others.

Some companies are already developing their own applications that allow the customer to visualise the product through augmented reality as if it were in front of him. In September 2017 Ikea launched its *Ikea Place* application. This allows to select through a digital catalogue the furniture that can be bought in the current store. Once it is selected, the system displays the 3D object of the furniture in the room (see figure 4.11a). Through the camera the client can place it in a different place, move closer to it, check its measurements, and try different combinations to see how it would look in the home.

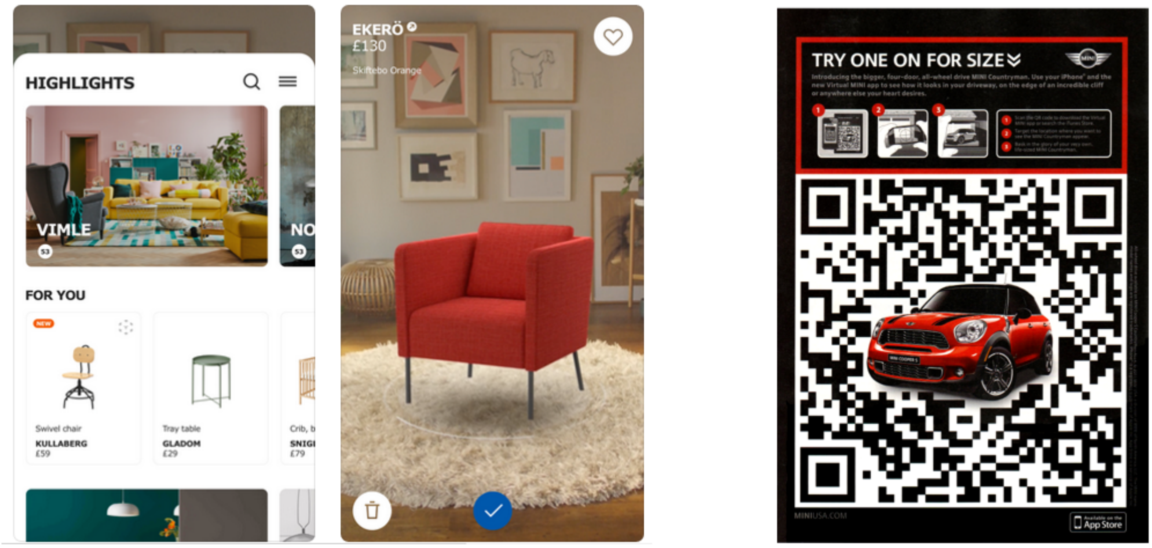
In conclusion, mixed reality in the advertising industry also has a lot of potential and will strongly influence the sales of some products.

4.2.5 Industrial Manufacturing

Computers are becoming more and more important in manufacturing. They monitor the production processes and allow us to create more and better products. This is why there is a need for these computers to be able to control production, identifying each product in a unique way. Through augmented reality, this problem can be solved using barcodes.

Barcodes is one of the technologies that has made this industry more advanced. Most companies use this technology or a similar one. These provide unique codes for products. This code is usually a key to identifying a set of important properties of this product such as when it was factored in or what it is composed of. The codes are used by the production machines themselves, by the supermarket employees or by the customers themselves who want to know more about their properties.

Another interesting use is to assist employees in case they do not know how to use a machine. Through the use of AR glasses, the employee visualises the machine and the system



a) Ikea Place b) Mini QR code advertisement

Figure 4.11: Examples of mixed reality marking-based applications

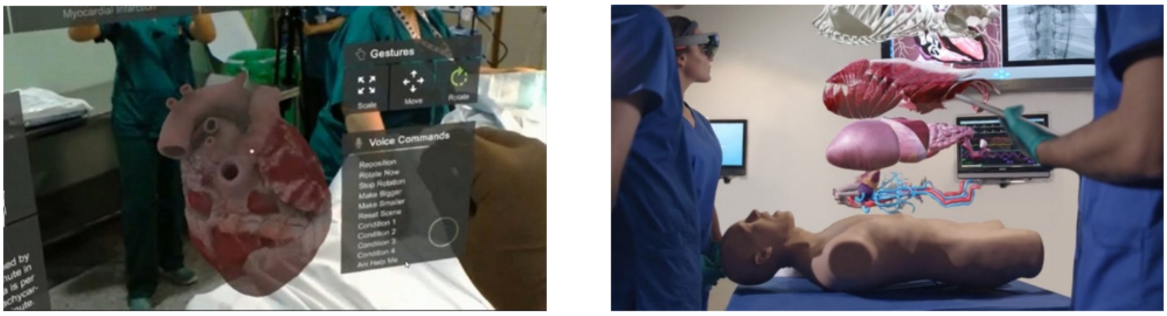
displays an interface where it guides the employee in the process he wants to achieve.

Finally, another application that augmented reality has in this field is making mock-ups or showing them to customers. When companies have to make a very expensive product it is important to make previous models in case the customer is not satisfied with the final product. Instead of printing or producing the model, the client would visualise the 3D model of the product in the form of a hologram with the advantage of having a very intuitive interface (see figure 4.12).

Figure 4.12: Example of mixed reality manufacturing-based applications

4.2.6 Medicine

In medicine, the application of mixed reality is completely different from other sectors. This is mainly due to the fact that the data used by the applications normally depend on very



a) HoloSurg b) VimedixAR

Figure 4.13: Examples of mixed reality medicine-based applications

complex systems specific of the medical field.

One of the problems that delayed the entry of mixed reality into hospitals was the complexity of the systems. These were very spacious and difficult to use. It was an inconvenience when it comes to be incorporated in the hospitals. In addition, it required previous formation. However, thanks to the improvements and standardisation of the mixed reality devices, they have managed to take their first steps in medicine. In Spain, October 2017, the first operation was performed by using mixed reality glasses (HoloLens) through a system called *HoloSurg* [12]. The doctor used the augmented reality to see the patient's heart in real time and take action based on it (see figure 4.13a).

Another application of mixed reality in medicine is to train doctors visualising parts of the human anatomy. The recent systems that put it into practice this concept are *HoloIVI*, *MediSIM* and *VimedixAR* (see figure 4.13b) among many others.

As mentioned in the simulation section, there are also applications used to treat psychological problems (see figure 4.9b).

Finally, the field where mixed reality is also introduced is in rehabilitation [GT94]. This usually consists of a camera that tracks the patient's movements, providing direct feedback on whether he or she is exercising well or not. In addition, it also collects the results obtained and carries out a study of the progress made. It also allows the use of gamification techniques and thus benefit from their advantages.

4.2.7 Education

Probably one of the most effective applications of mixed reality is in education. This could be because this technology arouses the interest of students and attracts them to experiment with it.

One of the first projects oriented of augmented reality to education was the *MagicBook*

[12] `www.xataka.com/medicina-y-salud/la-cirugia-con-hololens-no-solo-es-real-es-sorprendentemente-util`

a) Magic Book b) Galaxy Explorer

Figure 4.14: Examples of mixed reality education-based applications

project (see figure 4.14a). It is an educational book with a number of markers attached instead of drawings. Through a display device, the student can analyse the makers meanwhile reading the description of the figure and the system will superimpose the 3d model associated with it. In this way the student's interest in the subject is increased and learning becomes more dynamic. This system was used in subjects such as volcanoes or solar systems [BOE+07].

Nowadays, thanks to the evolution of technology, more advanced didactic systems have been developed where most of the information needed by the student is given by the system, either by text, in the form of explanatory audio, or by interacting with the system itself. As a result it can be found, for example, Microsoft's *Galaxy Explorer* application (see figure 4.14b), which is a evolved system from the previous project. It starts by placing the Milky Way in the room. a voice is in charge of explaining each planet and its properties. Meanwhile, the student can move through the galaxy just moving through the room and interact with the planets.

What makes mixed reality so powerful for education is the infinite number of applications that can be applied to it. Each subject contains so many concepts to explain through mixed reality. For example, in History, travelling to a virtual world of the past that recreates the historical concepts to be learned. In Biology, visualising any type of molecular. In Physics, interacting with different laws such the gravitational law. In conclusion, mixed reality in the field of education already have large resources and great potential to utilise.

4.3 Microsoft HoloLens

The Microsoft HoloLens is an OHMD that is also a wearable computer. It was launched on March 30, 2016 by Microsoft with the Windows 10 Operative System (OS) [Vro17] (see figure 4.15). Contrary to other wearable devices such as smartphones or tables, the HoloLens interact with the environment, read, analyse and use it constantly for its workings. This is thanks to a set of sensors that the device carries in the front composed of several cameras, an

Figure 4.15: Microsoft HoloLens

Inertial Measurement Unit (IMU) and microphones among others.

The process of analysing the surroundings is called **spatial mapping** and will be discussed later.

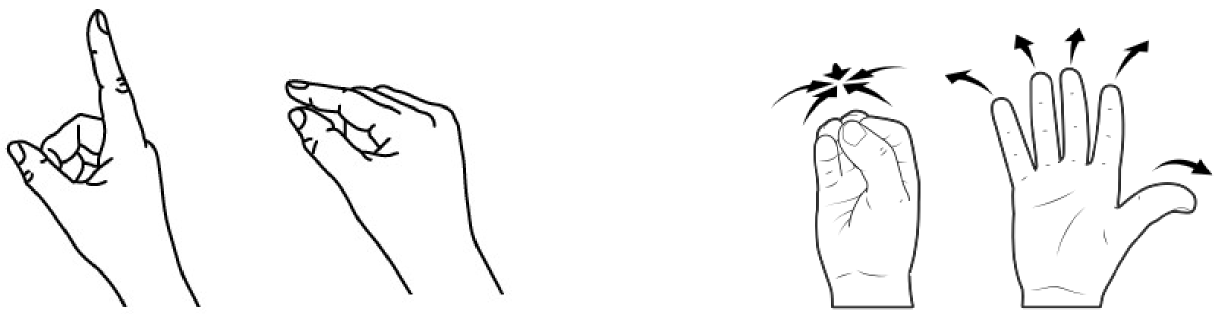
The HoloLens weighs 579 grams. It has a Intel 32 bit architecture with Trusted Platform Module (TPM) 2.0 support and a Custom-built Microsoft Holographic Processing Unit (HPU) 1.0 [13]. Its battery can last between between 2 and 3 hours. It has the communication and data input/output interfaces that the current modern devices have. The most important are Wi-Fi 802.11ac, Micro USB 2.0 and Bluetooth 4.1 LE.

The HoloLens use a new technology called **Holographic Lenses**, which allows to visualise the projected holograms in the real world from the user's eyes, achieving a sensation of realism. And finally, the **Human Understanding** system that allows to interact with the device through voice or gestures without the need of any physical controller composed of subsystems called Spatial sound, Gaze tracking, Gesture input and Voice support [14].

- **Spatial sound** allows the user to hear binaural audio which can simulate spatial effects, meaning the user, virtually can perceive, and locate a sound, as though it is coming from a virtual pinpoint or location.
- **Gaze tracking** allows the user to bring application focus to whatever the user is perceiving to navigate and explore, the technology can tell exactly what and where to show the images for each pupil to generate stereoscopic 3D illusions. It is similar to the pointer on a computer. It is handled by moving the head.
- **Gesture input** allows the use of hand gestures. These gestures permit the user to interact with the system. Currently, there are two types of gestures the *Air tap* and the

[13] docs.microsoft.com/en-gb/windows/mixed-reality/hololens-hardware-details

[14] www.abcomrents.com/wp-content/uploads/2018/01/HoloLensSpecSheet.pdf



(a) Air tap (b) Bloom

Figure 4.16: Gestures of the Hololens

Bloom [15].

- *Air tap.* is a tapping gesture with the hand held upright, similar to a mouse click or select (see figure 4.16a).
- *Bloom.* It is a special system action that is used to go back to the Start Menu. It is equivalent to pressing the Windows key. The user can use either hand (see figure 4.16b).

There are other more complex gestures that result from the air tap gesture, they are the *Tap and hold*, *Manipulation* and *Navigation*.

- *Tap and hold.* Hold is simply maintaining the downward finger position of the air tap. This allows actions such as "click and drag" to be used. By moving the arm it is possible to manipulate the previously selected action.
- *Manipulation.* Manipulation gestures can be used to move, resize or rotate a hologram. This action can be realised by clicking on the desired action, pointing with the gaze and *tap and hold* until the desired shape is achieved.
- *Navigation.* Navigation gestures operate like a virtual joystick, and can be used to navigate User Interface (UI) widgets, such as radial menus. It consists of tapping on the surface and moving the hand in the desired direction to slide through the interface.

• **Voice support** allows the user to use voice commands (similar to asking Cortana, Siri, Google a question). It also allows developers to use the Text to Speech capability to create voice inputs for their applications.

[15] www.docs.microsoft.com/en-us/windows/mixed-reality/gestures

4.3.1 Setting up HoloLens

The first time the HoloLens are started it is necessary to carry out a series of instructions to configure them [Vro17]. They will be explained in order

1. *Calibration.* It is the most important step. This application consists of aligning the projection distance of each screen to the pupils. The number that represents the separation distance is called **Interpupillary Distance (IPD)**. It consists of aligning the finger with a virtual template, placing it over it. This process should be done twice, once for each eye, closing the eye that is not being calibrated. The IPD is unique to each device and is not visible. if the device is not calibrated properly the holograms will not look good and may cause dizziness and future vision problems. It is not recommended to use the device on children under 13 years of age as their IPD is too low and can cause health problems.

2. *Learn gestures.* In this phase the use of the input gesture system is learnt. first the air tap is introduced, and then the boom. Afterwards, a hologram is deployed on which the controls for moving, scaling and removing a hologram are shown.

3. *Setting.* Finally, a series of parameters have to be configured, such as e-mail, Wi-Fi connection and language.

Once these steps are completed the main Graphical User Interface (GUI) is displayed. From the main menu it is possible to access all the functions of the HoloLens. It has a status bar at the top that indicates information about the Wi-Fi, the current time, the battery status and the current volume level. In the middle there are the favourite applications represented in square form including a Cortana. All applications can also be accessed via the '+' button. Finally, at the bottom there are two buttons for taking pictures or videos (see figure 4.17).

To access the desired section, the user can use the gaze and the air tap to select and open, or use the curtain and tell the user the action to be performed, for example "hey Cortana!, take a photo". To display or remove the menu, the bloom gesture has to be used. if the user is in an application, when doing the bloom, it will be paused and the system will return to the main menu.

4.3.2 Spatial mapping

The technology that makes the difference against other devices is the spatial mapping. Through the cameras incorporated in the device, it is able to take three-dimensional data from the environment creating meshes that cover the surrounding objects [Vro17] (see figure 4.18). The HoloLens process these meshes and generate a space with physical properties like the real world. The spatial mapping can be accessed from any application providing a more realistic inversion experience. This can result in situations such as follows: placing holograms on walls or tables, making holes in surfaces, recognising surfaces (this is

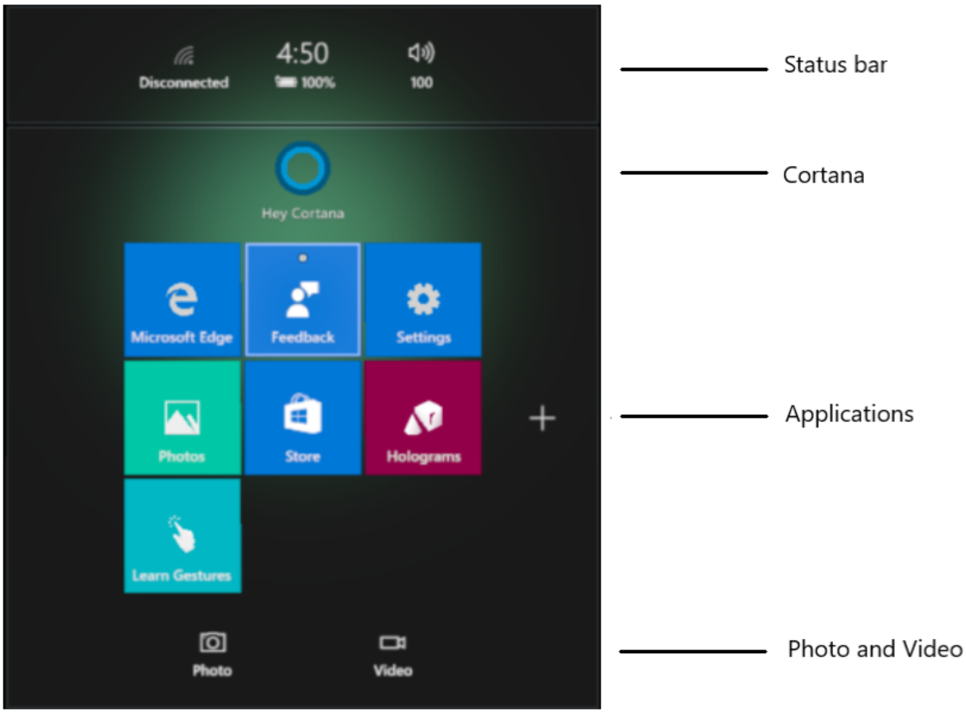


Figure 4.17: HoloLens main menu

called **mapping understanding**) or hiding holograms behind real objects (this is called **occlusion**).

How does spatial mapping works?[16] The depth cameras at the edges of the HoloLens detect a number of three-dimensional points that contain the x,y,z coordinates in the space. These points are called vertices. Once detected, a complex algorithm is responsible for generating triangles by linking related points by proximity. This algorithm cleans the noise and generates the meshes resulting from the union of these triangles. This process can be launched as a maximum at a speed of 0.2 seconds without affecting performance much, without taking into account the visualisation of the meshes, since it would increase the time.

Spatial mapping can be confused with object recognition. The spatial mapping has the objective of making approximate contours of the objects that surround the user, seeking efficiency and rapidity in order to detect changes in the environment quickly. this technology does not allow to map an object accurately and therefore it is not possible to identify it. One approach to this is spatial understanding but it is a long way from detecting small objects.

An important property of spatial mapping is occlusion. It allows to hide the part of the hologram which is behind a mesh from the view of the camera. as a result creates even more immersion.

[16] `talesfromtherift.com/hololens-contest-week-10`

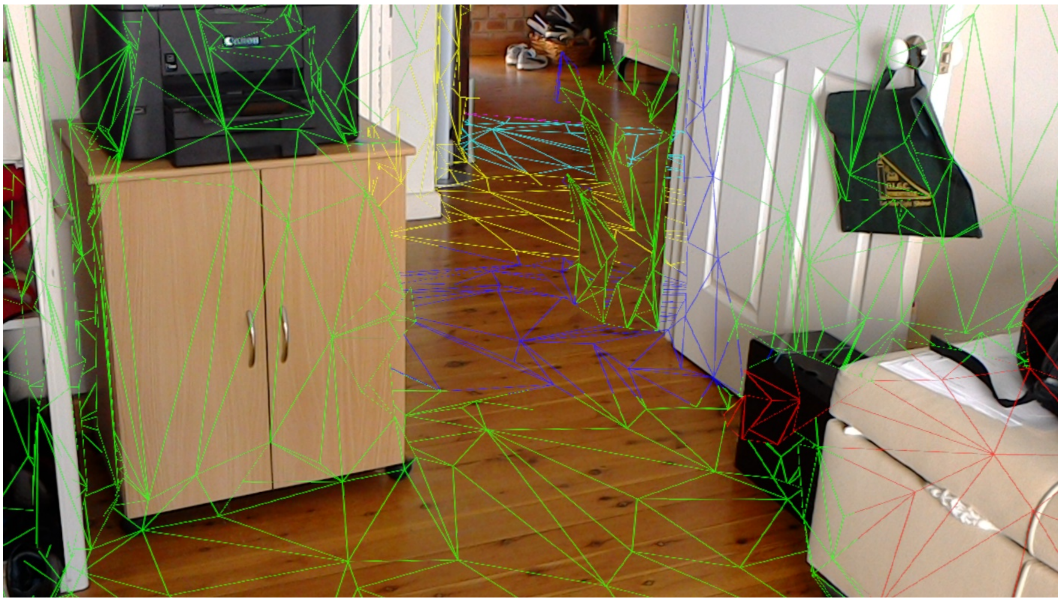


Figure 4.18: Spatial mapping

4.3.3 The device portal

Each HoloLens has a built-in web server that serves up pages that inform about the device and the state it is in. To access the device portal it is necessary to be in the same network as the HoloLens or to be connected to them via Universal Serial Bus (USB). When the Internet Protocol (IP) of the HoloLens is inserted in the browser, the device portal is displayed. Some of the features of the device portal are described below [Vro17].

- *Home.* It is the main menu. Here, it is possible to access to the basic information about the HoloLens, the battery, the Wi-Fi, the name, the windows version, etc. It is also possible to perform actions on the HoloLens, such as turning off, suspending or changing the battery settings.
- *3D View.* In this menu it is possible to visualise the mapping that the HoloLens is performing of the environment. Once it is mapped it is possible to download the mapped room file in *.xef* format.
- *Mixed Reality Capture.* In this view the image capture settings through the HoloLens can be accessed. It is possible to take photos, videos, and live streaming. The streaming has around 5 seconds of delay. It is also possible to capture only the holograms, the video, or the video merged with the holograms, just as it is seen through the holograms.
- *Processes.* Through this page all the processes that the HoloLens are performing, both operating system and other applications can be visualised and closed in case of any problem.
- *System Performance.* This area is used to access to the graphs of the HoloLens' actual performance. There are different graphs that can be consulted such as the frequency of

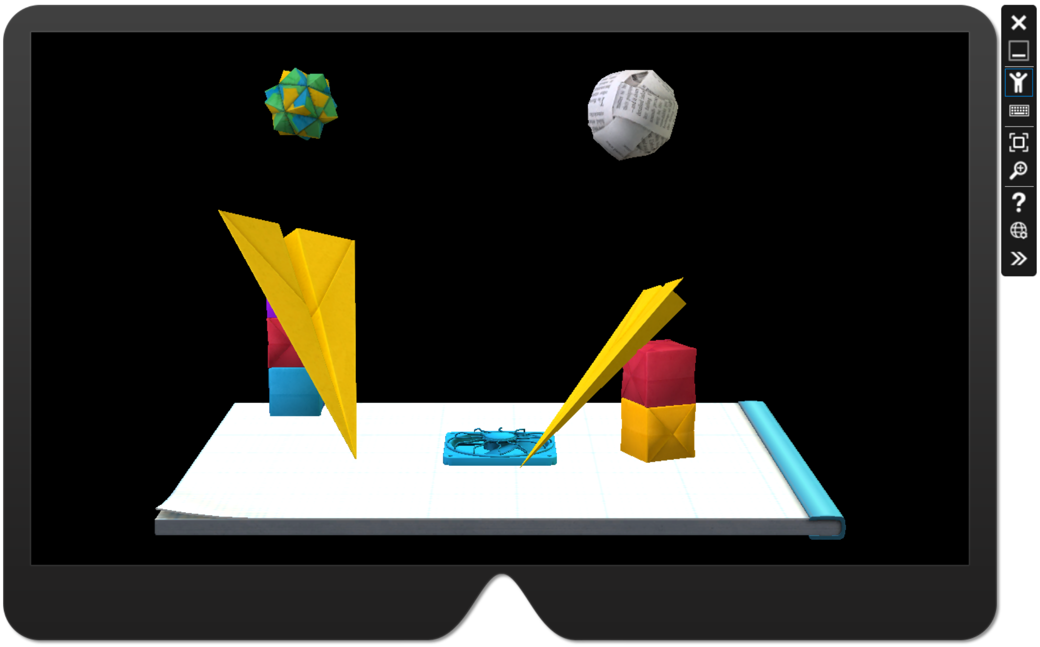


Figure 4.19: HoloLens emulator view

frames per second, the percentage of CPU, HPU and Graphics Processing Unit (GPU) in use and others.

4.3.4 Emulator

One of the tools that Microsoft provides to help in the development of applications for HoloLens is the emulator (see figure 4.19). As its name suggests, the emulator simulates the hardware architecture of the HoloLens and thus is able to load and run applications. The emulation is produced through *Hyper-V* software. It is a virtual machine management tool. It allows the creation of a virtual machine with the HoloLens requirements.

To solve the problem of emulating spatial mapping, the emulator has a room manager. This is a selection menu with 3D models of real rooms. It is also possible to load rooms from real rooms previously downloaded from the HoloLens device portal in *.xef* format.

In order to manage the network interface of the emulator the Hyper-V virtual machine has to be accessed. From there, it is possible to select which interface the emulator will access, the Internet, or creating a virtual one.

One of the main problems with the emulator is that it is not possible to simulate real streaming images from a camera. It is also important to try the applications regularly in the real HoloLens because the sensation is very important and can not be emulated.

Finally, to deploy an application in the HoloLens emulator. The project have to be opened in *Visual studio 2017* and run in the emulator option.

4.3.5 Unity

Unity is a **game engine** [Vro17]. It allows the creation of especially 3D environments for many different platforms. Unity is mostly developed in C++ and scripting languages.

But, what is a game engine? In the 90's, First Person Shooter (FPS) video games became very popular. In particular the game called *Doom* created by *Id Software* [Gre14]. The architecture of this game was clearly separated by 2 main parts. The *doom engine*, it was the part in charge of the rendering of textures, the physics, the collisions, and audio system among others, and the part that contained the most artistic side, like the models, the rules of the game, the effects, etc.. The developers realised that they could reduce the amount of work required in the next games by reusing the previous game engine or adding small improvements to it. Meanwhile, thanks of this, they could made a great artistic improvement with respect to the previous one. As a result of, the *modding* phenomenon emerged, these were modifications of a game by the community. They added to an existing game either new ways of playing or they created a totally different version of the game. Those things propelled the development of independent game engines, which sold their licenses to game developers. At the beginning each game engine was focused on a specific type of genre. Nowadays it is possible to create any kind of genre with the current game engines. Some of the most famous game engines that can be found today are Unreal Engine by *Epic Games*, *Unity* by *Unity Technologies*, *CryEngine* by *CryTek*, *AppGameKit* by *The Game Creators Ltd.* and many more [17].

What makes unity stand out from other game engines is its cost and its cross-platform. Unity is the leader in offering the deployment of applications to different platforms. there are currently 27 possible platforms for developing with unity. among them can be found for televisions, mobile devices, MR, consoles, etc. [18]. On the other hand, this tool adapts its license depending on the use that the game is given and the company that develops it, making it easier for small companies and individuals to use it. At present, the *Unity Free* version has no cost to individuals or companies with less than $100,000 gross annual income.

Like most game engines, unity has a very wide GUI and many concepts to know before programming. only some of the concepts will be introduced, otherwise the description would be quite extensive. The most important area for the programmer is the **inspector view** (see figure 4.20). This is where all the scripts and components associated with a GameObject and their values can be visualised. A **GameObject** is nothing but a container of components and scripts [Tec18]. The components are the base class attached to a GameObject. For instance the component *transform* indicates the position of a GameObject in the space, being able to initialise its scale, position and rotation. Through a script it is possible to manipulate the transform variables of a GameObject and, thus achieving movement or other possible

[17]Top game engines in 2018 `www.blog.instabug.com/2017/12/game-engines/`

[18]`unity3d.com/es/unity/features/multiplatform`

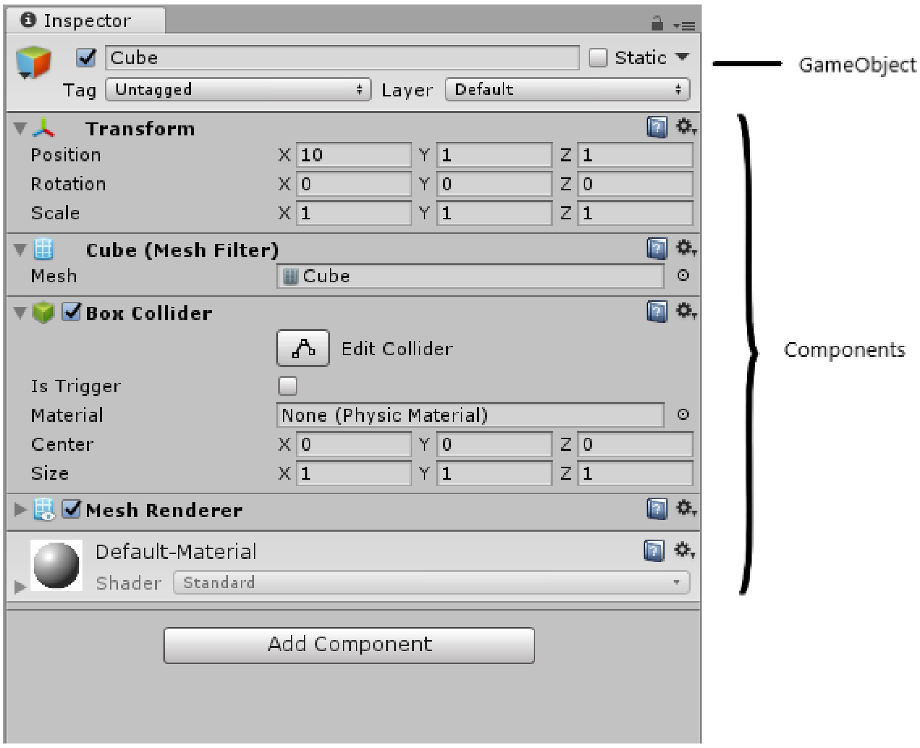


Figure 4.20: Inspector view - Unity GUI

behaviour. Once a GameObject is correctly defined it can be stored in **Prefab** form for a later use.

One important thing to consider is that unity is a **frame rate** system[19]. This means that in each frame, the system executes a sequence of events that are repeated in the next frames. The **delta time** value indicates the time that has occurred between the current frame and the previous one. The frame rate depends on the system architecture and the complexity of the program. For the HoloLens it is important to reach at least 60 frame per second[20].

Scripting

Unity uses scripting to program any type of behaviour in the game. The most commonly used languages for scripting in Unity are C# and UnityScript but others can also be used as Javascript or python [Tec18]. The base Application Programming Interface (API) of Unity scripts is MonoBehaviour. It allows to use the most basic functions for the programming of a videogame like *Update()*, the function that is called in each frame or *Start()*, which is called in the frame when the object is initially instantiated in the game. As mentioned above, the Unity system executes all the events of the life-cycle of a script repeating in each frame.

Scripting languages are transformed by interpreters also called scripting backends [Tec18].

[19] developer.mozilla.org/es/docs/Games/Anatomy

[20] docs.microsoft.com/en-us/windows/mixed-reality/performance-recommendations-for-hololens-apps

Currently, Unity supports three different scripting backends for depending on target platform: Mono, .NET and IL2CPP. However, Windows platforms support only .NET and IL2CPP.

IL2CPP means Intermediate Language to C-Plus-Plus (C++). Hence, its function is to convert the scripts into a C++ language for a later compilation. IL2CPP is a unique unity system. The cleaning IL2CPP is as follows: Once the different scripts are programmed, the unity C# game logic code is compiled to regular.NET Dynamic Link Library (DLL)s. Then, a system called *Unused Bytecode Stripper* takes care of removing classes that are not used to reduce workloads. Here is where the IL2CPP transforms the DLLs assemblies to C++ code. And finally the C++ compile the code for the desired platform.

.NET Framework is a technology that supports the compilation and execution of the latest generation of XML Web Services and applications [21]. Unlike IL2CPP, with .Net for Universal Windows Platform (UWP) it is possible to port the project in Unity to a eXtensible Application Markup Language (XAML) code solution from C# code for Visual Studio by using .NET . In summary, unity generates a project in a recognisable format for visual studio, thus delegating the previous process of compilation to it, which provides more powerful compilation and interpretation tools for windows platforms applications.

For developing MR applications it is possible to generate the project with XAML or Direct3D (D3D). The difference between those is whether the MR application uses the keyboard or not. In case of using it the project should be generated with XAML. On the other hand, D3D uses less load for the keyboard resource and therefore the application would perform a little better [22].

4.3.6 Mixed Reality ToolKit

Mixed Reality ToolKit (MRTK) is a collection of **scripts and components** for Unity game engine that aims to help and accelerate the development of applications for mixed reality windows devices. It is an open-source project [23]. This kit can be easily imported into Unity. It stands out because of the extensive documentation and examples that it provides. It also has a very active community that provides improvements and reports bugs of the tool. MRTK provides many APIs to facilitate the development of the HoloLens applications. These are divided into modules, some of the most important elements of the them are described below.

- **Input module.** It contains all the elements related to the user's interaction with the HoloLens. All the components of the cursor, also called gaze, are located here, which

[21] msdn.microsoft.com/en-us/magazine/mt590967.aspx
[22] docs.microsoft.com/en-us/windows/mixed-reality/exporting-and-building-a-unity-visual-studio-solution
[23] github.com/Microsoft/MixedRealityToolkit-Unity/

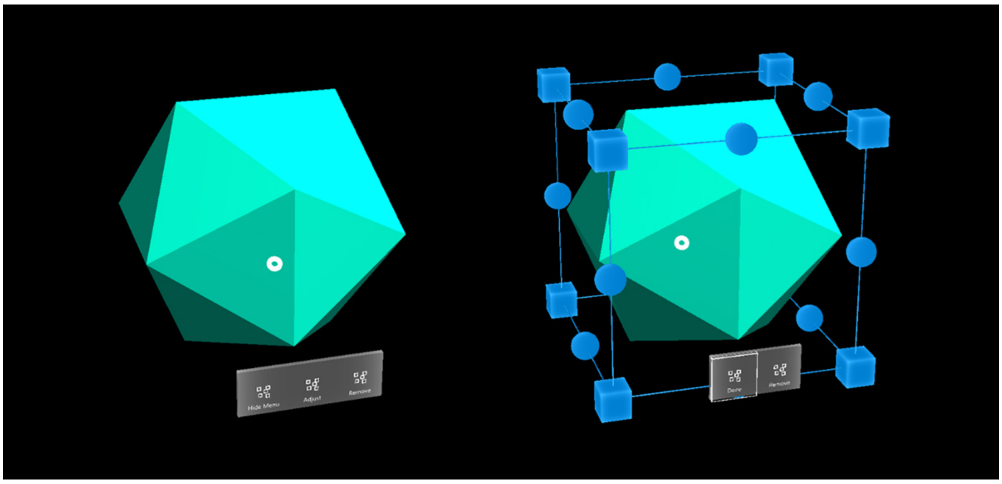


Figure 4.21: Graphical representation of a bounding box

allows us to manage the events that the user produces when making the gestures. It also provides different animations for the different gestures and designs of the cursor. This module also contains a gesture simulator for the unity simulator. It is very effective because it allows the programmer to test the application before compiling or deploying it into the HoloLens. Other utilities offered are APIs to manage the input of an Xbox controller or a voice handler, to associate voice commands to actions in the system in a more fluid way.

- **Spatial Mapping.** As mentioned above, spatial mapping allows HoloLens to analyse the contours of the space near the HoloLens. Through this module, the programmer can manage the parameters of the scanning, use its scanned meshes, or access the same points that the HoloLens detect to create their own algorithms at a lower level. Another advantage they offer is that of loading room models to simulate the scanning in the unity editor.

- **Spatial Understanding.** This API allows to generate smooth surfaces according to the previously generated meshes. It can only detect large vertical surfaces like the wall and horizontal surfaces of nearly any surface including floor and ceiling (see figure 4.22). Once the surfaces have been analysed, each surface has some data that is key to identifying the type of surface. These are the size of the surface, the height above the ground and the height above the ceiling. There is a table with a range of default values but more can be added. This table contains the name of the result surface and the values on which it would oscillate. For example a value would be the desk, with a surface between 1 square meter 2 and 2 square meters and a height of between 0.5 and 1 meter above the ground. All surfaces that match these values would be given the desk name. As a result, this allows programmers to interact with surfaces. This process is much more costly than spatial mapping. Therefore, it is designed to be

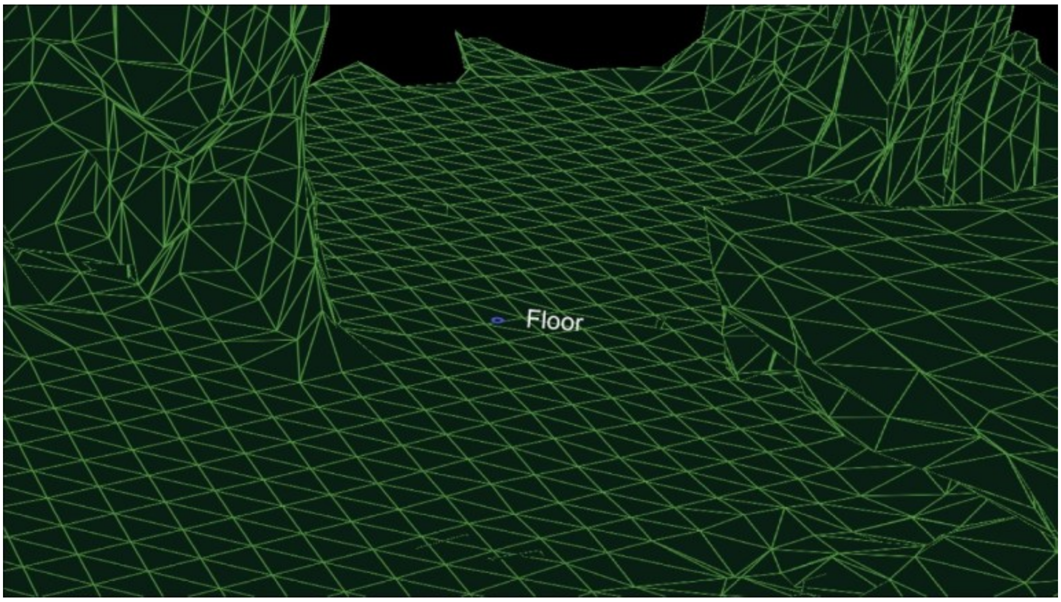


Figure 4.22: Spatial Understanding

carried out only once, at the beginning of the execution of the program or, eventually, when a specific surface is needed to be detected.

- **User Experience (UX).** This module provides all kinds of elements for the user interface that improve the application. Some examples are as follows:
  - *Bounding Box* is a component that allows the user to manipulate the physical properties of a hologram. It means, its size, position and rotation. When a hologram is selected, a box is displayed around the object with all the options in an intuitive way (see figure 4.21).
  - *Keyboard*, which is a similar to the one that is found by default in the HoloLens. Thanks to this element, writing can be implemented quickly.
  - *Interactive buttons.* MRTK contains prefabs of the interactive button 3D models that most Integrated Development Environment (IDE)s have and that are used most often. There are checkboxes, toggle buttons, sliders, loading animations and many more.
  - *Solver system* allows among others to create the sensation of magnetism on a hologram so that the hologram follows the user. In other words, they are components that are normally added to messages or holograms that users should not lose sight of. This is very useful because HoloLens has an unlimited interface and it is necessary to be able to control which elements need to be visible at any time.

Finally, there are other modules less used and more specific depending on the purpose of the application. These are *Spatial Sound*, *Utilities*, *Build* and *Sharing*.

4.3.7 Universal Windows Platform

UWP is a extension of Microsoft of the Windows Runtime platform that leverages the C++, C#, VB.NET, and XAML programming languages [24]. It is native for all the devices with Windows 10 OS. UWP uses the new .NET Core Framework. It offers a universal API for all different platforms with windows 10. It also offers services such as cloud, *Cortana*, artificial intelligence and cognitive services among others. UWP applications use the same distribution store, *Microsoft Store*. Lastly, UWP provides a unique cross-platform SDK, but depending on the objective of the application developed and the device target the developer can use different libraries. For example, for developing in HoloLens it can uses the standard libraries for UWP, plus specific mixed reality HoloLens libraries.

A common problem could arise when a device application for UWP is developed from a external IDE [Vro17]. It is a framework version problem which causes that some libraries could not be used in the IDE.

In case of developing a UWP application for the HoloLens in Unity this problem often happens. In this case, Unity compiler uses a different .NET version than the device target. The reason unity uses an older version is because it is more stable and can span more platforms, not only to UWP platforms. Since, a platform with a lower version of .NET could not run a higher version but the opposite can work. As a result, the developers have to code two times the same but in different ways, one for testing and debugging in the Unity editor with his emulator, and another one for executing in the HoloLens. The solution to this problem is to signal the compiler, with a condition sentence, which part of code will be run in the HoloLens and which with Unity. Otherwise, the Unity compiler will not allow to compile and deploy the project (see listing 4.1)

```
#if UNITY_EDITOR
    void Start () {
        //IN THE UNITY EDITOR
    }
#endif
#if !UNITY_EDITOR
    async void Start(){
        //IN THE HOLOLENS
    }
#endif
```

Listado 4.1: Compiler conditional in C# scripting

[24] www.windowscentral.com/what-makes-uwp

4.4 Computer-based music education systems

There are many ways to represent music and these have evolved over time. In the case of occidental music, the oldest music written was found in Greek and dates back to the 3rd century B.C. [BGP06]. Until the 8th century A.D., music was transmitted orally. From here, the first musical notes began to form since monks made annotations about Gregorian chants. As a result, the need to represent music in written form began to arise. It was not until the 10th century when the first music notations began to be standardised thanks to the copyist monks. The first musical notation were names of the sound that was written over the lyrics of the Gregorian Chants. Today there is a complex notation system that, like the languages, has differences depending on the geographical area.

The most basic and main element of the music is the **note** or tone. A note represents a sound and, therefore, a wave frequency. There are a total of 12 notes in a octave according to the occidental music notation system. The smallest measure that separates the distance between the notes is called **half step**, two half steps (or semitone) are called **step**, or whole step (or tone).

To be able to explain what an **octave** is, it is necessary to take into account that in a musical piece it is always based on a specific frequency. The entire note system is deployed on this **base frequency**. The note used for this is A (La). Normally the value given to the note A is 440 Hertz. However, it can be variable depending on the tuning of the instrument. According to the International Organization for Standardization (ISO) 16:1975[25], the standard tuning frequency of the note A is 440hz.

Once the value of the tuning note A is given, the entire note system is displayed. The frequency between a note and its higher octave is two times the frequecy of the first [Mar08]. In the equation 4.1 where *a1* correspond to a note frequency the octave is the entire number correspond the integer number corresponding to the distance of octaves from the note a and *a2* is the frequency at which the note a is found in that octave.

$$a_1 * 2^{octave} = a_2 \tag{4.1}$$

The reference note A is located in the fourth octave. Taking into account this, the note A in the next octave, number 5, would correspond to 440hz * 2 = 880hz or, on the contrary, in the 3rd octave it would correspond to 440hz / 2 = 220hz.

To be able to calculate the frequency of a note with respect to the reference note A, it is possible to calculate the distance ratio of a semitone between notes. Taking into account that between one octave and another there are a total of 12 semitones and the ratio between two

[25] www.iso.org/standard/3601.html

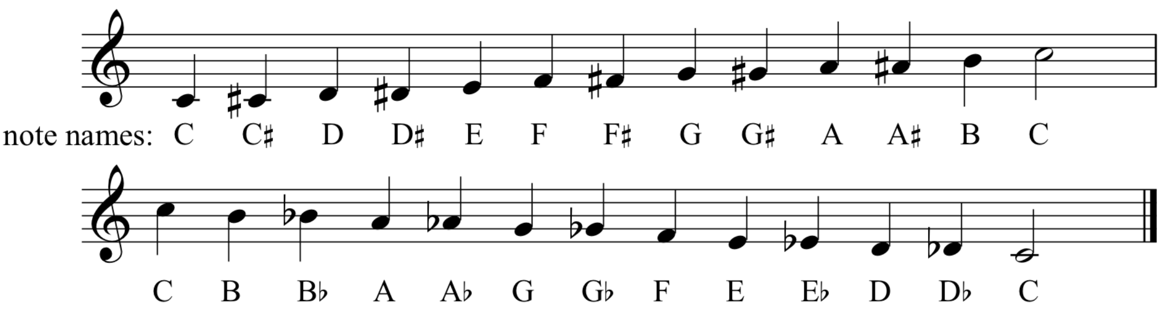


Figure 4.23: Chromatic scale from note C4 to C5

octaves is 2, the formula would be the following 4.2, where *hs* is the half step ratio.

$$hs = \sqrt[12]{2} \tag{4.2}$$

Therefore, through this *hs* value it is possible to calculate the frequency of any note in the system giving this resulting formula 4.3. Where *a* correspond to the frequency of the note of reference, n integer number of half step distance that is wanted to be calculated and b the frequency of note resulted.

$$a * hs^n = b \tag{4.3}$$

As explained above, there are a total of 12 notes in an octave. however, there are a total of 7 different note names that are associated with the frequency of the reference system used in function of A. These are: **A (La), B (Si), C (Do), D (Re), E (Mi), F (Fa) and G (Sol)**. In order to represent the other 5 remaining notes, the **flats** and **sharps** are used. These indicate that a note has a distance of one less half step in the case of the flat and more half step in the case of the sharp.

A **staff** is used to represent a note. A staff is nothing but than a container of notes based on 5 horizontal lines [Kor05]. Notes can be placed between a line or a space. The height refers to the frequency. The higher a note is in the staff the higher frequency has and the opposite. In the following figure (see figure 4.23) it can be observed how the 12 semitones of an octave are represented in a staff. The succession of notes that are separated by a half steps in an ascending or descending direction is known as the chromatic scale

The figure at the beginning of the staff is used as a reference for the notes on the staff. This is called a **clef**. There are many different varieties. They indicate the note to which each space or line corresponds and its octave. For example, the clef in figure 4.23 is known as Treble. In the Treble, the note G corresponds to the second line and of the octave number 4. From there, all the other notes can be deduced.

Before explaining the different note values it is important to introduce the **rhythm** [Kor05]. Rhythm is nothing but the measure of time used in music. The value of the rhythm is represented by the number of **pulses per minute**. In other words, if the rhythm value is 60, 60 pulses per minute are performed, which is 1 pulse per second, like a clock pulse. This value is usually represented at the beginning of a music sheet and can change during the piece.

A **measure** is the space on the staff that groups the notes together according to the **time signature**. The measures are separated by a single vertical **bar line** on the staff. The time signature is indicated next to the clef and indicates how many parts of time can be stored in each measure. As can be observed in the picture 4.24 the time signature is given by two values. The lower number indicates which type of note value that is used to represent a pulse. The upper number indicates the number of figures of the referenced type that can fit on a measure. As a consequence, the value of the figures of a note will depend on the rhythm and the time signature. The 4/4 time is one of the most used and simple to understand.

Figure 4.24: Time signature of 4/4

The representation system of the **note values** is simple and is based on the division of parts. The note values, as the name suggests, are the time values given to the notes. The figure with the highest value is the unit, called **whole note**. The whole note would fill a whole measure in a 4/4 time signature. Henceforth, the whole system of values representation is based on the division by two of the previous figure (see figure 4.25). As a result, the notes values are: **the whole, the half, the quarter, the eighth and the sixteenth**.

When it is desired to indicate that the notes are played at the same time, this is represented by placing the notes in the same imaginary vertical line. This is called **chord** (see figure 4.26). Depending on the distance between the notes, chords can be a different type. However, these will not be explained due to their complexity.

Finally, a note that is as important as any other is that of silence. As its name indicates, the rest symbol indicates the absence of sound in a part of time. The rest value can be the same as a note but it represented differently (see figure 4.27).

4.4.1 The piano

The piano is an instrument of the category of **percussion instrument**. The reason for this is the way it works. By default a piano usually has 88 keys, 36 black and 52 white, a total of 8 octaves. However, this can vary depending on the piano. The operation of a

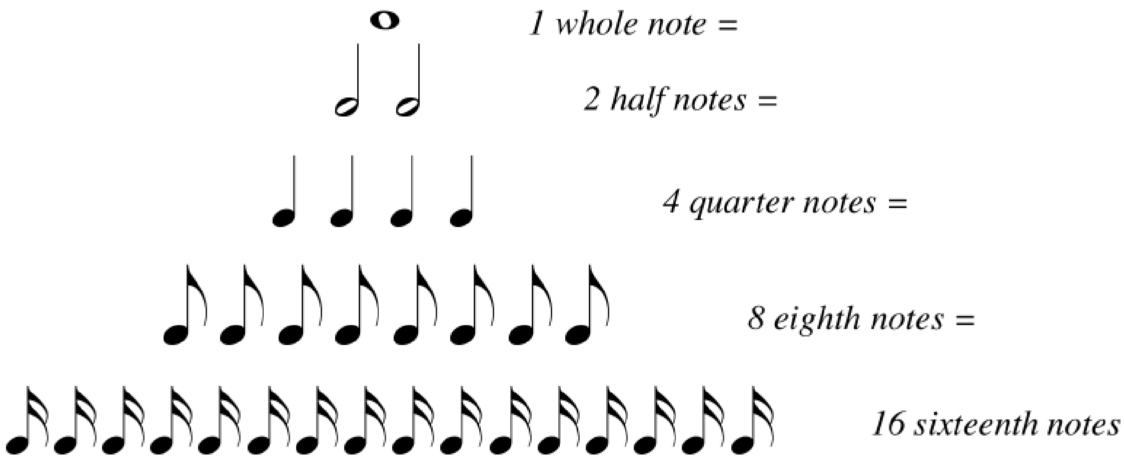


Figure 4.25: Diagram of the note values

Figure 4.26: Chords

Figure 4.27: Rest note values

traditional piano consists of a system of hammers that, when a key is pressed, activate the mechanism of the hammer, producing a hit on the string and, therefore, the sound. Each key has a different hammer and string associated with it. The strings are tightened looking for the desired frequency when it is hit.

The most important keys of the piano are the centre keys. These correspond to the octave number 4, which is used as the reference of the music system. The white keys represent the pitches with a name, i.e. A, B, C, D, E, F and G. On the other hand, the black keys represent the white keys altered one half step, a half step more, if it is on the right, and a semitone less, if it is on the left. The distance from one key to the nearest key is always one half step (see figure 4.28).

The traditional piano is played with both hands. But, it also has foot pedals that can modify the sound or the length of the notes. Since it has 2 hands, the notes of the score

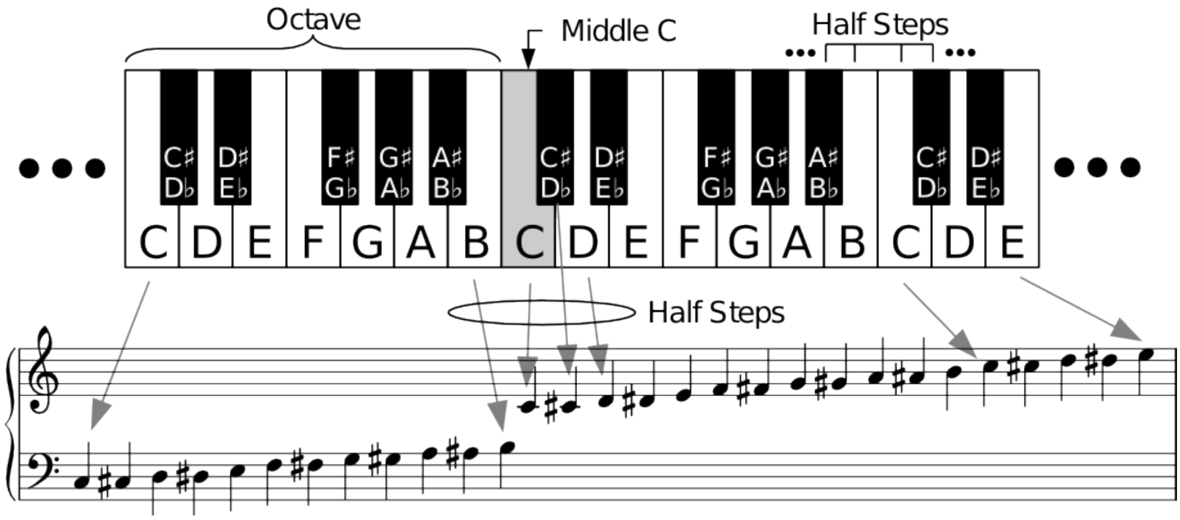


Figure 4.28: The piano keys related to the staff notes

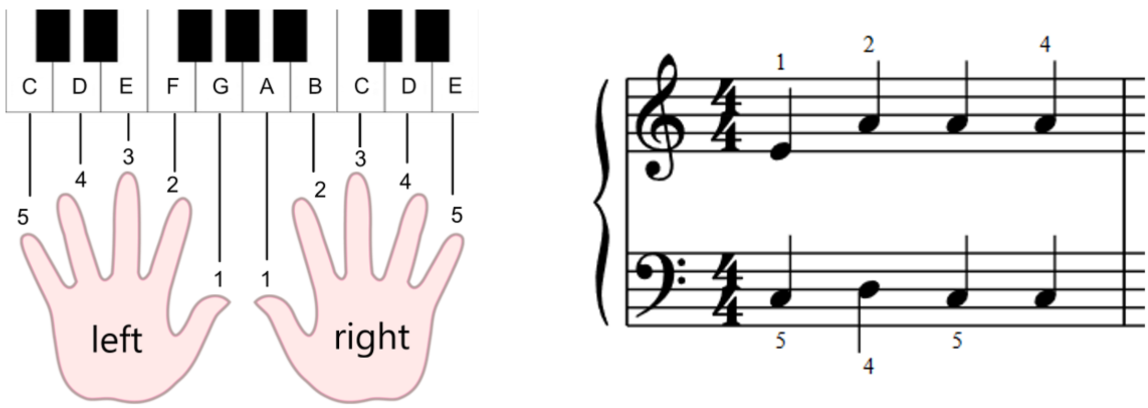


Figure 4.29: Piano Fingering Number System

are represented in two separate but linked staffs. The right hand plays the top staff and the bottom voice is played by the left hand (see figure 4.28). However, this is not always on.

To play the piano properly, it is advisable to sit with the back straight, with relaxed shoulders and creating a 90 degree angle at the elbow. The keys should be played with the fingertips and not with the whole finger. Each finger has a number associated with it (see figure 4.29). In the staff, the numbers are written next to the notes in order to facilitate the positioning of the fingers on the piano (see figure 4.29). The piano fingering are also useful tips to help the interpreter to position himself for the next note.

The digital piano

The music has also been modernised and some of its instruments have managed to adapt to the digital age. As explained above, the traditional piano system is simple and if it is simplified, it is based on the fact that each key emits a different sound. Therefore, a **digital**

piano is a system that receives an input, which is the value of the key, and emits an output, the sound corresponding to the key. As a result, many different digital pianos have been created. The standard digital piano consists of a keyboard identical or very similar to a traditional piano with a sound board that processes the information and reproduces the sounds. One advantage of this piano is that it can play any sound as desired.

Another advantage is that the sound it produces can be processed by another machine and used for other purposes. Another advantage is that the input data it produces can be processed by another machine and used for other purposes. The communication protocol used to send these signals is known as Musical Instrument Digital Interface (MIDI).

MIDI Protocol. A MIDI message has a total of 3 bytes [RM12]. Each byte represents a data type. The first byte represents the status and defines the meaning of the following two bytes. The midi protocol has many different types of messages but we will only focus on the behaviour of the keyboard. The messages are represented in hexadecimal (see figure 4.30).

- *1st byte - status.* The two cases that can occur on the keyboard are when a key is pressed and when it is released.
  - **Note on.** When the key is pressed, its value is 9n. where n can be any number since it is not relevant.
  - **Note off.** When the key is released, its value is 8n. where n can be any number since it is not relevant
- *2nd byte - data.* The second byte, taking into account the values of the state 9n or 8n, indicates the id of the note pressed. Since there are 128 different possibilities in a byte, there are 128 total notes that can be recorded (0-127), a total of approximately 10 octaves. The central note C, of octave 3, has the id of 60.
- *3rd byte - data.* this byte, according to the previous ones, indicates the velocity. This data determines how hard the key is pressed or the volume value of the key is pressed.

4.4.2 Music education applications

It is also in the music field where computer systems have been developed to help and guide the music student. The study of music usually begins when the student is a child and usually accompanies him or her during all stages. Studying music not only involves playing an instrument, but also learning many theoretical concepts in order to understand the music and know how to interpret it according to the context of the piece. The learning of an instrument is normally monitored by an expert tutor in that instrument.

In 1990, *Roger B. Dannenberg* with two piano teachers, developed the first expert teaching system for piano beginning students [DSJ+90]. It was called **Piano Tutor**. The system was

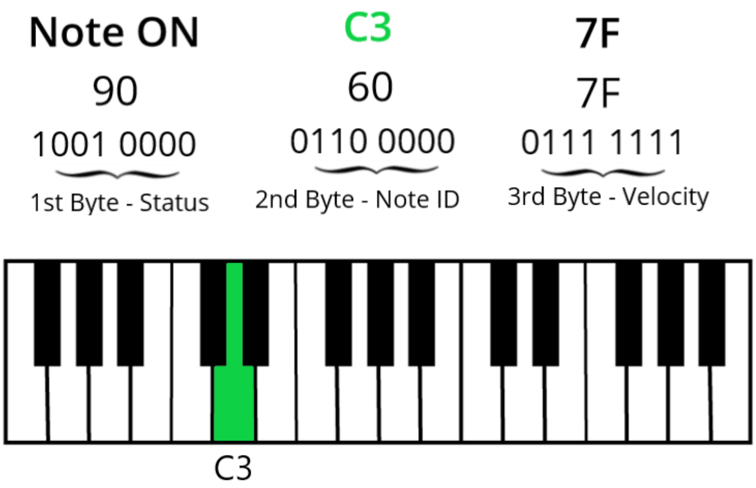


Figure 4.30: MIDI data parsing example

developed in Common Lisp and C for the apple Macintosh computer. It used two main elements, the computer and the digital piano. The MIDI data from the piano was processed by the system. The application consisted of a set of exercises in the form of music scores that the student had to play at the rhythm of the metronome. the system automatically turned the pages and could detect if there were any errors in the interpretation of the student. Finally, I returned feedback and suggested ways to improve.

Later, in 1995, **PianoFORTE** was released. It was described as a system for piano education beyond notation literacy [SWK95]. According to the creators of this system, two clear separations could be made in the way of studying a piece of music. The first part consists of simply being able to play the music without mistakes made. This is the longest process and the developers cite that when the student is learning, the teacher rarely emphasises in these aspects, since it is the student with the practice that can overcome the mistakes. On the other hand, when the piece is dominated, the teacher makes emphasis on the way of interpreting the piece and on perfecting its execution. This last part is the part that *PianoFORTE* tries to systematise. The architecture that supports this system is very similar to the *Piano Tutor* system. As it would be very difficult to evaluate an interpretation of an art like music, *PianoFORTE* analyses the small details in the performance. These are the exact measurements of note values, how the articulations are interpreted, the intensity between both hands, or the synchronisation when playing a chord or multiple notes at once. As a result the system displays the music score with feedback about the specific elements of the score that were played according to the above requirements.

In 2005, the first systems of teaching music through augmented reality began to emerge. The project called **Piano Tutor AR** offered a new way of representing music thanks to augmented [CFAW13]. This project was based on the same architecture as the previous ones, with a MIDI keyboard and a computer. However, it added one more element, the webcam. A marker placed on top of the piano was used to track the position of the piano (see figure 4.31a). The AR interface provided instant feedback on the notes played through colour over the keys. However, the positioning of the keys was not precise and had limitations. Nevertheless, it was one of the pioneers in introducing augmented reality to teach music.

Although the piano is one of the most played instruments and perhaps easy to implement thanks to the MIDI protocol, other applications have been developed for different instruments. This is the case of **Digital Violin Tutor** [YWH05]. It was realised in 2006. The architecture of this system is completely different from the previous piano systems. This creates a graphical interface on the screen of the 4 window layout to display the information. The first is a video of the teacher playing the piece, the second is a 3D animation of the performance of the song, the third indicates the feedback of the interpretation, and the fourth the positioning of the fingers on the violin. What makes this system special is its sound recognition system. This system compares the interpretations of the teacher with the interpretation of the student through the audio collected by a microphone. Analysing the sound frequency of the recordings extracts the notes and compares the results.

Due to the HMD growth, some education music projects started to adapt augmented reality to these devices. This is the case of the project developed by the University of Auckland, New Zealand, in 2013. It is called **P4P** [CFAW13]. They proposed a system that would combine the basics concepts of *Piano Tutor*, with the advantages of augmented reality through a HMD and, adding simple gamification elements to attract more attention of the students (see figure 4.31b). The architecture of the system was similar to that of the piano tutor but added the element of the HMD which computed the video processing from the central computer. The interface of the notes consisted of a number of lines that fell on the keys indicating when they should be pressed. In order to carry out the precise recognition of the piano, a total of 5 marks were placed around it in a strategic way. In this way, the measurement of the piano was indicated. The project was implemented in C# language with .NET framework. For capturing images from the camera OpenCV was used. And finally, For detection and tracking of markers, they used *NyARToolkit*, a port of *ARToolkit*.

4.4.3 HoloLens piano education systems

As mentioned in the previous section, HoloLens is an OHMD wearable computer. This allows developers to create simpler architectures since it eliminates a main element that was the central computer, which processed the images of the HMD device. As a result, some developers have already developed applications based on learning music, specifically piano, for the HoloLens.

Currently, one of the most famous projects in this field is **Music Everywhere** [26]. This

[26] www.music-everywhere.co

a) AR Piano Tutor b) P4P

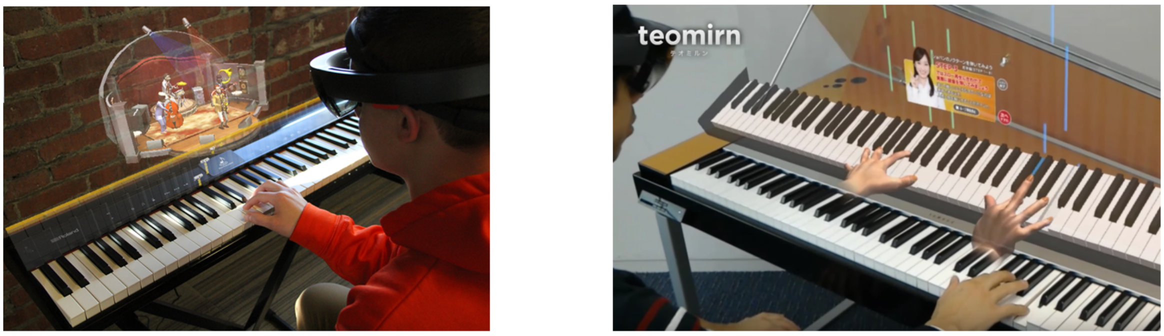
Figure 4.31: Augmented reality music education systems

application offers lessons about how to play the Piano. In order to teach the digitisation of the fingers, holograms in the shape of a hand are placed over the keyboard and simulate that a teacher playing the song (see figure 4.32a). Another of the features of this application is the possibility of playing the songs accompanied by other instruments simulating a band. While playing, on the top of the keyboard, a stage with an animated band is displayed. This application won the 1st prize of the *Unity / Microsoft HoloLens Contest* in 2017 [27]. Its architecture is based only on the piano and the HoloLens. The communication between both is made via Bluetooth using the MIDI protocol. It uses a layout over the piano with the keys in transparent to indicate which note should be pressed. The piano is positioned manually by the user, using the spatial mapping capabilities of the HoloLens.

Finally, **Teomirn** [28] is another application for HoloLens that teach how to play the piano. The aim of this system is to simulate a tutor to guide the user in the learning process (see figure 4.32b). For this purpose, several solutions are offered. One of these is to generate a virtual person who plays the music score so that the user can observe it. The other solution is based on the interpretation, the user plays the piece while in front of him two holographic hands are placed where the user should play as an example. One of the most powerful elements of this application is its level selection menu. This is directly connected to the *Teomirn* website. The user can select the piece to interpret from a 2D GUI. The advantage of this is the high scalability it provides as the levels are independent of the system.

[27] unity3d.com/es/partners/microsoft/hololens-contest

[28] uploadvr.com/teomirn-mixed-reality-app-teaches-play-piano/



a) Music Everywhere b) Teomirn

Figure 4.32: Hololens music education systems

Chapter 5

Methodology

5.1 Working methodology

The development of a innovation system has to be guided by a development methodology that could manage a fast changing development. Agile development methods include several characteristics that support these requirements. [Lar04] These methods usually apply iterative and evolutionary development, employ adaptive planning and promote incremental delivery among others. In other words, an agile development process implies a fast and flexible response to changes.

In the case of this system, it has been proposed to use a new technology that is still under development, the HoloLens. This causes a difficulty in estimating final results and deadlines of the project because it is an unexplored field of which the boundaries are unclear before the development of the project and of which there are few previous experiences.

On the other hand, the project is tackled individually with the eventual revision of the directors. Thus, it is important to keep a continuous communication and to have meetings in short periods of time. These are some of the reasons why it was preferred to use an **agile development process** for this project.

5.2 Development methodology

As mentioned above, according to the system requirements, it has been used an **iterative and incremental software development methodology** adapted to the specific needs of the project.

Iterative and incremental methodology is the adaptation of the *waterfall methodology phases* to the agile development process [LB03] with some modifications. As the name suggests, this methodology is based on the repetition of a series of phases, called iterations, in a continuous development to achieve the desired final product. Each time an iteration is completed, the results obtained are evaluated and, if the objective is not achieved, the next interaction work is planned.

As it is an agile methodology, it requires that the times between iterations should be relatively short but adjusted to the type of project and work team. In the case of this project, the time between iterations varied according to the needs of the project.

Some of the phases used in the iterations of this methodology are based on the waterfall model. However, there is a new phase called planning where the work to be done in the iteration is planned. Contrary to the waterfall methodology, the work applied to each phase will change depending on the needs of the project, giving rise to situations where there is no work on any phase. These are the phases of the iterative and incremental methodology (see figure 5.1) [LB03]:

- **Initial planning.** Before starting to develop a project, an initial planning is carried out to evaluate the scope of the it and discuss how it will be addressed. In this project, the topics discussed in the initial planning were the following:
  1. Evaluating the initial idea.
  2. Studying the scope of the project and evaluating whether the objectives to be achieved were possible in the time limit established.
  3. Organising the working methodology, establish working hours and a development methodology.
  4. Acquiring the material and resources necessary to develop the project.
  5. Setting a possible initial goal to create a solid base for progressing in the next iterations in the future.
- **Analysis.** It consists of listing the problems that it is planned to deal with in the iteration of the project in a structured way.
- **Design.** In this phase the problems analysed above are studied and outlined to have a clear idea of how to solve them.
- **Implementation.** Once we have studied how to solve the problems, the solutions to them are implemented in the system.
- **Testing.** In this phase tests are performed in order to verify that the system works correctly.
- **Verification.** In this phase tests are performed in order to verify that the system works correctly. During the verification process, the final objective of the project is discussed. If it has been reached, the deployment phase will proceed. Otherwise, the iteration cycle will continue until the next iteration.
- **Planning.** In the planning the results obtained in the iteration are evaluated and compared with the analysis previously realised. Based on this, the next objectives for the next iteration are defined.
- **Deployment.** It is the last phase that closes a project. In the case of this project, the documentation and presentation of the project have been prepared during this phase.

In this project, the time between iterations changed according to the current objectives set out in the plan. It usually scheduled every two weeks. During the meetings, the verification

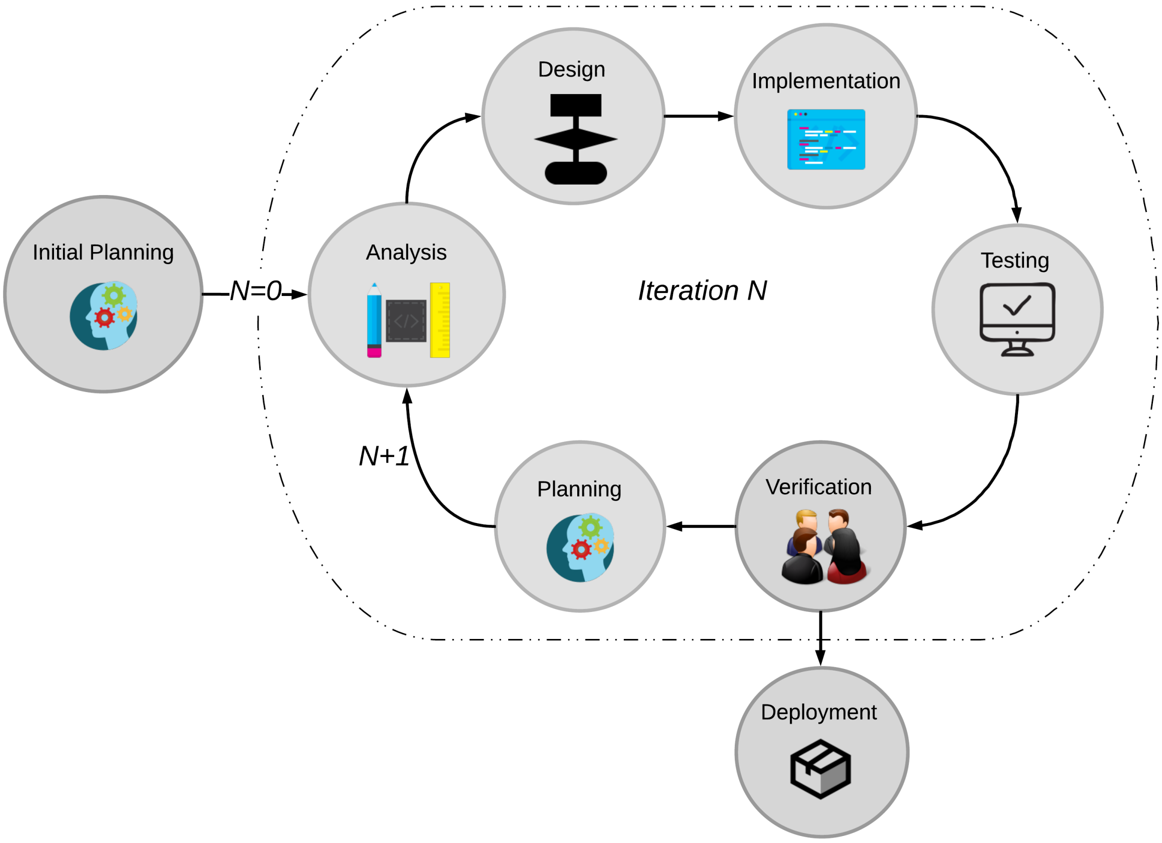


Figure 5.1: Iterative and incremental methodology phases

phase was approached, in which the directors discussed with the developer the milestones achieved and the problems that had arisen.

The project was developed in the *ORETO/AIR*[1] laboratory of the *Escuela Superior de Informática* in Ciudad Real, where all the necessary elements for the development were available. In the figure 5.2, it can be observed the working environment where the project was developed

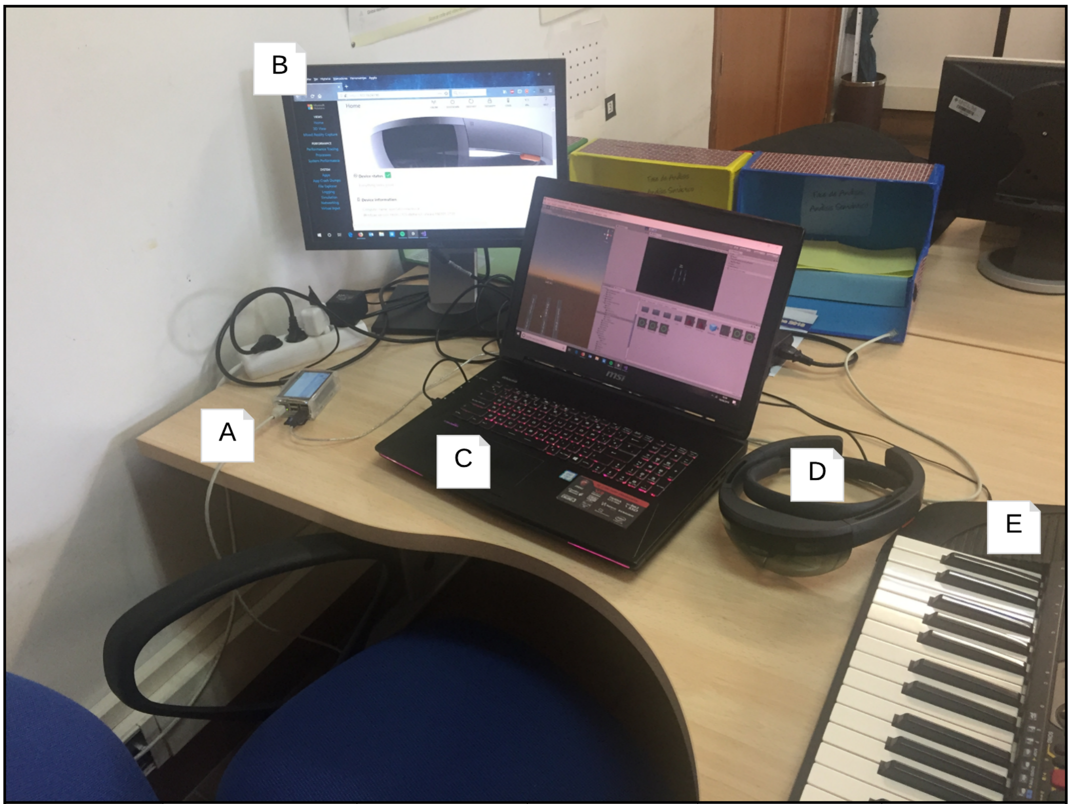
5.3 Resources

In this section it will be presented the assets and equipment used along the development of the project.

5.3.1 Hardware equipment

- **MSI GT72VR 6RD Dominator with Nvidia GTX1060** [2]. It is an upmarket laptop that provides high performance in graphical processing thanks to its powerful graphics card.

[1] air.esi.uclm.es
[2] es.msi.com/Laptop/GT72VR-6RD-Dominator.html



Index	A	B	C	D	E
Device	Raspberry Pi	Monitor	Laptop	HoloLens	Digital Piano

Figure 5.2: Working environment where the project was developed

- **Microsoft HoloLens** [3]. This device has been granted by Furious Koalas Interactive [4] for the development of the project. As already described in chapter 4, it is an Optic Head Mounted Display (OHMD) wearable computer that allows the visualisation of holograms in the environment. The HoloLens launched in November 2017 by *Microsoft* but can only be acquired for development purposes.

- **Digital piano keyboard with MIDI output**. It has been used a 49-key *Casio* digital piano keyboard in conjunction with a MIDI-USB Converter.

- **Raspberry Pi 3 Model B**. [5] It is an economical single-board computer launched in 2016. It has been designed to handle low cost computational tasks. It can be expanded with complements. In the case of this project, it has been incorporated a LCD 3.5-inch touch screen [6].

[3] developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details
[4] fksl.es
[5] www.raspberrypi.org/products/raspberry-pi-3-model-b
[6] www.amazon.es/gp/product/B01FX7909Q/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1

5.3.2 Software resources

Operative Systems

- **Windows 10 Pro with the update Fall Creators**. This version of the Windows OS was launched in October of 2017. It is a mandatory requirement for the using of mixed reality developer tools.
- **Raspbian Jessie** [7]. It is the OS that runs on the raspberry. It is a distribution of Linux provided by the Raspberry Pi Foundation optimised for low-performance ARM CPUs, like the architecture of the Raspberry Pi 3.

Development tools

- **Unity** is a cross-platform game engine developed by Unity Technologies, which provides support for the creation of video games in mixed reality. It was described in more detail in the chapter 4.
- **Visual Studio 2017**. It is an IDE developed by Microsoft. It has very powerful tools for the development of applications for systems based on Windows OS like the HoloLens.
- **HoloLens emulator**. It is a tool that runs the solutions into a virtual environment emulating the Hololens for developers proposes. It was described in more detail in the chapter 4.
- **Hyper-V**. It allows to create virtual machines on x86-x64 systems running on a Windows machine. It is needed to run the Hololens emulator. It was described in more detail in the chapter 4.
- **Git**[8]. It is a version control software used to improve system process development. An additional software called *Sourcetree* [9] has been used to speed up the management of the project. The repository of the porject is located in is located in Gitgub[10].
- **Putty**[11]. It is an Secure Shell (SSH) and telnet client, Windows platform which allows to manage other machines remotely.

Libraries and SDKs

- **Vuforia**. It is a powerful SDK for MR devices which uses the camera for object and marker recognition. It was described in more detail in the chapter 4.
- **Mixed Reality Toolkit**. It is a collection of scripts and components intended to accelerate development of applications targeting Microsoft HoloLens and Windows Mixed Reality headsets in Unity. It was described in more detail in the chapter 4.

[7] www.raspberrypi.org/downloads/raspbian/

[8] git-scm.com

[9] www.sourcetreeapp.com

[10] github.com/DiegoMolero/PianoMR

[11] putty.org

- **HoloLens Shader Pack**[12]. It is a set of materials and textures for for unity with futuristic thematic and optimised for the HoloLens.

Programming languages

- **C#** [Vro17]. The programming language used for the creation of scripts in Unity. It is native to .NET and provides a better code completion feature in Visual Studio.
- **Python** [AL99]. The programming language used for establishing connection and sending the data received from the piano to the Hololens.
- **Shell Scripting**[13]. Shell Script (`SH`) is the is the scripting language used to synchronise and automate the process of managing and sending the piano data.

Documentation tools

- **LaTeX**. This document has been created with the LaTeX text editor. This software was created by *Leslie Lamport* and released in 1984. It is open source software and is especially used for scientific articles. The class used is *esi-tfg* [14] and was developed by the *ARCO Group* [15]. The platform used for the development in LaTeX of the project has been *Sharelatex*[16].
- **Lucidchart**[17]. It is a tool for designing professional diagrams online. It was used to generate the diagrams of this report.
- **Gimp**[18]. It is a powerful tool for digital image editing. It was released in July 1995 and is open source licensed. It has been used to edit and create images for both, the documentation and the system implementation.

[12] `assetstore.unity.com/packages/vfx/shaders/hololens-shader-pack-89989`
[13] `www.shellscript.sh`
[14] `bitbucket.org/arco_group/esi-tfg`
[15] `arco.esi.uclm.es`
[16] `www.sharelatex.com`
[17] `www.lucidchart.com`
[18] `www.gimp.org`

Chapter 6

Architecture

In this chapter it will be presented the architecture of the system. The main problems and how they have been solved in detail will be also discussed from the point of view of the developer. In order to explain them correctly, a top-bottom approach will be adopted, supported by graphical diagrams.

6.1 Architecture overview

In the Figure 6.1 below it can be observed the diagram of the architecture. The architecture design follows a layered model. The advantage of this model is that as each layer is separated by a specific functionality, it is possible to have greater control and organisation over the development of the system. Each layer and its components will be explained in detail in the following sections. The structure of this system can be classified into 3 main blocks or layers that are related between them. Each layer contains modules which represent a set of components that solve specific problems of the system.

- **Communication layer:** It is the lowest layer located in the diagram and also represents the lowest level of the system. Its function is very important and is considered as base of the system. It is responsible for guaranteeing that the raw data generated from the piano reaches the HoloLens correctly. It is composed of two modules:
  - *Piano Input Module* is the module in charge of collecting the data from the piano and storing them in order to utilise them. The piano is connected by USB to the system responsible for receiving the data. Specifically, it uses a MIDI USB converter which transforms the keyboard input into MIDI data.
  - *Network Module* is responsible for the entire communication process between the piano and the main system, the HoloLens. In other words, it is a module that acts as a bridge between the piano and the HoloLens. The communication with the *piano input module* is done via local UDP ports, while communication with the main system is done via TCP sockets.

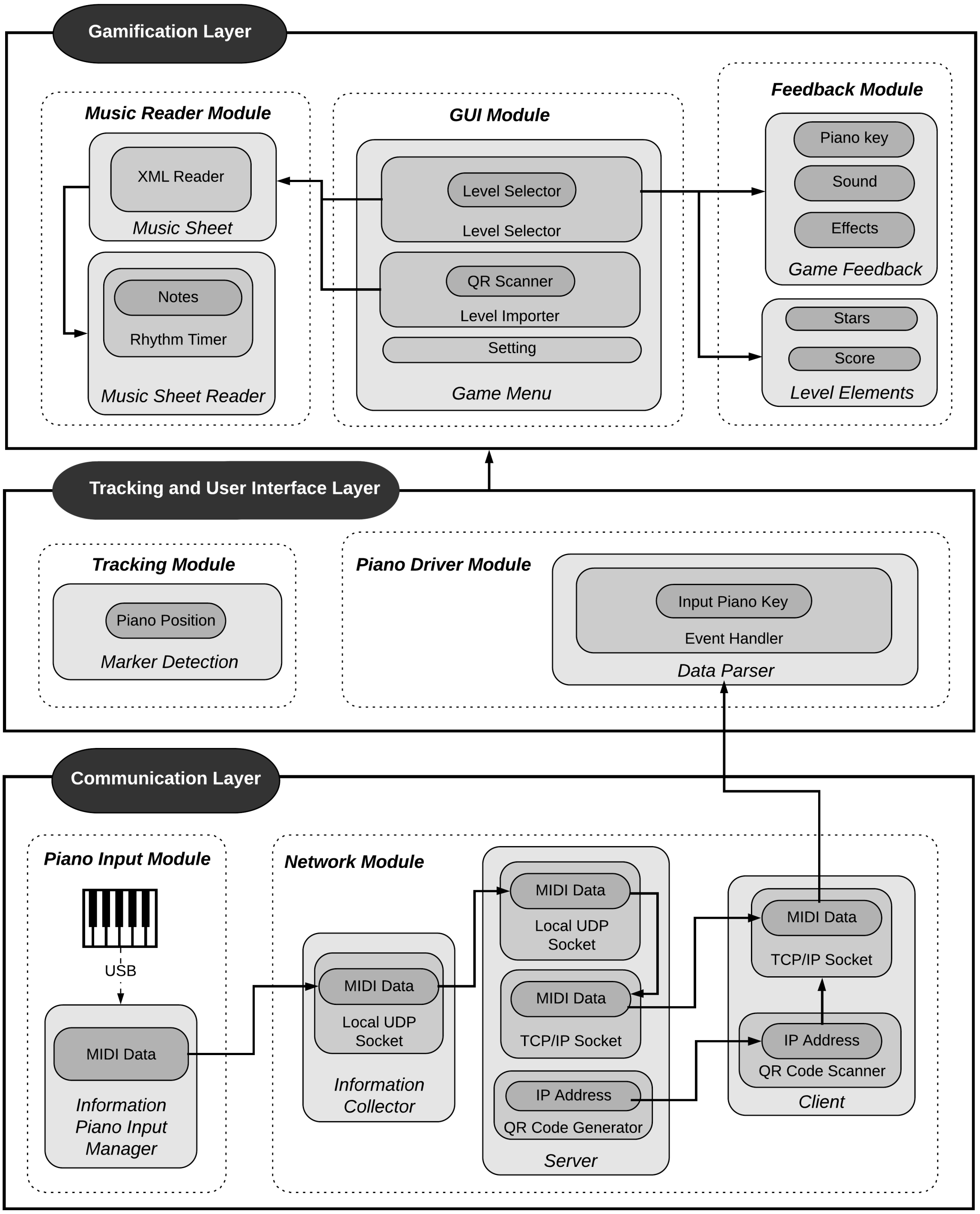


Figure 6.1: Detailed architecture diagram

- **Tracking and user interface layer:** This layer acts as an intermediary between the communication and the gamification layer. It is mainly responsible for the correct handling of the user's actions as well as the deployment of the system. It has two main modules:
  - *Tracking Module* locates the position of the piano, which will be the reference to display the whole system. This is achieved through the recognition of markers provided by the Vuforia SDK, in which the piano keys represent the marker that has to be identified.
  - *Piano Driver Module* interprets the data received by the piano and transform them into events. In other words, it is in charge of converting the MIDI data of the piano that collects the communication layer and transforming into direct actions on the virtual keys of the system. As can be observed in the figure 6.1, the piano data stream reaches this module. It processes the data and sends them to the Gamification layer which will receive the processed data and act as the game controller.
- **Gamification layer:** It is in charge of the entire game-oriented and graphical part. This part is also conceived with the highest level of abstraction, since this part is totally independent from the Hololens architecture. It is divided into 3 important modules:
  - *Music Score Module* is in charge of the reading logic of a virtual music score for its reproduction in the game. It is divided into two parts:
    * The *Music Sheet* contains all the information about the song, which notes it contains, in which measure they are belonged to, how fast the song has to be played, etc.
    * The *Music Sheet Reader* collects and processes the score information according to the time the song is played, and creates the notes in the game in order to be played in the piano by the user.
  - *Feedback Module* is the module in charge of rewarding the player by means of achievements, good comments in the form of audio and a scoring system among others.
  - *GUI Module* represents the visual options that the player can select in the game, such as the selector level menu, the setting menu, the instructions to follow in order to scan a level, etc.

  The first one, the *Music Score Module*, is in charge of the game logic. The second, GUI module, is responsible for managing the graphical user interface elements. And, the last one, *Feedback Module*, is in charge of managing the user's rewards and the game feedback.

6.2 Communication layer

The main purpose of the communication layer is to send the data to the system correctly. It is the lowest layer in the system and was the first to be developed. It contains two different modules that will be explained next: **Piano input module** and **Network module**.

6.2.1 Piano Input Module

This module is in charge of receiving the piano data for later delivery. As can be seen in the figure 6.1 it contains two main elements: *the digital piano* and the *information piano input*.

The digital piano

The digital piano will operate as the game controller of the system. The digital piano must have a MIDI output (see figure 6.3a). in order to send the information from the piano to a computer it is required to use a MIDI USB converter like the one shown in the figure 6.3b. In order to connect the midi converter correctly, the input of the cable must be plugged into the output of the piano.

Information piano input

It is the interface in charge of collecting the information provided by the USB MIDI converter for its later sending. For this purpose, a software called *amidi*[1] has been used. It is Unix a command-line utility which allows to receive and send data from or to external MIDI devices. Once the piano is on, and the USB converter is connected to the computer, **amidi**

[1] linux.die.net/man/1/amidi

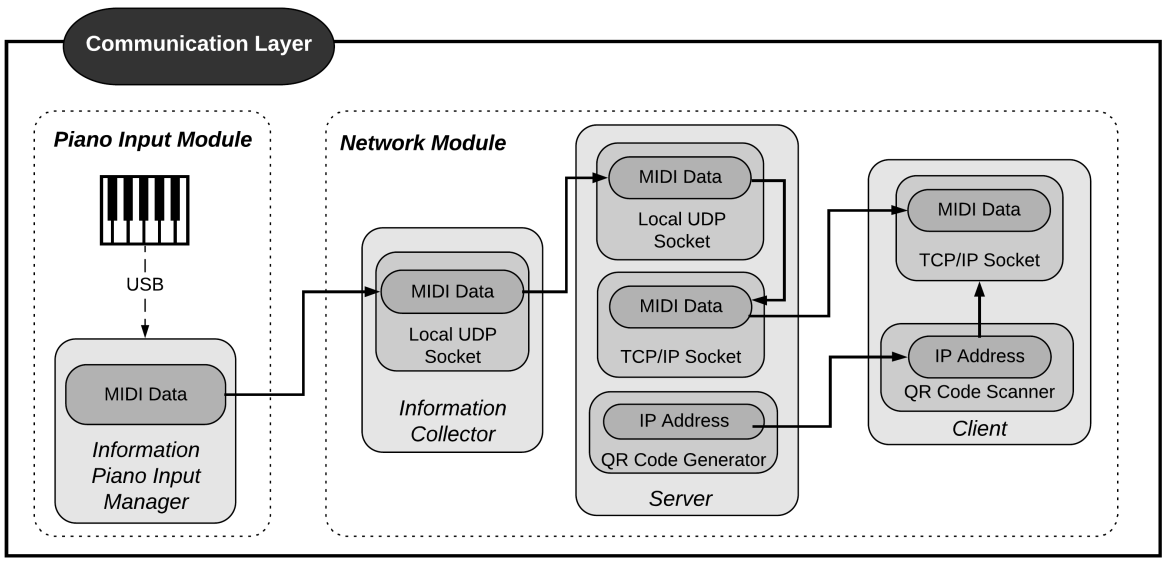


Figure 6.2: Communication layer diagram overview

a) Piano MIDI output   b) USB MIDI Converter

Figure 6.3: Piano MIDI details

has to be commanded to collect the data and print it on the screen. There are two *amidi* options needed for this purpose:

- *-p, –port=name*. This option selects the desired port to transmit or receive data. By default, the name used is "hw:1,0,0", which indicates audio channel of the the USB connected to any port.
- *-d, –dump*. It prints data received from the MIDI port as hexadecimal bytes.

As a result, the following command have to be entered in the console. Later, when a key of the digital piano is pressed, the console display will show the hexadecimal value of the MIDI data produced on the keyboard (see figure 4.30).

```
amidi --port="hw:1,0,0" -d
```

Listado 6.1: Command used for getting MIDI data

6.2.2 Network Module

This module is in charge of handling the entire communications system so that the data obtained in the piano input module reache its destination correctly. As shown in figure 6.1, it is divided into 3 parts or subsystems: *the information collector, the server and the client*.

The network part is very important in this system. It is needed a low delay and without losing network packages between the Hololens and the piano. Moreover, in order to develop a user friendly system, it has been tried to minimise the user's intervention in the communication setup process.

For sending the data from the piano to the HoloLens, a device as communication bridge, called Raspberry Pi 3 Model B, has been used. It is an economical single-board computer with a Linux distribution Operative System (OS) called Raspbian. It has Wi-Fi 802.11 b/g/n, Bluetooth 4.1, Ethernet socket and 4 USBs which are enough for achieving the communication. On the other hand, the connectivity technologies that Hololens can use are Wi-Fi 802.11ac and Bluetooth 4.1 Bluetooth Low Energy (BLE). Therefore, there have been considered two main alternatives for communication: Bluetooth and Wi-Fi, which will be discussed later.

Information collector

This is the part in charge of collecting the data that *amadi* generates and send them to the server. As explained before, the Raspberry Pi receives data from the piano and sends it to the HoloLens. Which means the server will be running on **Raspberry Pi**. As a consequence, the objective is to pass the *amidi* data to the local server. The tool that has been used for this purpose is *netcat*[2]. It is a Linux utility is used for create and manage Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) connections. Thanks of pipes, the output data from *amidi* can be sent to any port by using the *netcat*. As it is a unidirectional and local data transmission, it has been decided to use UDP ports for the communication. This is how *netcat* sends the information to the UDP port of the server.

```
amidi --port="hw:1,0,0" -d | nc localhost $LOCAL_PORT
```

Listado 6.2: Sending amidi data through pipes to netcat and the to the port of the server

Server

The server is the one in charge of connecting with the HoloLens and sending the information of the piano that it receives. It has been developed in *Python* programming language. As mentioned before, the server runs on the **Raspberry Pi**. For development in the raspberry it has been used *Secure Shell (*SSH*)* protocol between PC and Raspberry, which allows to remotely control the raspberry, and *git* for working with the code from the repository[3]. The server is composed of several sub-modules:

- **Local socket UDP.** Receives the piano data from a UDP localhost port.
- **TCP/IP socket.** Listens to establish a TCP connection for later sending the data stream to the HoloLens.
- **QR code generator.** Generates a qr that contains the IP of the interface connection of the server and display it on the screen for the later analysis by the HoloLens.

Finally, it will be explained the improvements made in the server subsystem once the previous tasks were completed correctly. The improvements were as follows:

- **Processes automation.** In order to simplify the initialisation the server for the user, the processes were automatised.

[2] linux.die.net/man/1/nc
[3] github.com/DiegoMolero/RaspberryPianoServer

- **Graphical server interface.** This is a simple GUI for displaying the server information in a more user-friendly way.

Local socket UDP The User Datagram Protocol (UDP) is a communication protocol belongs to the Transport Layer, layer 4, in the Open System Interconnection (OSI) 7-Layer Model[Pos80]. It provides package communication. Contrary to TCP, UDP is not reliable and does not need to setup connections before sending. This means that UDP does not control the data flow in the communication. UDP packages travel from a source port to a destiny port. UDP is encapsulated into IP datagrams. Hence, in order to send data, it is needed to know the port and the IP of the destiny device. In this case the IP is local, so it is "127.0.0.1" also called localhost. Through the python *socket* library, a UDP listening socket is created on the port that receives the server at initialisation. When it receives a UDP message it processes it, removing the line breaks, and sends it to the server process in charge of the TCP connection (see the listing 6.3). If the TCP connection has not yet been established, the server rejects the message.

```
def startUDP(port):
    print("Starting UDP Local Server, port:"+port)
    sock = socket(AF_INET, SOCK_STREAM)
    sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
    sock.bind(('localhost', int(port)))
    sock.listen(0)
    client, addr = sock.accept()
    while 1:
        data_piano = client.recv(BUFFER_SIZE)
        data_send = data_piano.replace('\n','') # Removes the first line break
        print("Local data piano recieved:"+data_send)
        sendData(data_send+'\n') # The line break signal the final of the message
```

Listado 6.3: Opening UDP socket for connecting with amidi

TCP/IP Socket This sub-module, as mentioned before, is in charge of establishing the connection with the main system, the HoloLens, and sending you the information provided by the previous function, UDP socket. Since the main system must not lose the connection at any time with the piano, the communication requirements between the HoloLens and the server are to achieve fast, controlled and without loss of packets communication. Although it was finally decided to use TCP sockets for communication, several alternatives were considered before: **Bluetooth** and **TCP or UDP** connection.

- **Bluetooth.** It is a low range wireless link technology created for substituting the cable communication between computers and electronic portable devices [KHM08]. It can

send and receive synchronous and asynchronous data. Besides, it does not required to be connected to the Internet and it is able to provide an user-friendly configuration setup. Hence, according to the communication requirements of this project, Bluetooth looks a good solution.
In particular, Bluetooth Low Energy (BLE) is the communication technology supported by the Hololens. BLE has been developed for reducing the power consumption of wireless communication on mobile devices. The BLE protocol stack is similar than in classic Bluetooth. In BLE there are two roles, the advertisers and the scanners. Advertisers are the devices that send broadcast data in advertising packets which contain information about the channel for packet transmission. The scanner device listens continuously to any advisement and sends a connection request message to the device when a advertising packet is received[GOP12]. It also possible to achieve bidirectional data communication between both roles with Point-to-Point Protocol (PPP).
The device which starts sending advertisements is the master and can connect with multiples devices for sending data. In this project this role would be played by the raspberry which would send the key signal produced by the piano. On the other hand, the other device scans and tries to find an advertisement, if it does not find any, it goes to sleep until the next period of time and scan again. This function would be done by the Hololens.

Although BLE looks a good solution for this context, it was rejected due to these reasons:

- **Lack of experience and knowledge in the field of Bluetooth communication:** Despite the fact that this is not a technical problem, the lack of experience with BLE would have a strong impact on the planning schedule thus, producing delays in the development of the project.
- **Shortage of documentation and information about implementation of BLE:** Due to the Hololens is an early technology there are not many documentation and examples of BLE implementations yet.

- **TCP or UDP**. Another solution was to implement a Wi-Fi communication between the devices. This alternative has more impact on the system and on the user because both devices are needed to be connected previously to the same network or to the Internet. However, this architecture provides more different solutions through various communication protocols. Specifically, TCP and UDP were the alternatives studied for this solution.

  The implementation of UDP for the project has advantages and disadvantages. Due to it is not a reliable protocol by default, it is needed to designing a mechanism to verify if the Raspberry or the Hololens are available for receiving or sending data and control the data flow. Besides, the system would have to control if some data has been lost. However, due toUDP is not reliable, the communication could be faster than TCP.

Transmission Control Protocol (TCP) is a bidirectional unicast connection-oriented protocol that provides reliable, ordered and error-checked communication through byte data stream [FS11]. This means that, once two devices establish a TCP connection, it guarantees that each package sent will arrive correctly to the destiny. TCP belongs to the Transport Layer, layer 4, in the Open System Interconnection (OSI) 7-Layer Model. It is encapsulated in IP datagrams, IPv4 or IPv6, which refers to the Network Layer. Like in UDP, it is needed to know the IP address and the port of the target device.

According to this system, **it is preferable to use a reliable connection because none package should be lost**, otherwise, the input given by the piano keys will not correspond with the behaviour of the system. Moreover, if the connection is lost the program should detect it and pause or stop the execution until the connection restarts it.

Finally, it had been decided to implement the TCP protocol because it is closer than UDP to achieve the network project requirements. Besides, After performing some tests with the piano and the Hololens emulator with TCP, the messages obtained from the piano were fast enough to have a good playing experience. The code below (see listing 6.4) shows the function that opens the TCP socket, and listens to the connection with the HoloLens. Once connected, the *sendData()* function is called from the UDP process and uses the *TCP* socket created to send the piano data.

```
def sendData(data_piano):
    if 'conn' in globals():
        conn.send(data_piano.encode())  # echo
        print('Data sent: '+data_piano)
def setupTCP():
    s = socket(AF_INET, SOCK_STREAM)
    s.bind(("", TCP_PORT))
    s.listen(1)
    global conn,base
    conn, addr = s.accept()
    serverTCP_Connected()
    while 1:
        base = conn.recv(BUFFER_SIZE)
        if not base: break
```

Listado 6.4: TCP socket waiting for connection

QR code generator The QR function is used for displaying a QR Code with the IP of the server. Later, the HoloLens analyses the IP encoded into the QR and connects with the server.

This solution arises from the problem that the IP of the server that provides the data will not be static. In other words, it will change depending on the network to which it is connected. Therefore, the HoloLens will need to know the IP of the server before connecting to it. One of the most important requirements is that the user should be able to connect to the server in an easy way, minimising his interaction with the system. Therefore, the worst case possible for this solution would be that the user would enter the IP numbers through a virtual keyboard in the HoloLens. From here, two better solutions arise: **IP Discovery** and **QR code**.

- **IP Discovery.** This method consists on sending broadcast UDP messages to all the devices listening to one well-known port. The sender is the Raspberry. The message would contain a private key which both devices know plus the IP address of the server. Once, the broadcast information is received by the other device, in this case the Hololens, it decodes the IP and starts the communication.
  The process was tested, first, between Python scripts, one client and one server, and then, between a Python script and the Hololens emulator. The result between Python was successful. Nevertheless, it was not for the Hololens. Through *Wireshark*[4] it was concluded that the broadcast message with the IP information uses UDP protocol but encapsulated in ADwin configuration protocol. Computers usually allow this traffic but Hololens firewalls reject it. In conclusion, this solution was discarded because it is not possible and secure to modify the firewalls of the Hololens.
- **QR Code.** It allows to encode data in a 2-D graphical representation of a square with symbol patterns [Inc11] (see chapter 4 for more information). This symbol can be decoded captured by a digital camera and decoded later. Hence, this technology provides a good solution for sending the IP address of the server to the Hololens.

  As a result, the server located on the Raspberry Pi obtains the IP address of the network that it is connected. It could be connected with Ethernet or WiFi, but not both at the same time. Once the IP address is known, through the libraries *pyqrcode*[5] and *pypng*[6], it has been used created a function that generate the QR code, store it in the local folder and displays the QR code image on the screen (see listing 6.5).

[4] www.wiki.wireshark.org/
[5] www.pypi.python.org/pypi/PyQRCode
[6] www.pypi.python.org/pypi/pypng

```
def drawQR(code):
    url = pyqrcode.create(code)
    url.png('qr.png', scale=6, module_color=[0, 0, 0, 128], background=[0xff, 0xff, 0xff])
    img = PhotoImage(file="qr.png")
```

Listado 6.5: Generating a QR code

Processes automation In order to automatise the execution of the server communication processes it has been created a bash script. The purpose of this is to make it easier for the user to start the server. The following listing 6.6 shows the most important part where piano data are processed. To start the server it is only necessary to run the *.sh* script. For assigning a port for local communication it has been used the tool *shuf*, which gives a random number. The *hostname* [7] is the program that is used to display the current host, domain or node name of the system. Through the *awk*[8] tool it allows to split the necessary information, in this case the IP, and is stored in a variable. Afterwards the server is launched, and next, *amidi*, which its working was explained in the previous sections.

```
#Assing a random number from 2000 to 2999 to the variable LOCAL_PORT
LOCAL_PORT="\$(shuf -i 2000-2999 -n 1)"
#Taking IP Adrress for QR Code and assing it to the variable IP_ADDR
IP_ADDR="$(hostname -I | awk '{print $1}')"
#Run the pyhton main program that revieves the piano data and communicates with the Hololens
python main.py $IP_ADDR $LOCAL_PORT &
#Run amidi listening to the port 1 and pass the input through port $LOCAL_PORT by netcat
amidi --port="hw:1,0,0" -d | nc localhost $LOCAL_PORT &
```

Listado 6.6: Atumatization of the processes in the server

Graphical server interface This system is aimed to non-expert users, who are not generally comfortable with the traditional console. For this reason, having a graphical interface on the server could be an important part to improve the user experience. With this objective, it has been designed, through the Python module called *tkinter*, a GUI that represents the information in a organised and well-looking way in the server. The GUI shows the QR Code (see *figure 6.4*) which will be used to connect the HoloLens. Besides, it displays more information about the status of the connection. The size of the windows has been designed to fit with the Liquid Crystal Display (LCD) display of the raspberry which has a 3-inches screen size with a resolution of 320x480 .

[7] linux.die.net/man/1/hostname
[8] /www.tutorialspoint.com/awk

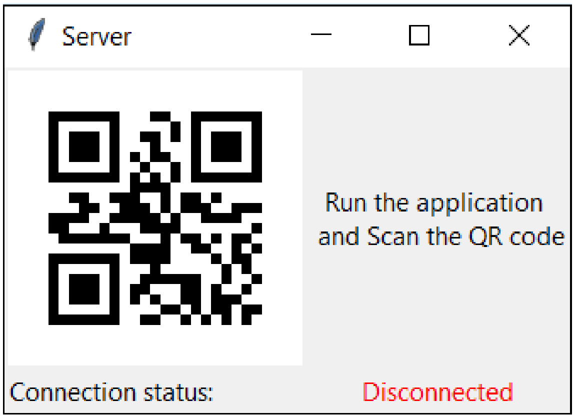


Figure 6.4: GUI of the server

Client

The role of the client belongs to the main system, the HoloLens, which will connect to the server via the TCP protocol to receive the MIDI data that the piano outputs. As shown in figure 6.1 the customer is divided into two subsystems:

- **QR Code Scanner.** It is in charge of accessing the camera of the HoloLens, scanning the QR code and decode it.
- **TCP/IP Socket.** It connects to the TCP server through the IP obtained previously by the QR scanner to receive the piano data.

QR Code Scanner. As explained above, in order to connect the HoloLens to the server it is necessary to know the IP in advance as it is dynamic and can change. With the aim of minimising the user interaction in this process, it has been decided to use a QR code. Therefore the objective of this sub-system is to analyse this code. For this purpose, it has been used a plugin which uses *ZXing*[9] adapted to the HoloLens. *ZXing* is an open-source, multi-format 1D/2D barcode image procesing library. Mike Taulty adapted the *ZXing .NET* version to the HoloLens for for an easier development [Tau16].

This plug-in works as follows. First, the camera frames are captured in real time. these frames are analysed by the *ZXing* library. In case the frame contains a qr code, it is decoded. The decoded message is saved and sent to the TCP module in order to start the connection with the server. When it is necessary to scan the QR code, the system displays a sample model showing the steps to be followed in front of the user (see figure 6.5).

The code in listing 6.7 shows how this process is done. If the scan result is null the scan is performed again until a result is found. It is important to explain that this function can not be executed in the unity editor. For this reason the conditional compiler is used, as explained

[9] `www.nuget.org/packages/ZXing.Net`

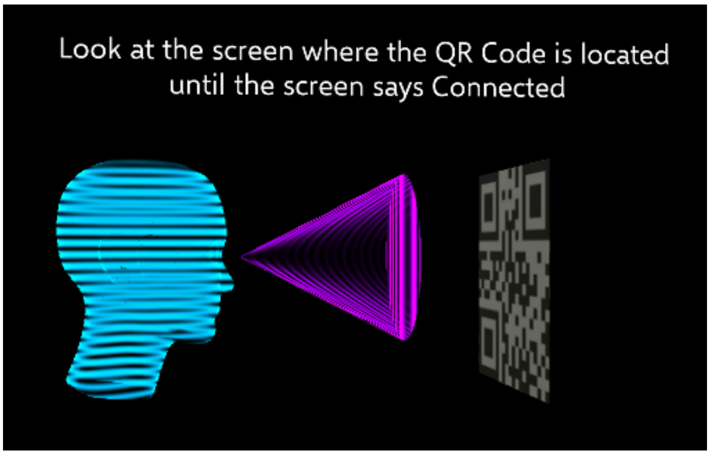


Figure 6.5: Instructive model for QR code scanning

in chapter 4, in order to display it without errors in the HoloLens.

```
  public void OnScan(){
#if !UNITY_EDITOR
    MediaFrameQrProcessing.Wrappers.ZXingQrCodeScanner.ScanFirstCameraForQrCode(
        result =>{
          UnityEngine.WSA.Application.InvokeOnAppThread(() =>{
            if(result == null) OnScan();
            else{
                AppManager.IpAddress = result;
                AppManager.NextState();
            }
          }, false);
        },TimeSpan.FromSeconds(30));
#endif
    }
```

Listado 6.7: QR code scanning process

TCP/IP socket client. Finally, once the qr code IP has been decoded correctly, the HoloLens connect to the server to receive the data. The API used for networking communication in the Hololens is *Windows.Networking*[10] which refers to UWP. It provides to create TCP sockets for sending and receiving data in an asynchronous way. By default the port used for the connection is 8000. In the following code you can see a simplification of how the data is received by the client. In the following listing 6.8 it can be observed a simplification of how the client connect to the server and how the data is received.

In figure 6.6 it can be observed an Unified Modeling Language (UML) diagram with the

[10] docs.microsoft.com/en-us/uwp/api/windows.networking

relationship between the classes of the client side. The *StageManager* class is in charge of managing all the states the system passes through and will be explained in the gamification layer section. The *Placeholder* detects and decodes the QR Code in oder to get the IP address. The *TPCommunication* establishes communication with the server.

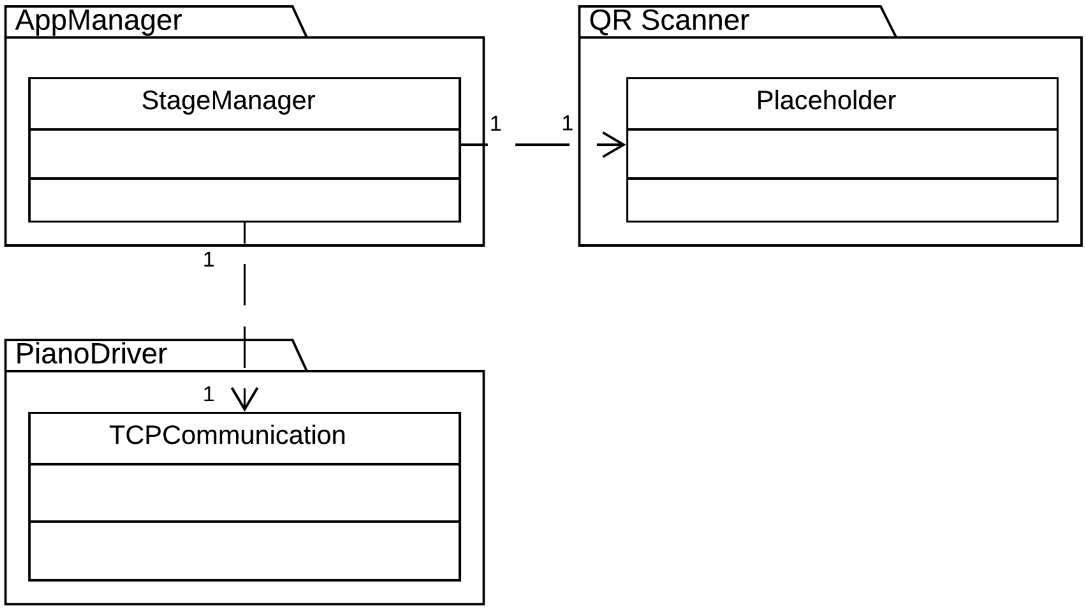


Figure 6.6: Communication layer class diagram. Client side

```
private async void TCPclient(){
    _socket = new StreamSocket();
    Debug.Log("Connecting to "+Host+":"+Port);
    HostName serverHost = new HostName(Host);
    await _socket.ConnectAsync(serverHost, Port);
    _connection = true; //Connection Suscess
    Stream streamIn = _socket.InputStream.AsStreamForRead();
    StreamReader reader = new StreamReader(streamIn);
    await reader.ReadLineAsync();
    while (_connection == true)
    {
            string response = await reader.ReadLineAsync();
    }
}
```

Listado 6.8: TCP client connection process

Communication layer summary

To conclude with the explanation of the *communication layer*, a case of use has been created schematised in a sequence diagram (see figure 6.7). In this diagram it can be seen the 3 systems that constitute the architecture, the digital piano, the raspberry pi and the HoloLens. First, the server is launched. When this happens, *amidi* starts listening to the piano data, the UDP ports are opened, the IP code is generated with the acIP address to which the raspberry is connected and then, the socket UDP is created and starts to listen for connections. Once the server is started, the user plays a piano key. The keyboard generates the midi data with the key information and sends them through the MIDI converter to the Raspberry Pi. *Amidi* receives the data and passes it to the server via netcat through an UDP socket. When this data arrives to the server, as it has not yet established a TCP connection with the main system, it discards the data.

On the other hand, the HoloLens load the application and the system starts the qr code detection process. When the user looks at the server GUI Raspberry Pi which displays the QR code of the server address, the HoloLens detects the QR and decodes it. Once decoded, it establishes a TCP connection with the server. Finally, when the user presses the key the information passes through the server and reaches the HoloLens.

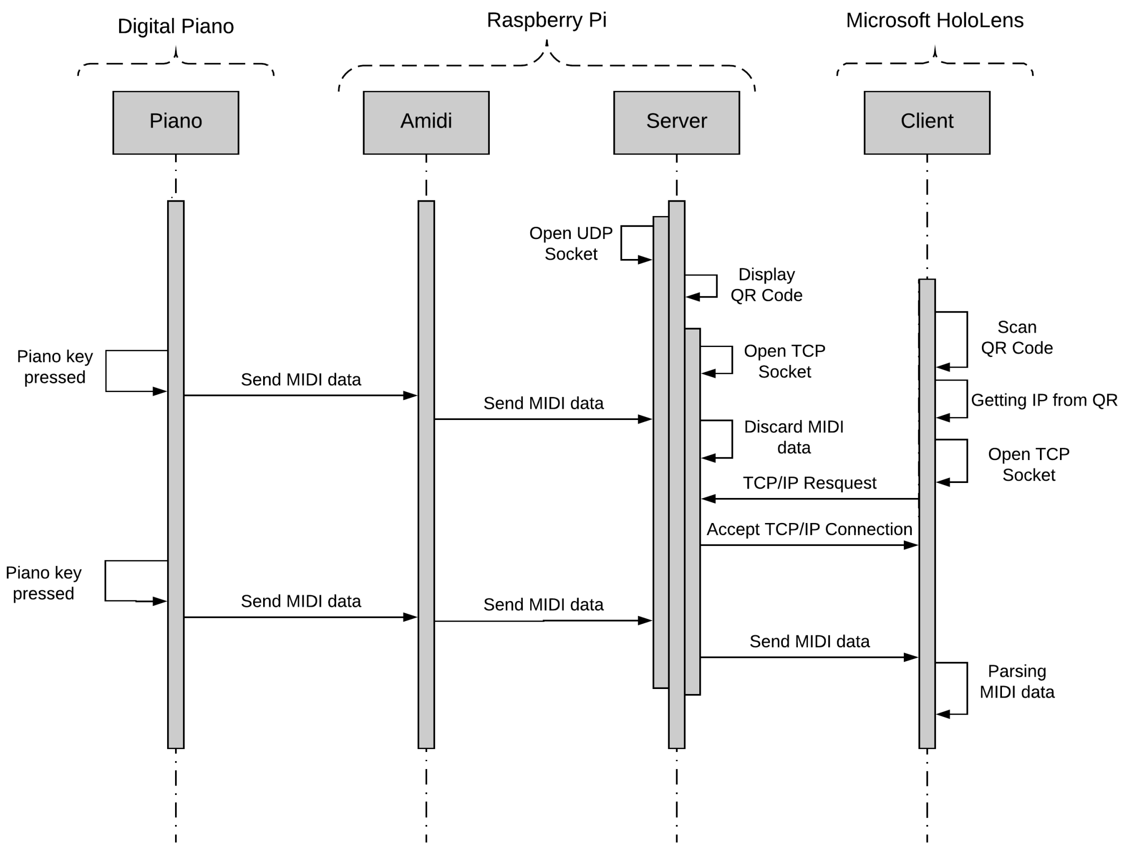


Figure 6.7: Communication layer sequence diagram

6.3 Tracking and user interface layer

This layer is located in the centre of the system and is in charge of abstracting the functions of the HoloLens, separating the part that is more focused with the HoloLens from the apper layer. In other words, if a developer were in the top layer he/she would not need to know how the HoloLens work, because in this layer they have been provided with the tools to program without that knowledge. In particular, there are two modules that provide this obstraction:

- **Piano driver module.** It processes the data provided by the communication layer transforming the midi format into information about the keys pressed in the system. Then, it triggers events with the information it has obtained.

- **Tracking module.** It detects the piano keys to map their position and thus display all the elements of the game in reference to this position. This is achieved thanks to the Vuforia SDK marker recognition.

6.3.1 Piano driver module

This module is the one responsible for converting the raw data coming from the digital piano into useful information for the system. These parsed data correspond with each different piano key actions.

A digital piano keyboard can change mainly among other models in the number of keys that it has. However, all the pianos have at least the keys corresponding to the octave number 4. In order to cover the largest number of digital pianos with MIDI output it has been taken only the keys of the centre, which are also the most used in music. Hence, the system will process the keys from the central note C, called C4, to his octave, C5, which is a total of 12 keys (see *figure 6.9*). Although it is using only 12 keys, the system has been designed with scalability in mind. The impact of adding new input keys in the system is very low.

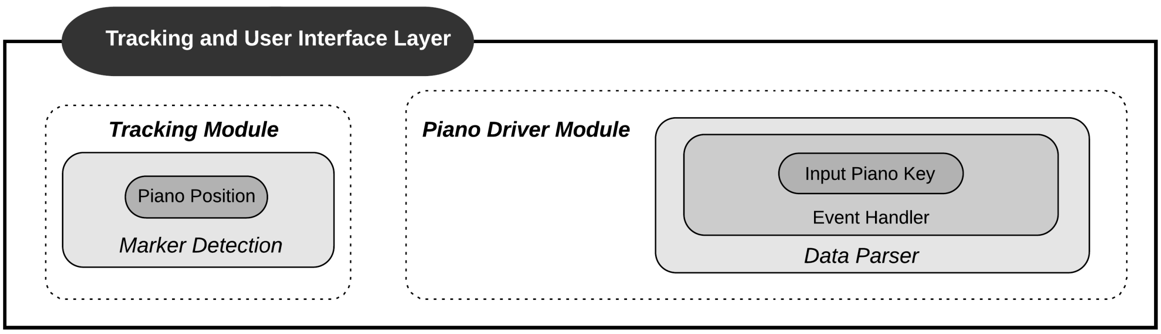


Figure 6.8: Tracking and user interface layer diagram overview

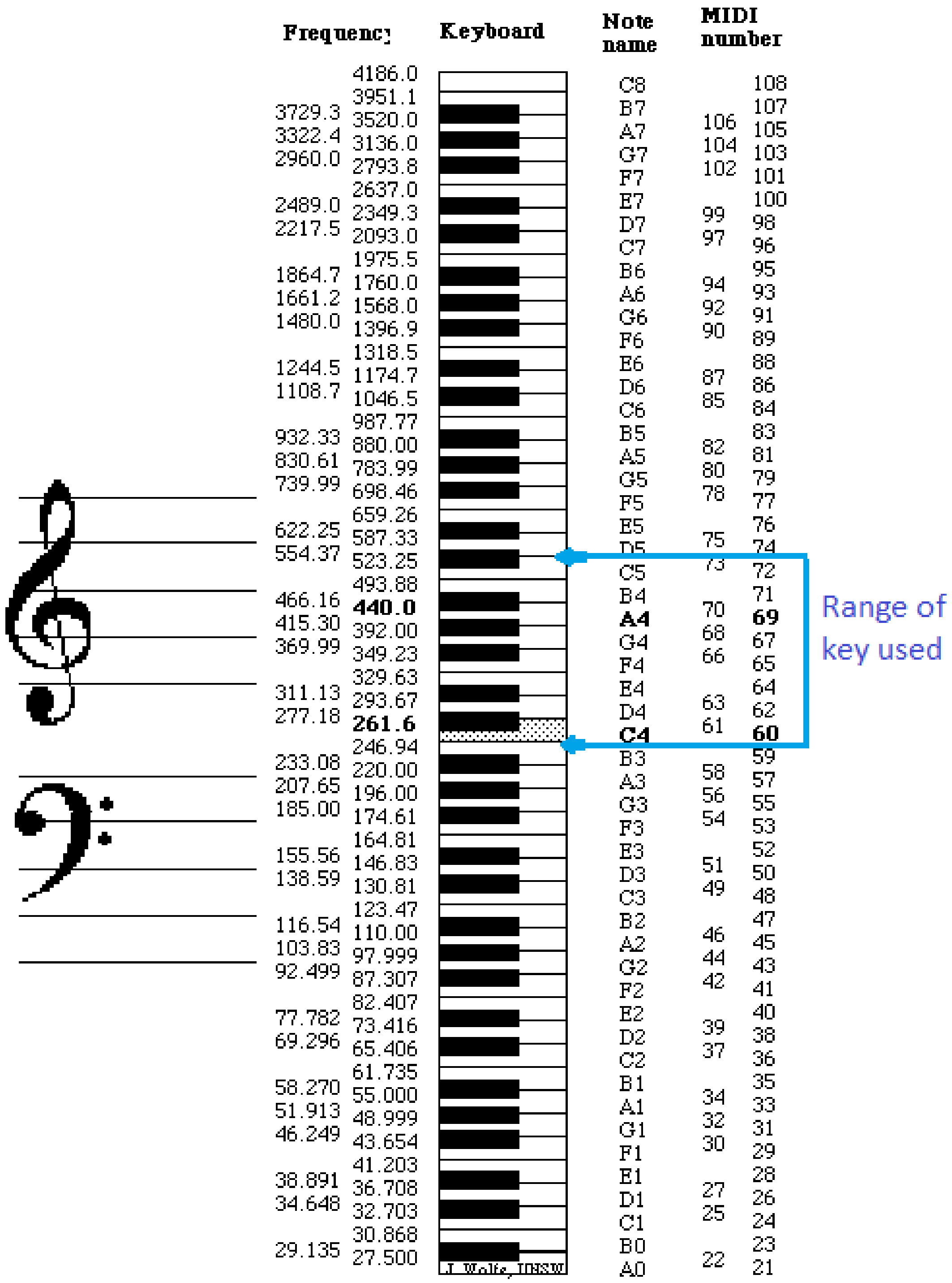


Figure 6.9: Piano key scheme with the MIDI digit belonged to each key

Data parser

How does the data parsing work? Previously, a list with the key IDs has been established in which each note name is associated with its hexadecimal representation in the keyboard (see the figure 6.9). When a data stream is received by the socket TCP it sends the information to the *PianoDriver*, which is responsible for processing the information. Due to the problematic

Note	**Raw Piano Data**	**Data Processed**
C4	Press: 90 3C 7F Release: 90 3C 00	Press: 60 / True Release:60 / False
C4#	Press: 90 3D 7F Release: 90 3D 00	Press: 61 / True Release:61 / False
D4	Press: 90 3E 7F Release: 90 3E 00	Press: 62 / True Release:62 / False
D4#	Press: 90 3F 7F Release: 90 3F 00	Press: 63 / True Release:63 / False
E4	Press: 90 40 7F Release: 90 40 00	Press: 64 / True Release:64 / False
F4	Press: 90 41 7F Release: 90 41 00	Press: 65 / True Release:65 / False
F4#	Press: 90 42 7F Release: 90 42 00	Press: 66 / True Release:66 / False
G4	Press: 90 43 7F Release: 90 43 00	Press: 67 / True Release:67 / False
G4#	Press: 90 44 7F Release: 90 44 00	Press: 68 / True Release:68 / False
A4	Press: 90 45 7F Release: 90 45 00	Press: 69 / True Release:69 / False
A4#	Press: 90 46 7F Release: 90 46 00	Press: 70 / True Release:70 / False
B4	Press: 90 47 7F Release: 90 47 00	Press: 71 / True Release:71 / False
C5	Press: 90 48 7F Release: 90 48 00	Press: 72 / True Release:72 / False

Table 6.1: Representation of the raw data obtained from the piano processed by the system

of UWP aforementioned in the chapter 4, it has been created an script that emulates the raw data produced by the piano and received through the network. This script is only executed with the Unity editor with the propose of achieving faster developing and testing of the system. In order to emulate a piano in the most similar way for the programmer, it has been used the keys of the keyboard as piano keys (see figure 6.10).

The piano used for the development gives the following data (see table 6.1). The raw piano data has been previously analysed with the program *amadi*[11]. It is represented in Hexadecimal and is divided in 3 parts: the status, the key and the velocity (see chapter 4). Therefore the *PianoDriver* divides the data stream into three and analyses it. First, it verifies if the first byte refers to the state of pressing keys, which is 9n (see more information in the chapter4). Then, it checks next byte value, which corresponds to the note of the piano keyboard and store it. Then it analyses the last byte that shows if the key has been pressed or released. And finally, it invokes an event with the note id and the action performed, pressed

[11] `linux.die.net/man/1/amidi`

or released.

```
public void RecievePianoData(string data)
{
    string[]  data_fragmented = data.Split(default(string[]), StringSplitOptions.
        RemoveEmptyEntries);
    if (int.Parse(data_fragmented[0], System.Globalization.NumberStyles.HexNumber) !=90)
        return;
    key_value = int.Parse(data_fragmented[1], System.Globalization.NumberStyles.HexNumber);
    if (int.Parse(data_fragmented[2], System.Globalization.NumberStyles.HexNumber) == 0){
        aux_activate = false;
    }else{
        aux_activate = true;
    }
    pianoEvent.Invoke((KeyNote)key_value,aux_activate);
}
```

Listado 6.9: Parsing piano raw data

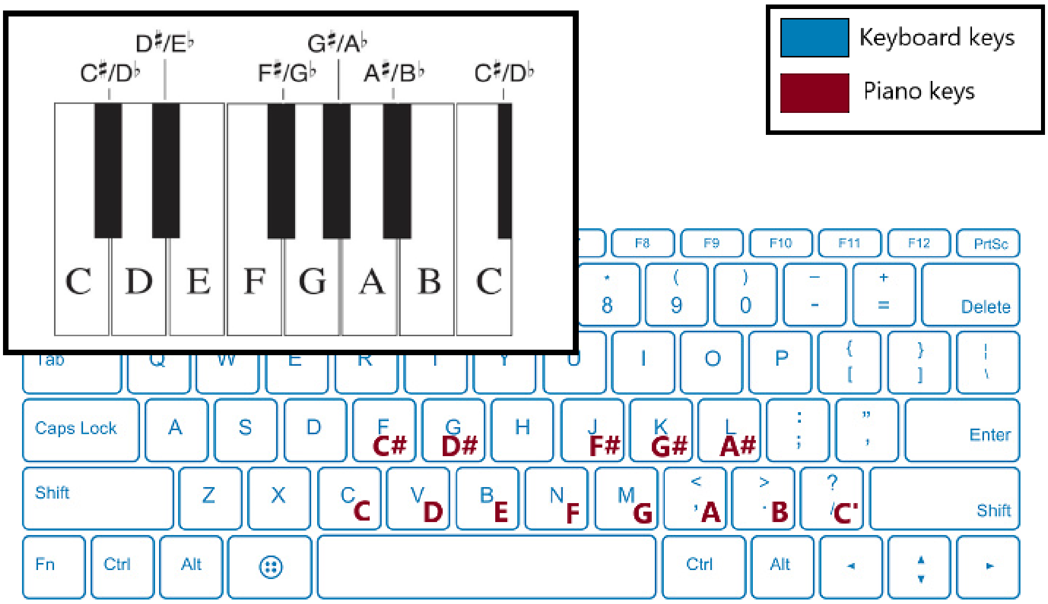


Figure 6.10: Representation of the keyboard compared to the keys of a piano and its notes

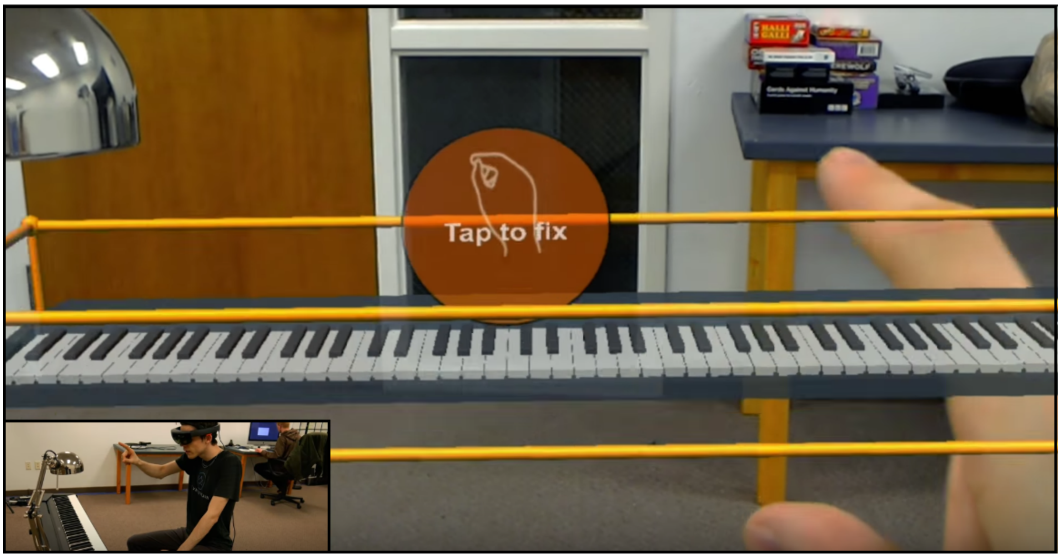


Figure 6.11: Sample of how *MusicEverywhere* maps the piano position

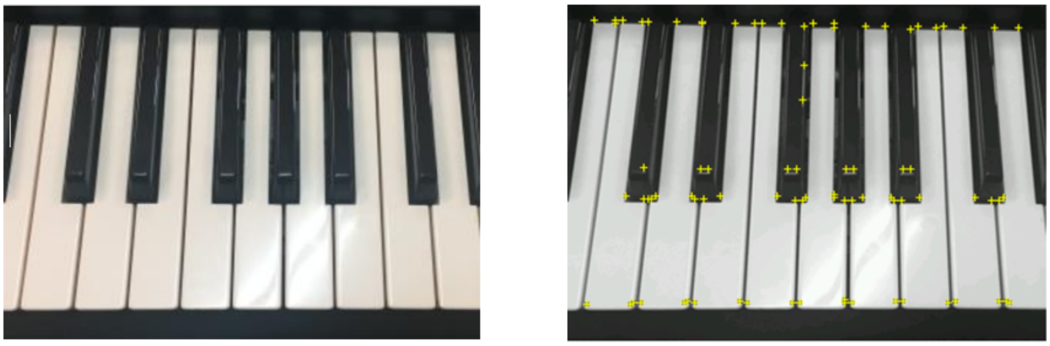
6.3.2 Tracking module

Once the Hololens has received and parsed the piano data, it is needed to calculate the position of the real piano in the space. For this task some alternatives can be implemented.

One possibility is placing some markers around the piano like in the figure 4.31. The markers would be analysed with the camera and would map the position automatically. However, although the positioning of the markers is an intuitive method, the piano mapping accuracy depends on the user. This would cause a different experience each time the system is launched. Besides, it is necessary to have one more element for the system, which is the marker.

Another similar alternative consists in placing a standard model of a piano on the real one, helping with spatial mapping capability of the HoloLens like *MusicEverywhere App* does (see *figure 6.11*). This model would have the size of a real piano because the size of the piano keys is standardised and should not change between other different pianos. It is a simpler task than the other solution but it is still tedious and imprecise. Besides, the system can not control if the user has been placed the model correctly.

The alternative solution chosen solves the disadvantages of the others. Thanks to the use of *Vuforia*, the camera of the Hololens can detect and map the piano [Inc18]. This technology verifies in real-time if each frame captured by the camera matches the patterns of the image target that wants to identify. An image target is nothing but a marker with the difference that it can be any image desired. Using the data obtained by the tracking: distance and size, *Vuforia* is able to calculate the position of the real image on the world regarding to the Hololens. Thanks to this, it can ensure that the mapping of the piano will be almost the same for different executions and the position of it will have a good level of accuracy. As a result,



a) Desired image to be recognized b) Processed image for Vuforia recognition

Figure 6.12: Image processing for later recognition with Vuforia

the virtual holograms will be situated just above each piano key.

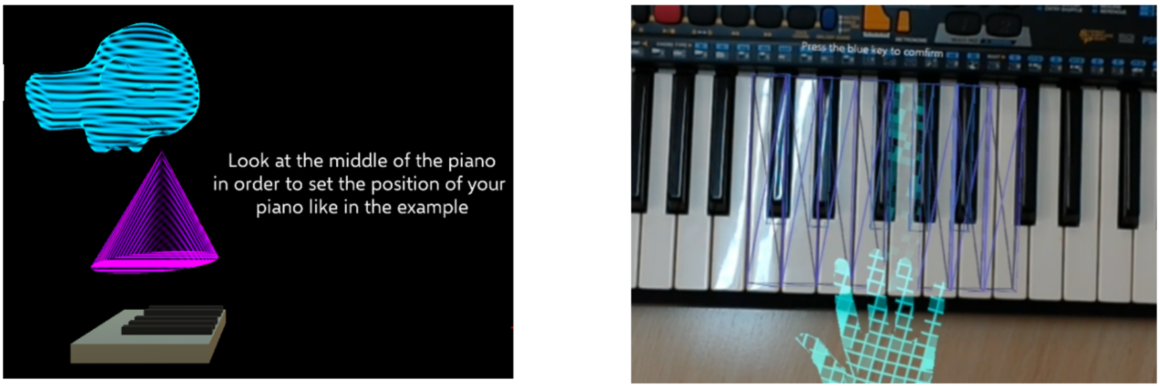
Vuforia marker detection

As explained above, Vuforia allows the recognition of images as markers. In addition to this, it is a SDK optimised for the HoloLens and provides the necessary tools to perform this process easily in Unity. In order to detect an image target it is first necessary to format the image so that it can be recognised by Vuforia. To do this, from the portal of Vuforia it allows us to upload the photo. Once the image is uploaded, the Vuforia portal generates some marks in the image and scores it (see figure 6.12). These small marks are used to recognise the image, the greater the number of marks the easier it will be to recognise it for the system. When Vuforia is running, it captures the video frames in real time, converts them to black and white and compares them to the processed image target.

An important problem to mention is that when Vuforia is working **it is not possible to access the live stream** function of the *device portal* of the HoloLens. The reason for this is that these two processes cannot access the camera at the same time and, as a result, it produces errors in the system. Currently, there are no efficient alternatives that can solve this problem externally.

The image targets created from the Vuforia portal can be downloaded and imported directly into the unity project, or can be accessed remotely via Internet. In this case, the local method has been used. Once the image targets have been imported into the project, a GameObject has to be created with the real size of the image desired to be recognised. If this size is not the optimal one, the reference to the image target will have wrong values. The Vuforia API provides a script that produces an event when the image is detected. By default, when the image is detected, the GameObject children of the image target are instantiated in the scene.

In the case of this system, when the keyboard is detected, the outline of the piano is



a) Instructions to indicate how to scan the piano

b) Information about saving the position of the scanned piano

Figure 6.13: Process of mapping the piano position

displayed over the real piano with some instructions. The instructions indicate that when playing the key it points to, the digital piano position will be saved and the entire system will be displayed there (see figure 6.13b). By combining this system with the data received from the piano, it is possible to map the exact middle position. Because, since all the piano keys are the identical, the key pressed may be one of the end keys of the piano and, therefore, it would not be correctly mapped. In this way, the keys mapped are ensured to belong to the middle of the piano and that the system is working correctly. As in the case of scanning the qr, it also appears in front of the user some instructions to indicate how to map the piano position (see figure 6.13a).

Tracking and user interface layer summary

To conclude with the explanation of section, a class diagram that abstracts the behaviour of this layer in the system has been designed (see figure 6.14). As mentioned before, the *StageManager* is responsible for managing the states of the system. Once the connection has been established correctly through *TCPConnection*, the system starts the process of mapping the piano position. The *DefaultTrackableEventHandler* triggers an event when the image target, in this case the piano image (see figure 6.12), is detected. This event displays an instruction model on the piano that indicates the user which key should be played in order to save the piano position. When the key has been pressed, the *PianoDriver* will create an event, *PianoEventKey*, with the information of the key pressed that will be received by the *ActivatePianoVuforia*. Finally, if the key pressed is the correct one, the piano position will be saved and the main menu of the system will be displayed. The menus will be explained in following section.

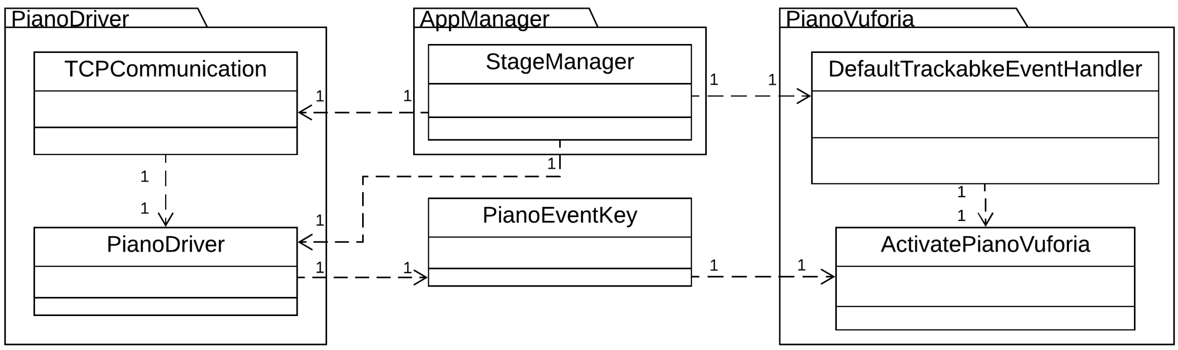


Figure 6.14: Class diagram of the tracking and user interface layer

6.4 Gamification layer

This layer handles all the game logic part of the system. As can be seen in figure 6.15, this layer contains 3 main modules.

- **GUI module.** This module deals with all the aspects related to the user interface, mainly with the menus. It also manages the states that the system goes through.
- **Music reader module.** This module is mainly in charge of reading the music pieces and displaying them in the game. In short, it controls the execution logic of the levels.
- **Feedback module.** It displays the information about how the user is playing. It also provides the gamification aspect, such as achievements or scores.

6.4.1 GUI module

Before starting to explain how this module works in the system, it is necessary to clarify how the *StageManager* works.

As its name indicates, *StageManager* is responsible for managing the states through which the system flows. In order to understand it quickly and easily, a state diagram has been created which abstracts the behaviour of it (see figure 6.16). All the states that the system goes through have already been introduced in the previous sections except the *MenuGame* and *Playing* state which will be explained below.

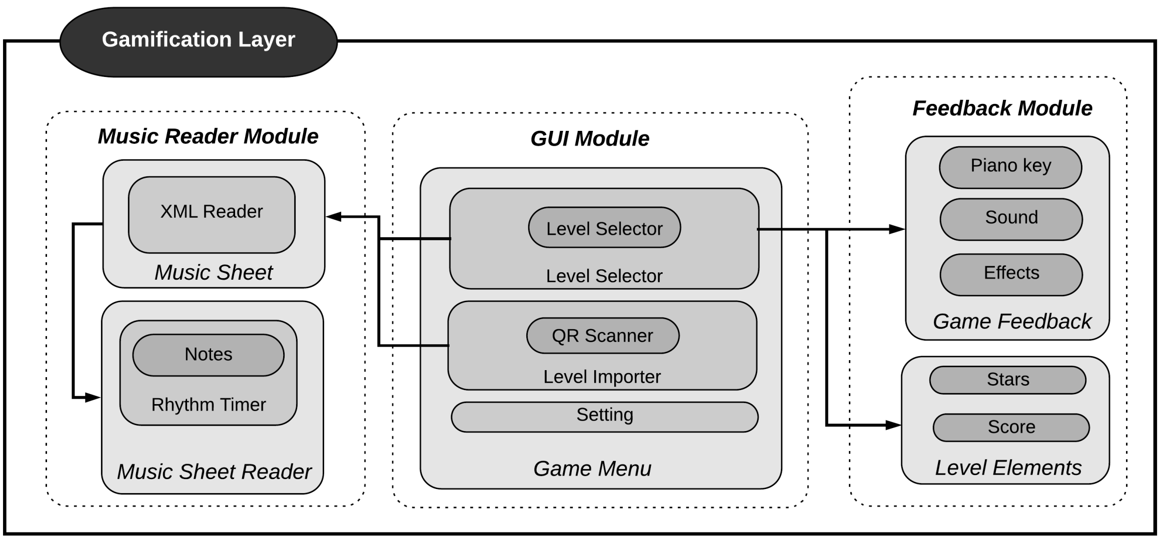


Figure 6.15: Gamification layer overview

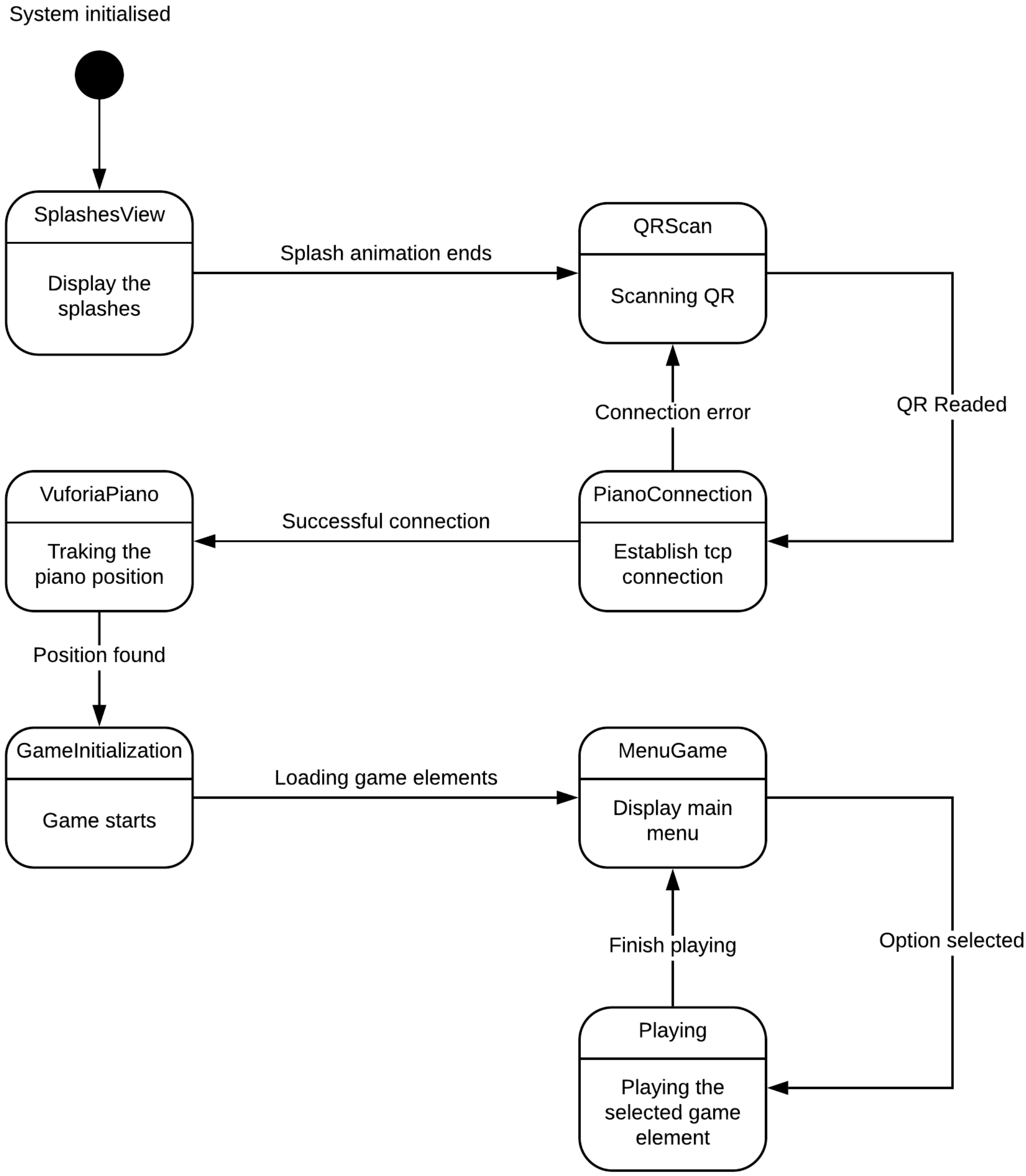


Figure 6.16: A state diagram of the *StageManager* behaviour

A standard GUI in a system like HoloLens usually consists of a 2D window that follows the user view with options. Through using the gaze cursor the user selects these options and airtaps them in order to launch a option. In this system, as the game is developed in relation to the position of the piano and not the whole room, the GUI can be approached differently to get more advantages. In addition, the digital piano input as controller provides simpler interaction with the system. Therefore, the chosen solution consists of displaying bars over the keys with the indicative name of the option linked to that key. As can be seen in figure 6.15, three different menus plus the main menu have been distinguished with different behaviours and separate classes. In the diagram it can be seen how the menu classes are related to each other 6.20.

- **Main menu.** As indicated, it is the main menu. From this, it is possible to access other menus.
- **Menu setting.** From this menu the main system options can be configured.
- **Menu levels.** From this menu it is possible to select the levels that are predefined by the system to play them or consult their statistics.
- **Level import.** From here new levels can be imported and played through the QR scanner.

Main menu

This menu is loaded when accessing the *MainMenu* state as shown in the state diagram of the figure 6.16. Its functionality is simple. All menus work in a similar way. It includes a script that contains the prefabs of the other menus. This script is subscribed to the events of the *PianoDriver* script which provides the piano data. Then, it is specified which key of the piano will load the following menu. In figure 6.17 it can be seen how the main menu looks like in the system. When a key associated with another menu is pressed, the *Playing* state is loaded in the *StageManager* (see figure 6.20).

Setting

This menu looks similar to the main menu (see figure 6.18). Three different options can be made in it.

- *Mute notes or unmute notes.* This option is very useful, since it provides the option to play with the sound of the notes that the system produces artificially through samples or, to use the real sound of the digital piano.
- *Restart score.* It resets the score value of all levels of the game to 0.
- *Return.* It returns to the main menu. For the *StageManager* it would be back to the *MainMenu* state (see figure 6.16).

Figure 6.17: Main menu

Figure 6.18: Setting menu

Level selector

In this menu the predefined levels that can be played are displayed. As it can be observed in the figure 6.19, from the piano it is possible to navigate through the list of levels and consult the information about them, going from left to right. This menu is fully scalable. To add a new level, it is only needed to create a new level sphere based on a prefab and load it in the list of levels. It is also possible to return to the main menu. When it is placed on a level, the song of that level is reproduced in order to add more information about it. The additional information about the levels such as the stars and the score will be discussed in the *Feedback module* section.

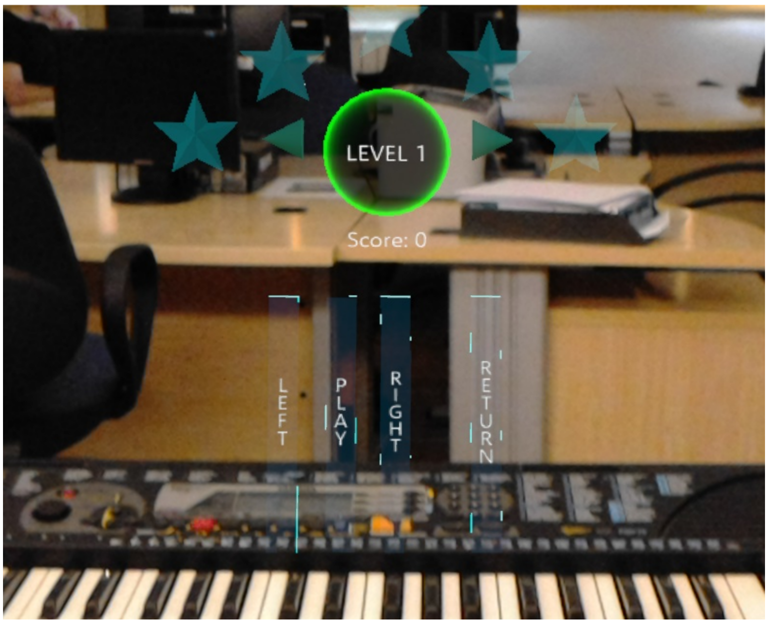


Figure 6.19: Level selector

Level import

One of the features that makes this system very **scalable** is the level import option. As its name suggests, it allows to easily add new levels and play them in the system. Unlike the list of levels, here it is not necessary to recompile the system solution, **the user can add and play the new levels**. Before explaining it how import a level it is going to explain **how to create a level**. In this case, an online web tool called *Flat*[12] has been used. It is an important requirement to have a basic knowledge of music to create a level as a music composition tool will be used. It is also possible to consult the chapter 4 where it is discussed about the music notation system in order to understand how to create a score. Once a composition is finished it has to be exported in Extensible Markup Language (XML) format. Specifically, in the MusicXML 3.0 format. MusicXML[13] is a standard open format to represent digital music sheet. Once the score has been downloaded in MusicXML format, it has to be uploaded to the Internet and then, generated a QR code with the link to the download. For this task, it has been used web tool that is able to to this in a quick and easy way called *Scanfer*[14]. With the these tools the process of uploading and creating a music score is considerably simplified but many alternatives could be used. the class that manages this behavior is called *MenuManagerImport* and can be seen in the diagram of the figure 6.20.

On the other hand, once it has generated the QR with the download link to the music sheet, the HoloLens will perform the following steps:

1. **QR scanner.** As can be observed in figure 6.15, in this step the system also makes use

[12] www.flat.io
[13] www.musicxml.com
[14] scansfer.com/send-or-upload-a-file

of the QR scanner. In the same way as in the previous section the QR is parsed. When the system is ready to scan the QR a message like the one in figure 6.5 is displayed indicating how to scan it.

2. **Download Music XML file.** When the Uniform Resource Locator (URL) has been obtained, it is sent to the *XMLReaderMusic* class, which downloads the file. If the download fails, the system will return to the main menu.
3. **Read XML file.** The *XMLReaderMusic* read and processes the music sheet transforming it into a playable level. How the *XMLReaderMusic* and the level work will be discussed in the following section about the *Music score module*.
4. **Play the level.** After creating the level, the system informs the user that the music sheet has been read correctly and can start to play the level.
5. **Option menu.** when the level is finished, the system displays the score obtained and asks if the player wants to re-play the level, import a new level or return to the main menu.

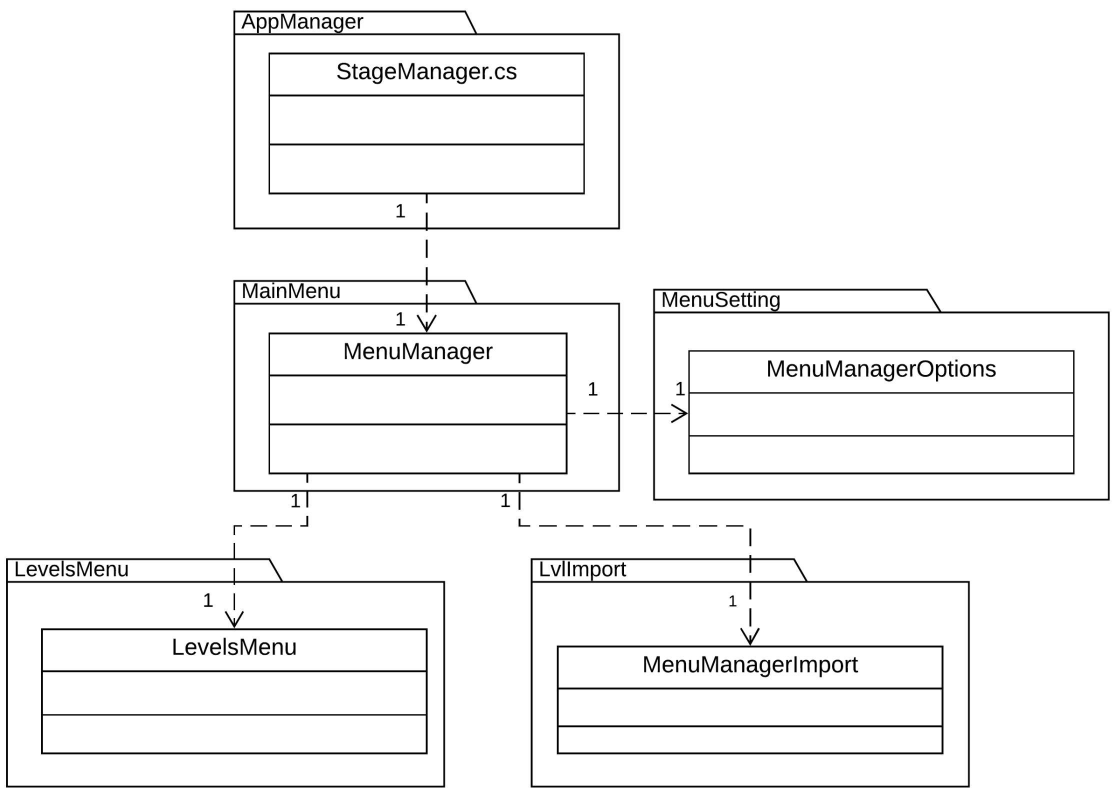


Figure 6.20: Menus class diagram

6.5 Music reader module

This module is used to process the scores and display them in the game. As it can be observed in the figure 6.15, there are two main parts of this module. These parts simulate the process of how a real score would be read and they are part of the *Level*.

- **MusicSheet.** It contains all the information about a song: the notes, the length of the notes, the rhythm, etc.
- **MusicSheetReader.** It is the subsystem responsible for managing the logic of the level execution in the game. Inside it it can be distinguished the element in charge of reading the scores among others.

6.5.1 Music sheet

As explained before, the MusicSheet function is similar to that of a real score. It contains data about the song. Before explaining what contains a music sheet it is necessary to explain the values of a *NoteMusicSheet*.

A *NoteMusicSheet* contains the information relevant to a note (see figure 6.23). These are following values:

- **Note.** It is the pitch of the note. In other words, to which key on the piano this note corresponds.
- **Measure.** The number of the measure to which the note corresponds.
- **Beat.** It indicates to which part of the pulse this note belongs. i.e. in a 4/4 compass a note could belong to one of the 4 parts.
- **PartBeat.** Indicates to which part of the setting the note belongs. Given the previous example of a 4/4 compass pulse, each pulse would have 4 divisions. The note can belong to one of the 4 divisions.
- **Finger.** The finger is additional information about the note, which helps the pianist know which with finger he/she should play the note.

On the other hand, a *MusicSheet* is a container of *NoteMusicSheets* and general information about the music score such as follows:

- **Tempo.** The real measure of the tempo are the Beats Per Minute, which indicate the speed at which the song has to be played.
- **Measure.** The number of the measure to which the note corresponds.
- **TotalMeasures.** It simply indicates the total number of measures.
- **BeatsPerMeasure.** Indicates how many pulses fit into one measure. For example, if this number is 4, it means that the compass used will be a 4/4.
- **Notes.** The list with all the *NoteMusicSheet* that the music sheet contains.

An important function of this class *getNotes()*. It receives as parameters, the measure, the beat and the part of the beat in which the player would be playing the song. As a result, the function returns the notes that contain these values.

To create a score, it is possible to create a GameObject with the *MusicSheet* and manually add the values of each note and music score from the *Unity Editor*. However, since this method is poorly scalable, a more practical solution was designed. This is the XML reader, which was introduced when discussing the import of a level in *XMLMusic* format.

XML music reader

The MusicXML format uses a representation of the notes based on their sequence and duration of time. This means that there is no information about which compass the note belongs to or on which pulse. In order to explain in a graphic way how the XML reader works, a flow diagram has been made (see figure 6.21).

In this system there are two different cases for reading an XML, importing from the internet (which was explained in the previous section) or in the local system folder. In the latter case, it is necessary to enter the name of the XML file so that it can be located and read. A library called *System.XML* has been used to read the XML file where the *XDocument* class is used. This library allows to read theXML node elements of the file.

In order to solve the problem of the different representation of data in XMLMusic format, it has been defined a variable that represents the moment of time in which the notes are placed based on their duration.

At first, the rhythm and tempo values are initialised, to know how many parts have a measure. According to this, the values of *NoteMusicSheet* aforementioned, to which each note belongs, are assigned in a sequential order. As a result, the values of the notes are obtained in the system format and are stored in a *MusicSheet*.

In relation to figure 6.21, there are some elements in the XML that receive a different behaviour for the system:

- **Rest.** Unlike in this system, there are no rests, since it would not make sense to represent a silence. These are represented by the absence of notes. Therefore, when a silence is detected in the XML, its duration is calculated, the compass counter is increased and the next element is called.
- **Alterations.** In this system each note is linked to an exact pitch. An alteration represents a half-step rise in the note it represents (see more information about alterations in the chapter 4). Therefore, when an alteration is found, the final pitch is calculated and the value of it is assigned to the *NoteMusicSheet*.
- **Chords.** When a note in the *MusicXML* has the *chord* tag implies that it is placed in

the same position as the previous note. So, if it is a chord it will have the NoteMusicSheet values of the previous one. Hence, the *Tempo* value is not updated.

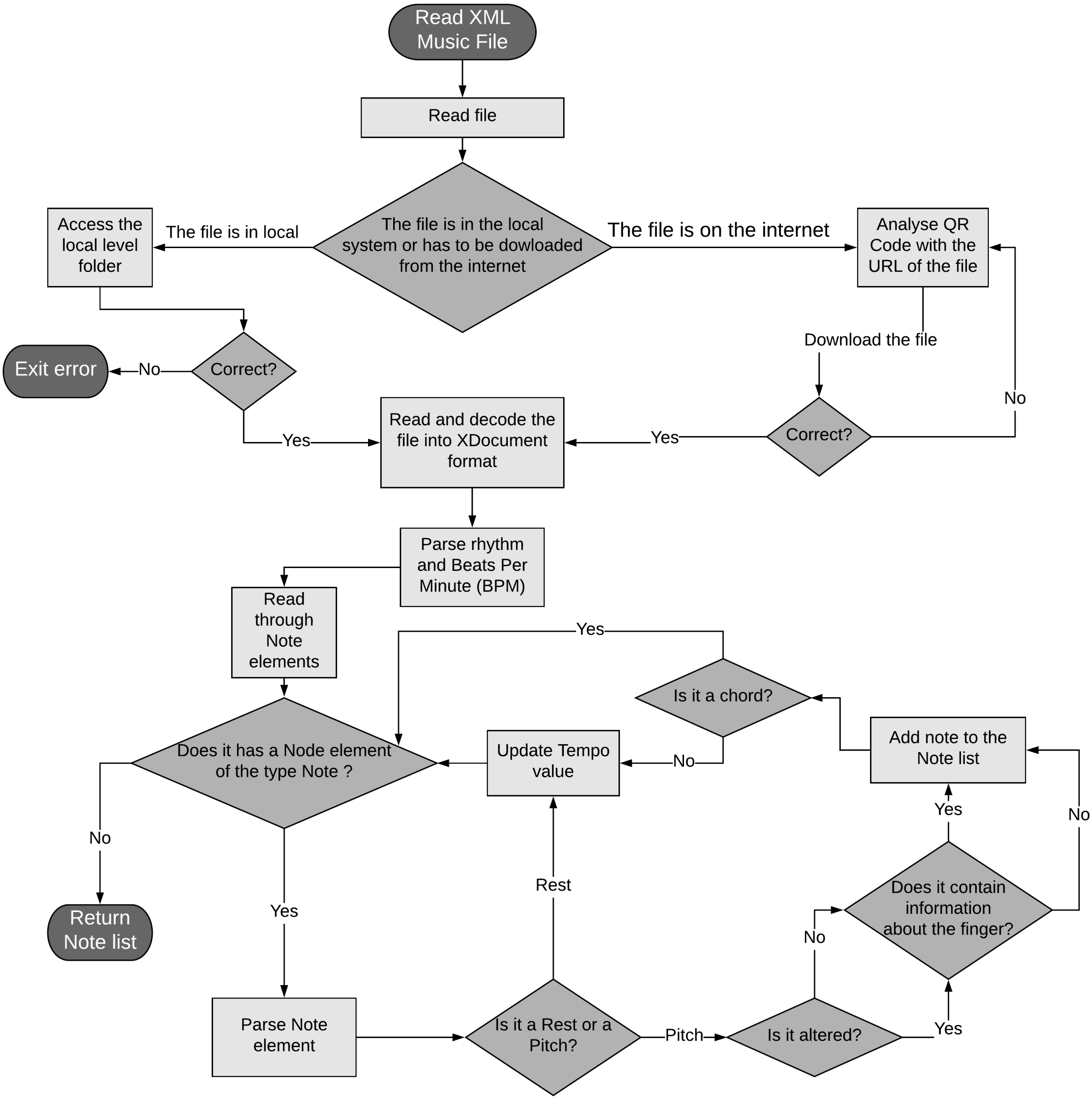


Figure 6.21: XML music reader flowchart diagram

6.5.2 Music sheet reader

In relation with the previous paragraph, the music sheet reader part is in charge of reproducing in the game the information the *MusicSheet* contains. Besides, it is also the manager of all the logic of the execution of the level. There are 4 main parts that constitute this subsystem:

- **MusicSheetManager.** It is a script responsible for sequentially reproducing the information about the song contained in the *MusicSheet* and controlling the execution of the game in relation to the first.
- **Note.** It is the graphic representation of the notes of the score in the game. In this prototype, the notes descend to coincide with a sphere associated with the piano key of the note type.
- **Triggers.** The triggers are the figures that check if the note has been pressed correctly.
- **Feedback.** It stores the information about the execution of the level. It communicates with the *MusicSheetManager* and is responsible for displaying all the information related to the user's feedback. It will be explained in the following section.

MusicSheetManager

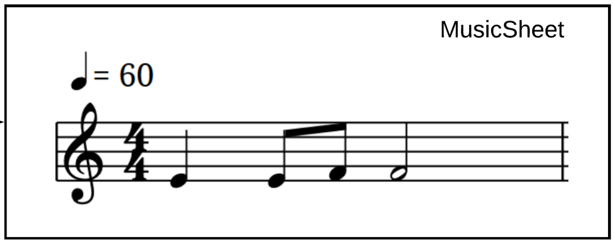
As explained above, *MusicSheetManager* controls the level execution. It can get the *MusicSheet* directly or read a *.xml* file locally through the *XMLRederMusic* class explained above (see figure 6.23). One of its main functions is to read a *MusicSheet*. How does it work? When a level begins a timer is started. This timer updates the values of the measure, beat and part beat. Depending on the Beat Per Minute (BPM) and the time signature (see figure 4.24) of the *MusicSheet*.

Given the example of a song with a time signature of 4/4 and a BPS of 60, the process would be as follows (see figure 6.22). A BPM of 60 means that 60 pulses will be performed in one minute, which is 1 pulse per second. Since it is a 4/4 time signature, there are going to be four quarters per measure. One pulse is equivalent to a quarter note, so every 4 seconds a measure will be played. A *PartBeat* refers to the fourth part of a quarter, which means that this system allows the reproduction up to a precision of sixteenth notes.

In every *PartBeat* update, the *MusicSheet* is asked whether there are any notes in that position. If there are notes, new GameObject of the type Note are introduced into the game (see figure 6.23).

Note

The note is the element that the player visualises and represents a pitch of the song. In this prototype, notes are shaped like a sphere. When a note is created, it is placed at a certain height over the piano. It contains an Unity component called *rigidbody* that gives it the



MusicSheet.getNotes()	♩				♪		♪		𝅗𝅥								
PartBeat	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
Beat	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	2
Measure	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
Timer (second)	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2	2.25	2.5	2.75	3	3.25	3.5	3.75	4

Figure 6.22: Example of how the *MusicSheetManager* reads a *MusicSheet*

weight property that allows it to descend to the piano. The velocity of note descent also depends on the execution velocity of the song. There is *Target* property in each *Note* which indicates which piano key the note should fall on. The finger variable indicates the ID of the finger suggested to the player. This number is located inside the sphere. When the notes fall, they follow a column that makes it more intuitive for the user to know what trigger they are linked to.

It should be mentioned that since this system is developed in Unity, which is a game engine, its architecture is based on frames. Hence, the timer, which invoke the notes in the system, will be checked in each frame. In case of a frame drop, some notes will be lost in the system. However, it is an unusual scenario.

Triggers

When a level starts, the *MusicSheetManager* calls the *LoadTriggers* that creates the necessary triggers depending on the notes that are going to be used, with the aim of not having more noise on the keyboard (see figure 6.23). The triggers are the elements that control when the note should be played. They are represented in the system with purple spheres above each key. Each sphere contains a Unity component called a *collider*. A *collider* provides physical operations in the game such as collision detection. When a *Note* descends and is placed on the trigger it launches an event which indicates whether the note is inside the trigger. This event is captured by the *ActivatorNote*. The *ActivatorNote* also receives the piano event indicating which note has been played. If the key is pressed correctly and the note is located inside the trigger, the *ActivatorNote* will increase the score in the *MusicSheetManager* function. Otherwise it will be decreased.

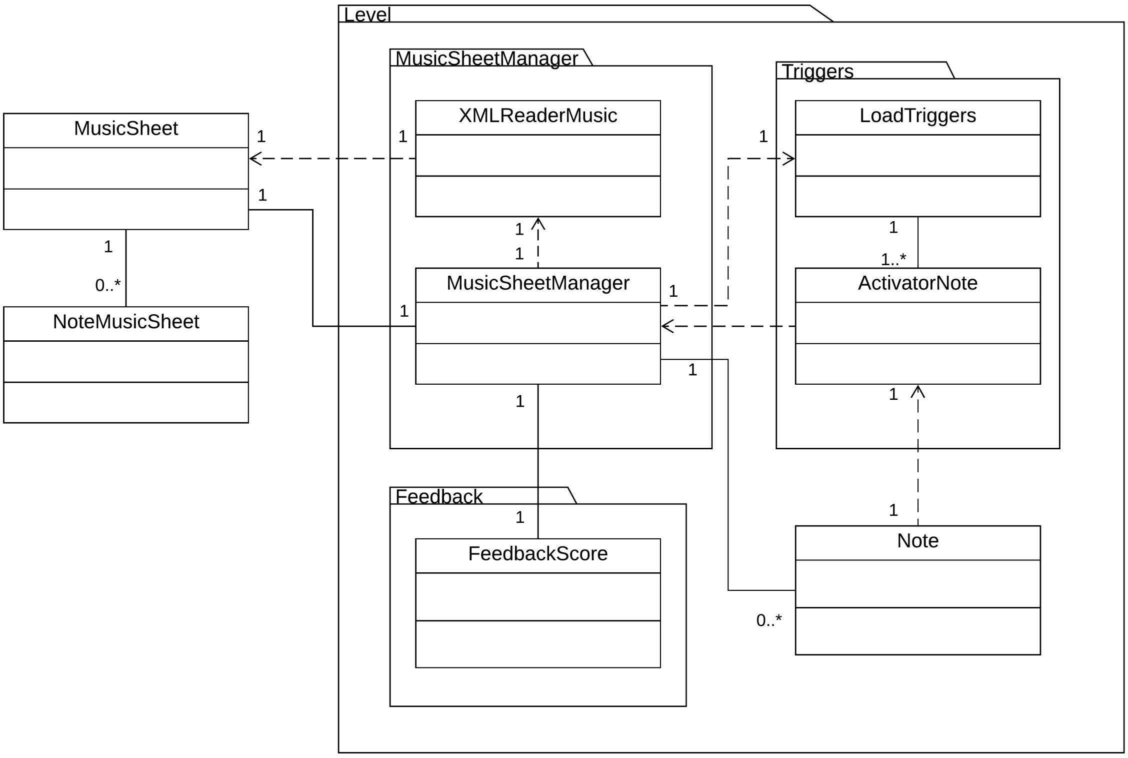


Figure 6.23: Music reader module class diagram

6.5.3 Feedback Module

Last but not least, the feedback module is responsible for displaying the information for the user and incorporating the rewards-oriented gamma elements. It can be distinguish between the **level elements** and **game feedback** subsystem.

Level elements

With the aim of creating a system that can generate competitiveness and attract more player attention, two new elements have been added that work as a reward. These elements also value the user's interpretation of the level. These elements are two:

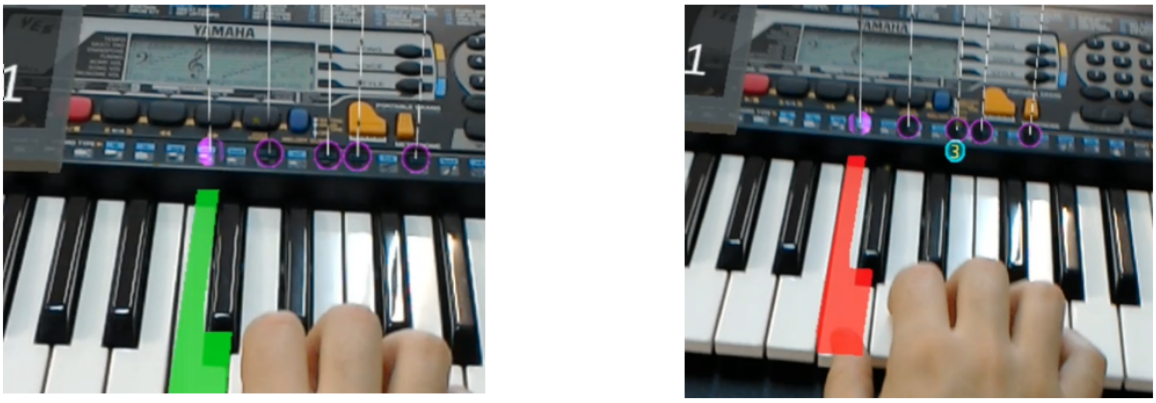
- **Points.** The points are the measure used to evaluate the performance of the user. Each key pressed correctly is a point that accumulates in a **streak**. When the streak grows it can increase the **combo multiplier**. If a key fails the combo and the streak returns to its initial number. The combo multiplies the value of the streak and the sum to the total score. All this information appears on the game side on the piano in the form of a display panel (see figure 6.25).
- **Stars.** The stars represent the prize that the player can obtain at the end of a level. These stars are awarded according to a final score number obtained.

When a level finishes the point and star values are saved in a JSON file. This way, the progress can be registered for the next time. The process of reading and writing the JSON file is performed with the *Newtonsoft.Json*. In the class diagram A.1 it can be observed how it is designed.

Game feedback

This subsystem includes all the functions that display information to the user about his in the game. Specifically, when playing a level, several mechanisms have been designed to provide feedback.

- **Piano key.** It consists of colouring the key that the user is pressing with the colour of the action he has in the system. There are two colours, red to indicate that the key is wrong and green to indicate that it has been pressed correctly (see figure 6.24).
- **Sound.** When a combo number is increased, a sampler with a positive message is reproduced. This message changes depending on the value obtained.
- **Effects.** The effects are star-shaped particles that are instantiated on the trigger of the note that has been correctly pressed.



a) Correct key pressed

b) Wrong key pressed

Figure 6.24: Feedback key pressed

Figure 6.25: Feedback points panel

Chapter 7

Results

In this chapter it will be discussed about the results obtained after the development of this project. The iterations of the project and its results will be detailed. The final product, *HoloMusic XP*, will be described. In particular, which playable levels the final prototype has and why will be explained. How the rewards have finally been represented. And finally, it will be discussed the estimation of the project budget.

7.1 Work distribution

As explained in chapter 5, the development methodology used is iterative and incremental. Throughout this section, it has been summarised the main tasks that have been performed in each iteration. The total number of iterations carried out was nine. At the end of each iteration there was a meeting where new tasks were defined to be performed during the next iteration. The iterations realised were the following:

Development Iterations	
1# Initial Planning	
Starting Date	12/2/2018
Duration	2 weeks
Based on the idea of creating a system that could connect music with mixed reality, the different options were evaluated. It was decided to develop a system that could teach music with a piano using the advantages of mixed reality. Once the idea was clear, the research was carried out on how to connect a piano to the mixed reality glasses. It was concluded that with a linux-based system it would be possible. Researching, it was found a cable that allows to collect the information from the keyboard and process it by USB in the computer, so it was also acquired. It was decided to purchase a raspberry pi to host the linux system and assemble the communications part.	

2# Communication research	
Starting Date	26/2/2018
Duration	2 weeks
In this period the necessary resources became available, specifically the piano, the raspberry pi and the midi USB converter. Linux OS, Raspbian Jessie, was installed in the raspberry and it was investigated how to get the piano data. Finally, the amidi linux tool was found, which allowed to see on screen this data. Also the repositories were created and the necessary development tools were downloaded and installed to develop both in the HoloLens (Unity, Visual Studio, MRToolKit, etc.) and in the Raspberry.	

3# Communication protocols	
Starting Date	26/2/2018
Duration	2 weeks
Once the piano data was available correctly through the cable it was investigated how to send it to the HoloLens. Bluetoth, TCP and UDP protocols were studied and compared in both devices, HoloLens and Raspberry. Tests were performed by sending data from the local raspberrry, then to the HoloLens emulator hosted in the laptop and finally to the HoloLens. The alternative chosen was finally TCP although it was not fully implemented in this iteration.	

4# Communications	
Starting Date	13/3/2018
Duration	3 weeks
In this iteration tests were performed to send TCP packages between the raspberry pi and the HoloLens using static IP. Meanwhile, a shell script was created which allowed the data received by the piano to be passed to the TCP server. For this purpose, since the data was going to be passed locally, UDP sockets were used. In addition, linux pipes were also used. Once this point was reached, the problem of how to send the IP so that the user would not intervene emerged. Then, two alternatives were found: IP discovery and QR code. The first alternative studied was IP discovery because the user had to intervene the least, but finally QR was chosen. As a result, it was researched how to draw a QR code in Python for displaying it on the raspberry screen and how to scan it with the HoloLens. One library was found that allowed this to be done on the HoloLens and another on the user-friendly Raspberry that solved the problem of coding QR.	

5# Tracking and user interface	
Starting Date	3/4/2018
Duration	2 weeks
On the server side, a GUI was implemented to make the application more user friendly. On the client side, the system states were defined and *StageManager* was implemented to control them. On the one hand, the part of the piano driver that translates the keyboard information was implemented. To speed up this process, a piano simulator was created which simulated the real data output of the piano by pressing the computer keys in the Unity Editor. On the other hand, the mapping of the piano position was implemented. Different alternatives were researched and compared with existing applications. Finally, after some successful piano recognition tests, it was decided to use the Vuforia library.	

6# Game prototype	
Starting Date	17/4/2018
Duration	3 weeks
In this iteration a prototyping of the logic of the levels was developed, where some balls descended on the piano keys. the *MusicSheetManager* and *MusicSheet* classes were created which simulate the score with information about the song and the musician who reads this score. Level 1 was introduced. Also, the option to mute the sound of the notes was added. The first prototyping of the game menu was created, which included everything in the same menu, the options and the levels together.	

7# Game scalability	
Starting Date	8/5/2018
Duration	2 weeks
Level 2 was introduced, Ode to Joy. the accuracy of the piano mapping was improved. The feedback system was added to indicate when a key was correctly pressed (the key is coloured green) and when not (the key is coloured red). The fingering in the note figure was added to indicate which finger to play with. The triggers design was changed and a column was added on top of them to follow the notes as they descended. Improved the scoring representation system, adding combos mechanics and a sound messaging system that plays a positive message when a streak is reached. Also, it was studied how music sheet levels could be imported from outside the system, i. e. without the need to recompile the solution again. As a result, a class was designed to read and transform *XMLMusic* to the MusicSheet format of the system, being able to import levels locally.	

8# Game improvements	
Starting Date	22/5/2018
Duration	1 week
In this iteration the star-based reward system was introduced, including persistence. The level menu was also designed and implemented making which is more scalable and visually appealing. Particle systems and animations were added when a star is obtained. From the level selector it is possible to see the results that the player have instead of the old ranking list. The feedback display was changed to a similar to a futuristic screen. Two more levels were added, 3 and 4 where the player has to use his left hand alone. It was implemented the option to import a level based on the *XMLMusic* format using the QR scanner library previously used.	

9# System deployment and documentation	
Starting Date	29/5/2018
Duration	3 weeks
This document was made in this iteration. It is the last iteration, where visual adjustments were made. Some errors were corrected. The last level, level 5, where the player has to play with both hands, was added. And finally, the game icon was included 7.1.	

7.2 HoloMusic XP

HoloMusic XP is the result of the development of this project. A commercial approach to the final product has been attempted. As a result a logo or icon has been designed (see figure 7.1). In figure 7.2 it can be seen how the application looks like from the main menu of the HoloLens. In addition, a **website**[1] has been created where all the multimedia content related to the final system is available.

Figure 7.1: Official *HoloMusic XP* icon

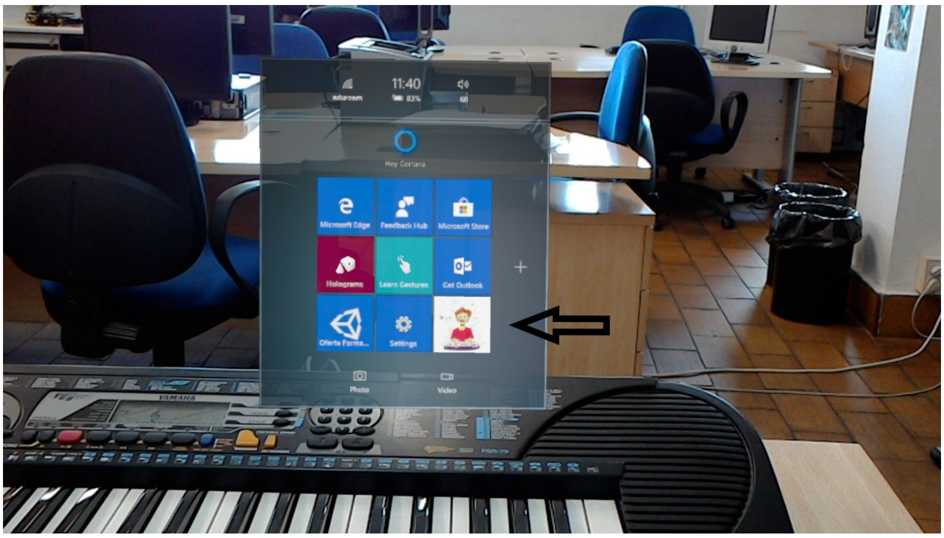


Figure 7.2: *HoloMusic XP* icon view in the HoloLens main menu

[1] `diegomolero.github.io/HoloMusicXP`

a) instructions about the fingering displayed over the piano

b) Positioning of the fingers on the piano keys in relation to figure *a)*

Figure 7.3: Instructions displayed before playing *Level 1*

In the final version of the system a total of 5 levels have been integrated. These levels generate a learning curve from the first to the last. Each level deals with different skills to be learned:

- **Level 1.** It is the initiation level. It represents the minimum difficulty. In it, the student will have to play the song with the right hand without changing the position. The purpose of this level is to familiarise the student with the piano. It mainly teaches the fingering representation system of right hand and how to coordinate the right hand using all fingers.
- **Level 2.** This level is the theme song of the Ode to Joy. As the previous level, it is played with the right hand in the same position. This level is going faster than the last one. The objective is to test the understanding of the previous level and to show that with that little it is possible to interpret a melody.
- **Level 3.** At this level the left hand is worked on. The left hand usually plays chords while the right hand plays the melody. With the aim of learning to move the left hand on the piano keyboard, chords are interpreted sequentially.
- **Level 4.** In relation to the previous level, in this level, the played chord is the same as the previous one, unless the chords are played at the same time and not sequentially by the left hand.
- **Level 5.** In the last level, the previous learning is combined. This song is played with both hands. the left hand plays the chords on the left and the right hand the melody on the right. The goal is to synchronise both hands at the same time.

In order to explain the fingering, the positioning of the hands and the keys to be used, at the beginning of each level, instructions are displayed in the form of a 3D model (see figure 7.3). By touching any key the game can be advanced to play.

In figure 7.4 it can be seen a capture taken during playing at level two. As it can be observed, the blue spheres descend towards the white keys following the thread that joins them. When they collide with the purple sphere, the triggers, the user has to press the key. If he/she presses it correctly, the key will be coloured green as in the image.

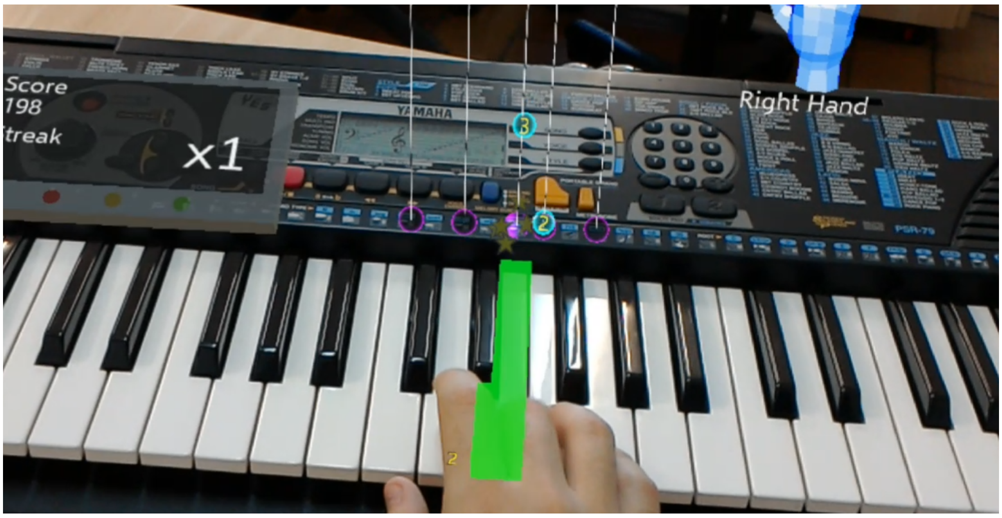


Figure 7.4: Level 2 gameplay. *Ode to joy*

The next capture corresponds to level 4, where the student must become familiar with the chords played with the left hand. As it can be seen, the blue spheres, that represent the notes, are descending simultaneously representing a chord. In addition, the notes include information about which fingers should be used for each note. If the user has doubts about the fingering, he/she can consult the example image, located in the upper right-hand corner of the figure 7.5.

Figure 7.5: Level 4 gameplay. *Chords*

As a result of the gamification elements in the system, it has been possible to design a user-friendly representation system of achievements similar to other popular games. In the level selection menu it is displayed the maximum score and the stars obtained. As can be seen in figure 7.6, the blue stars represent stars not yet achieved and the yellow stars represent stars that have already been achieved.

As explained in the architecture chapter 6, the stars are rewards that are obtained based on the points obtained in that level. therefore, 3 different situations can occur:

1. To indicate that a level has never been played all the stars appear in blue and the score is 0 (see figure 7.6a).
2. To challenge the player to achieve the best level score, when he/she has obtained one star or more but not all of them, the number of points required to obtain the next star is displayed (see figure 7.6b).
3. If all the stars have been obtained, they will all appear with a message advising that the level is complete, however it is possible to continue playing to increase the score (see figure 7.6c).

a) A level that has not been played yet

b) A level where 1 star has been achieved

c) A level with all the stars obtained

Figure 7.6: Rewards display from the levels menu

Figure 7.7: Rewards display when a level is finished

Finally, when a level is finished a text with the score and the stars obtained in that level is displayed along with animations of the appearing stars and particles. If a new better score has been obtained, the "new record" message will appear as shown in the figure 7.7.

7.3 Resources and costs

7.3.1 Project budget

Most of the cost of this project has been spent on the hardware devices.

This project requires very specific and special devices such as the Microsoft Hololens. Although these were borrowed, they will be taken into account for the final calculation. The development edition of the HoloLens currently price 3.000 dollars which in euros is equivalent to 2.595€[2].

The digital piano used is an old device that is no longer available for sale, but the current keyboard models with MIDI output of medium quality cost around 150-200€[3].

The *Raspberry Pi* used is the version 3 model B, which costs around €35. To this device, it has been added a touch screen for 20€of value. And finally, the MIDI USB converter which can be found for 6€[4].

For simplicity, the used laptop and additional peripheral devices have not been taken into account.

In relation to the software, a package of *visual shader*[5] valued at 24,57€has been used. In addition, a one-month license was used to create diagrams in the flowchart application valued at 6€. on the other hand, free student licenses have been used for platforms such as *github*, so they have no impact on the cost of this project.

On the other hand, the project started on February 12 and was completed on June 20. This makes a total of 18 weeks of work or 4 months, rounding out. It has been worked full time. according to a report[6] published in 2015 about the Spanish employment market, a junior computer engineer developer of interactive applications using MR earns an average of 24.000€per year, which makes a total of 2.000€per month. Therefore, it can be estimated that the final budget of an engineer for this project given these conditions would be 8.000€.

As a result, the calculation of the total cost of the development of the project is **10.861,55€** (see table 7.1).

---

[2] www.microsoft.com/en-us/hololens/buy

[3] www.thomann.de/gb/midi_master_keyboards.html

[4] es.myxlshop.com/j-s-supply-cable-usb-midi-interfaz-pc-para-teclado.html

[5] assetstore.unity.com/packages/vfx/shaders/hololens-shader-pack-89989

[6] www.infoempleo.com/informe-infoempleo-adecco/

Resource	Cost	Amount	Total
Microsoft Hololens	2.595€	1	2.595€
Digital piano	175€	1	175€
Raspberry Pi 3 Model B	35€	1	35€
MIDI to USB Converter	6€	1	6€
Touch screen	20€	1	20€
HoloLens Shader Pack	24,57€	1	24.57€
Lucidchart license	6 €/month	1 month	6€
Personnel salary	2.000 €/month	4 month	8.000€
Total budget			**10.861,55€**

Table 7.1: Cost breakdown

7.3.2 Code statistics

During the development process the code was hosted in the *GitHub* repositories. Due to the fact that in the architecture there were two well-defined parts, server and client, Two separate repositories were used.

- **RaspberryPianoServer.** On the other hand, this repository contains the entire part of the server, previously described in the architecture. According to GitHub a 88.8% is python code and 12.2% is shell code. A total of **46 commits** have been made during development in this repository. During the development, it has been added a total of 1,230 and deleted 545 code lines.

  As can be seen in figure 7.8, this part was developed during the beginning of the project and then during the delivery phase some adjustments were made. This is because this repository contains the part of the communications layer of the server that was seen in the architecture chapter 6. Once this part was developed correctly, there was no requirement to improve it.

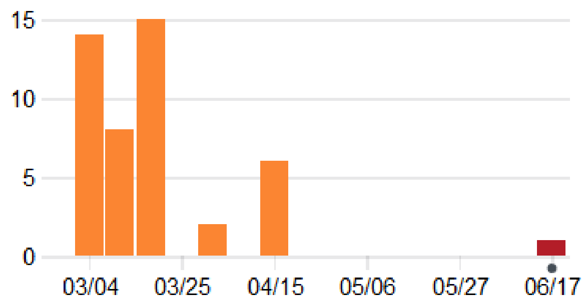


Figure 7.8: RaspberryPianoServer repository. Commits per week

Figure 7.9 represents the amount of code added (in green) and the amount of code deleted (in red) in each week. It can be appreciated 3 initial peaks that coincide with the iterations where the problem of communication between devices was approached.

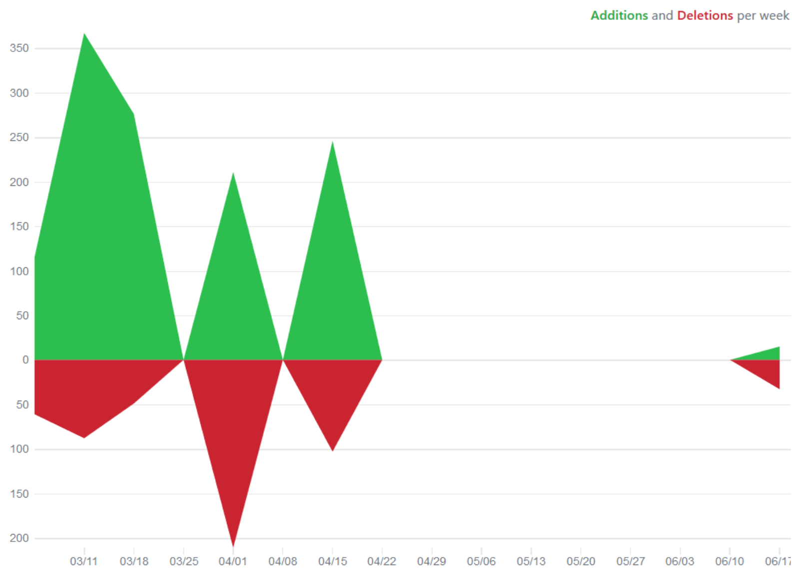


Figure 7.9: RaspberryPianoServer repository. Additions and modifications per week

- **PianoMR.** This repository contains the entire part of the development aimed at HoloLens. The 94.6% of the code in this repository is in C# language. A total of 59 commits have been made during development in this repository.

  Like the other repository, there is a peak of commits at the beginning that coincide with the development of the communications layer. After, there is observed a parabola that decreases at the end of the project.

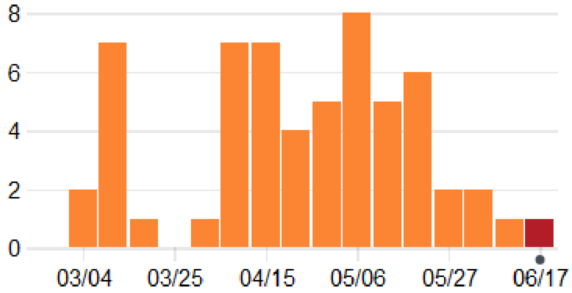


Figure 7.10: PianoMR repository. Commits per week

In figure 7.11 an huge peak can be appreciated at the beginning of the repository. It indicates the creation of the project where all the code was generated and the liberties to be used are added.

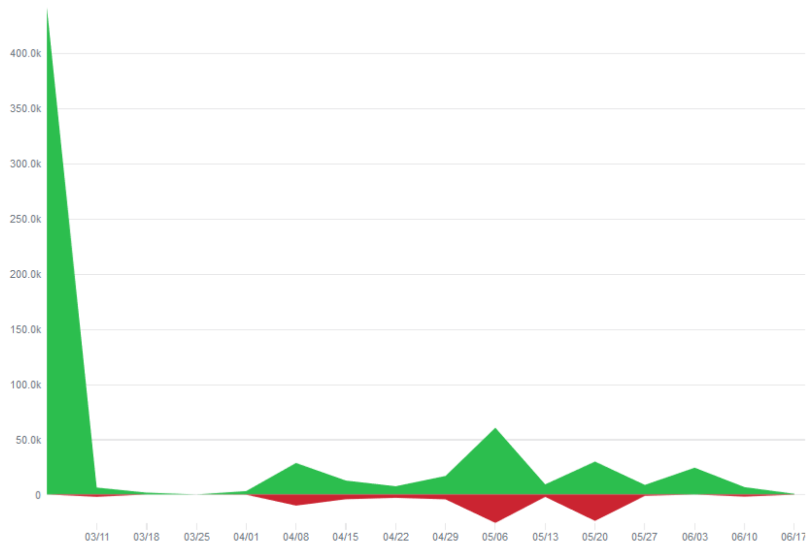


Figure 7.11: PianoMR repository. Additions and modifications per week

Chapter 8

Conclusions and future work

This chapter will describe the conclusions obtained after the development of the project, where it will be discussed how the objectives proposed at the beginning have been approached. Some problems experienced during development will also be discussed. And finally, a set of possible tasks to be accomplished in the future will be listed.

8.1 Accomplished objectives

In relation to the objectives defined at the beginning of the project described in the chapter on 3, it has been determined that all the objectives have been successfully achieved.

The general objective of facilitating the learning of music and piano through the design, creation and development of a system based on mixed reality and gamification techniques through Microsoft Hololens has been accomplished in accordance with the specific objectives.

- **Intuitive learning.** This sub-objective has been achieved with the creation of a representation system of musical notes that abstracts the information and represents the notes of the score in an easy and intuitive way. This form of representation does not take into account the length of the notes but where it is located on the piano, so the user can acquire piano skills by reading and performing a song without having to understand the traditional musical notation system. As a result of using mixed reality, it is possible to relate the notes without knowing what pitches they are, i.e. simply by identifying the pitch with the actual position of the key. This was one of the solutions studied among many others and as it was the easiest solution to understand for a student without previous knowledge of music, it was the one that was implemented. In the chapter 7 other techniques are discussed that can be implemented in the future to teach more concepts such as notes or musical figures.
- **Immediate feedback.** In order to complete this objective of providing the user with immediate feedback about how he or she is performing his or her interpretation, the user is able to see directly on the keyboard, thanks to mixed reality, whether the key pressed is correct or not. Through the colours, the users can associate their actions, with red if they have made a mistake and green if they were correct. Also thanks to

this information, the system will give a final evaluation on the interpretation based on points and stars as rewards. In this way the user will be able to know immediately the results of his interpretation.

- **Incremental learning.** Through the progressive system of levels implemented, this objective has been achieved. As introduced in the chapter on 3, specific learning objectives have been separated at each level. In this way, the student would start at level one and gradually complete the levels. The levels are designed to be repeated several times.

- **Student's motivation.** To increase the motivation of the student who uses the system, some gamification techniques have been incorporated that generate challenges and objectives in the interpretation of the levels. In this way, this sub-objective has been achieved. These have been the elements used for this purpose:

  – *Points:* This measure represents the number of notes that have been played correctly at that level. If the user strings hits without making mistakes, he/she will get more points. With the points a competitive social and challenging factor is added. If a previous score is exceeded, a new record will be set.

  – *Stars:* The stars are rewards that are obtained at the end of a level. The aim is to motivate the student to repeat the level several times until the maximum number of stars has been reached. The stars are obtained based on the points scored.

- **Easy to use** In order to transmit a good experience when using a system based on new technologies it is necessary to be intuitive to use and to execute. In the case of this system, it has been prioritized the most convenient solution for the user. Although it is necessary to perform many tasks before using the system, it has been studied how to automate these tasks without the user intervention. Specifically, the most critical parts are the following:

  – *Deployment of the server on the Raspberry Pi.* It has been built an executable that starts automatically the server and a familiar graphical user interface.

  – *Connection to server.* To connect with the server, since it was necessary to know the IP, it was studied and implemented the system based on the QR codes. Currently, QR code is used in many sectors and is not usually unknown to users. To be able to connect to the piano data provider server it is only necessary to look at the Raspberry Pi screen when the HoloLens indicate it.

  – *Mapping the piano position.* It was also studied the most precise and easy way to map the piano. As a result, a simple and precise way to recognise the position of the piano was implemented through the Vuforia brand recognition library. To do this, the user has to stand on top of the piano and look down following the instructions given by the system.

  - *Interaction with the system* As explained in the chapter 4, Microsoft Hololens use hand gestures to control applications. This system has been designed to avoid the use of these gestures. Only the piano is used to control the actions of the system.

- **Scalability.** This prototype has been designed to be scalable, separating the parts that could be expanded in the future. These are the parts that have been designed to be scalable

  - *Communications layer.* The communications layer is clearly separated from the whole system, allowing in the future to implement a different type of communication such as Bluetooth without affecting the whole architecture.
  - *Range of notes.* The system has been prepared for easy programming of new MIDI processing, such as the involvement of the range of notes that the system processes.
  - *Level creation* To facilitate the creation of new levels and avoid the need to depend on programming skills, a standard XMLMusic format reader has been implemented which allows to translate the music sheet. These files can be created the from an external music score editor, and added to the system as a new level.
  - *Import levels.* To be able to add new levels without having to recompile the application, it has been included the option to import a level from the internet and play it. This greatly expands the scalability of system levels.
  - *Levels menu.* The level menu has also been designed to be fully scalable. From the Unity editor it is possible to add new levels without limiting the functionality of this menu.

8.2 Difficulties faced

During the development process, many difficulties were faced. One of the reasons was that there was no previous knowledge of most of the technologies used. Apart from the above, these were some of the problems.

- Some libraries needed for this project could not be supported by the *Unity* development environment due to the versioning problem mentioned in the chapter on 4. As a result, some functions had to be programmed twice, one for the *Unity* and one for the HoloLens. The problem with programming it in the HoloLens was that in order to test the code it was necessary to execute and debug it directly in the HoloLens. This caused a lot of delay in programming and could already take up to 2 minutes each time the application was reloaded.

- Since HoloLens is a technology that is still under development, there are some versions of *Unity* that are not stable for HoloLens. The unstable versions produced a constant drop of frames in the execution of the application. The holograms also trembled, causing dizziness. Other more stable versions did not have these problems but were not compatible with the Vuforia library because they were too old. Finally, the 2017.2.0f3 version was chosen which is compatible with Vuforia.

- When using the HoloLens portal to live stream the user's view, in addition to lowering the frame rate, it can create conflict with libraries that access to the HoloLens webcam. This forces to stop the video execution and then resuming it when the QR or Piano is finished scanning, in the case of this system.
- One difficulty was how to approach the representation of notes through mixed reality. The idea was that the note would intuitively represent which key it was going to be associated to. Other similar systems were researched in order to compare ideas. Most systems used vertical lines attached to the keys. This solution is not scalable because if there were many notes at once nearby it would be confusing. Also, other systems deployed a keyboard over the real one which is used as an example of the song. This solution was also confusing and the piano model could hinder additional information. Finally, it was decided to use a sphere as a representation of the notes descending towards the key to be pressed. To clarify which key it is, a column is used that follows the note when it descends (see figure 7.4).

8.3 Future work

In the following list there are a number of lines of work that could be done in the future to add new improvements and content to the system:

- One way to add more levels would be to create a **web portal** on the Internet like a online course. Following a format similar to **online courses**, an exercise would be explained and then the QR code would be generated with the link to the XML score. Using the points system, it could be the condition that a minimum number of points be awarded for moving on to the next lesson.
- **New gameplay mechanics** Currently the system uses spheres that represent the notes descending in the direction of the piano keys. As it is a mixed reality system, different mechanics could be created to make the system more varied. For example, a cartoon character who had to overcome obstacles while running. The speed of the character would represent the speed a of the score. The obstacles would be marked with the note to which they belong. If the note of the obstacle is pressed correctly, the player will jump. Otherwise it will collide indicating that it has been wrongly played. In this way, the notes could be learned, as each obstacle represents a note that must be known in order to be skipped.
- Most of the digital pianos that are manufactured today have the function of connecting via Bluetooth to other devices to send you keyboard data. One of the possible inte-

gration in the future would be to add a new Bluetooth **module** to the communications layer. With this, in addition to reaching more devices in different ways, the number of systems implicated in the communication process could be reduced. Having only two devices, the electronic piano and the HoloLens.

- In order to obtain more value when importing levels, it could be included **the option to save them in the HoloLens** in order to avoid having to download them every time a user wants to import a different level. As a result, a menu for selecting the downloaded levels should also be added.
- In this prototype only a total of 13 keys are used, from the central DO (C4) to the (C5). In order to be able to include scores with more notes, these values should be defined in the system so that it can understand them. Since the system was designed for scalability, it is not a difficult task to complete.
- The system currently only includes the option to mute the sound of the notes. It could be included an option to increase or decrease the volume of the effects.
- **Improving the reward system**. Specific achievements could be added by completing specific actions. It could also use the stars to unlock the following levels to make the star system more useful.

Chapter 9

Conclusiones y trabajo futuro

A Lo largo de este capítulo se describirán las conclusiones obtenidas tras el desarrollo del proyecto, donde se debatirán como se han abordado los objetivos propuestos al inicio de este. También se comentarán algunos problemas encontrados durante el desarrollo. Y finalmente, será listado un conjunto posibles tareas para cumplir en el futuro.

9.1 Objetivos cumplidos

En relación con los objetivos fijados a inicio de proyecto descritos en el capitulo 3, se ha podido determinar que finalmente todos los objetivos fueron completados satisfactoriamente.

El objetivo general es facilitar el aprendizaje de la música y el piano a través del diseño, la creación y el desarrollo de un sistema basado en técnicas de realidad mixta y gamificación a través de las Microsoft Hololens ha sido completado de acuerdo al cumplimiento de sus subobjetivos.

- **Aprendizaje intuitivo.** Este subobjetivo ha sido cumplido con la creación de un sistema de representación de notas musicales que abstrae la información y la representa de manera fácil e intuitiva las notas de una partitura tradicional. Este forma de representación no tienen en cuenta la duración de las notas sino donde se encuentra esta tecla en el piano, así, el usuario puede para adquirir destreza en el piano leyendo e interpretando una canción sin necesidad de conocer el sistema de notación musical tradicional. Como resultado de la realidad mixta, se pueden relacionar las notas sin necesidad de saber que notas son, es decir, simplemente identificando la nota con la posición real de la tecla. Esta solución fue una de las que se estudió entre muchas otras y al ser la más fácil de entender para un estudiante sin previo conocimiento sobre música fue la que se implementó. En el capitulo 7 se hablan de otras técnicas que se podrán implementar en el futuro para enseñar más conceptos como las notas o las figuras musicales.
- **Retroalimentación inmediata.** Con el fin de completar este objetivo de informar al usuario de forma inmediata sobre cómo está realizando su interpretación, el usuario podrá ver directamente en el teclado, gracias a realidad mixta, si la tecla pulsada es

correcta o no. A través de los colores el usuario puede asociar sus acciones, rojo si se ha equivocado y verde si ha acertado. También gracias a esta información, el sistema dará una valoración final sobre la interpretación basada en puntos y estrellas en forma de recompensas. De esta manera, el usuario podrá saber inmediatamente los resultados de su interpretación.

- **Aprendizaje incremental.** A través del sistema progresivo de niveles implementado se ha logrado alcanzar este objetivo. Tal y como se introdujo en capitulo 3, se han separado objetivos de aprendizaje concretos en cada nivel. Así, el estudiante empezaría por el nivel uno completando progresivamente los niveles restantes. Dichos niveles están diseñados para ser repetidos varias veces.
- **Motivación del estudiante.** Para incrementar la motivación del estudiante se han incorporado algunas técnicas de gamificación que generan retos y objetivos en la interpretación de los niveles. Esto se ha hecho utilizando dos elementos:
  - *Puntos:* esta medida representa la cantidad de notas que se han tocado de forma correcta en ese nivel. Si el usuario encadena aciertos sin equivocarse obtendrá más puntuación. Con los puntos se crea un elemento competitivo social y de reto. Si se supera una puntuación anterior, se registrará como nuevo récord.
  - *Estrellas:* las estrellas son los premios que se obtienen al terminar un nivel. Estas, buscan como objetivo incentivar al estudiante para que repita el nivel de manera sucesiva hasta lograr el máximo de estrellas. Las estrellas se obtienen en base a la puntuación obtenida.
- **Fácil de usar.** Con el fin de transmitir una buena experiencia al utilizar un sistema basado en tecnologías nuevas es necesario intuitivo a la hora de utilizarlo así como en la ejecución del mismo. Se ha priorizado la solución mas cómoda para el usuario. A pesar de que es necesario realizar muchas tareas previas a la utilización del sistema, se han estudiado cómo automatizar estas tareas sin que el usuario intervenga. Concretamente las partes más críticas fueron las siguientes:
  - *Despliegue del servidor en la Raspberry Pi.* Se ha creado un ejecutable que arranca automáticamente el servidor y una interfaz gráfica familiar para el usuario.
  - *Conexión con el servidor.* Para conectarse con el servidor, ya que era necesario saber la IP, se estudió e implementó el sistema basado en los códigos QR. Actualmente el código QR se usa en muchos sectores y suele resultar familiar para los usuarios. Para poder conectarse al servidor proveedor de los datos del piano solo es necesario mirar la pantalla de la Raspberry Pi cuando las HoloLens lo indiquen.
  - *Mapear la posición del piano.* También se estudió la forma más precisa y fácil para poder mapear el piano. Como resultado, a través de la librería de reconocimiento de marcas Vuforia se implementó una forma sencilla y precisa para

reconocer la posición del piano. Para esto el usuario tiene que situarse encima del piano y mirar hacia abajo siguiendo las indicaciones que da el sistema.

  - *Interacción con el sistema.* Como se explicó en el capitulo 4 las Microsoft Hololens utilizan los gestos de la mano para controlar las aplicaciones. Este sistema se ha diseñado para no tener que usar estos gestos. Únicamente se utiliza el piano para controlar las acciones del sistema.

- **Escalabilidad.** Este prototipo ha sido diseñado buscando ser escalable, separado las partes que podrían ser ampliadas para el futuro. Estas son las partes que se han diseñado pensado en la escalabilidad:

  - *Capa de comunicaciones.* La capa de comunicaciones esta claramente separada de todo el sistema, permitiendo en el futuro poder implementar otro tipo distinto de comunicación, como por ejemplo, el Bluetooth sin que afecte a la demás arquitectura.
  - *Rango de notas.* El sistema ha sido preparado para poder programar de forma fácil nueva información MIDI como puede ser la implicación del rango de notas que el sistema procesa.
  - *Creación de niveles.* Para facilitar la creación de niveles y no tener que depender de saber programar, se ha implementado un lector del formato estándar XMLMusic el cual permite traducir, la partitura la cual puede ser creada desde un editor de partituras externo, y agregarla al sistema como un nuevo nivel.
  - *Importar niveles.* Para poder agregar nuevos niveles sin necesidad de recompilar la aplicación se ha incorporado la opción de importar un nivel de internet y poder jugarlo. Esto expande mucho la escalabilidad de los niveles del sistema.
  - *Menú de niveles.* El menú de niveles también ha sido diseñado para ser totalmente escalable. Desde el editor de Unity se puede incorporar nuevos niveles sin limitar la funcionalidad de este menú.

9.2 Dificultades encontradas

A lo largo del desarrollo fueron muchas las dificultades encontradas. Uno de los motivos fue que no se tenían conocimientos previos sobre la mayoría de las tecnologías que se utilizaron. Quitando esto último, estos fueron algunos de los problemas.

- Algunas librerías necesarias para este proyecto no podían ser soportadas por el entorno de desarrollo de *Unity* debido al problema de versiones mencionado en el capitulo 4. Como consecuencia, había que programar dos veces algunas funciones, una para *Unity* y otra para las HoloLens. El problema de programarlo en las HoloLens era que para testear el código había que ejecutar y depurarlo directamente en las HoloLens. Esto causaba mucho retraso a la hora de programar ya podía llegar tardar hasta 2 minutos cada vez que se cargaba la aplicación desde las HoloLens nuevamente.

- Al ser las HoloLens una tecnología aun en desarrollo, hay algunas versiones de *Unity* que no son estables para las HoloLens. Las versiones no estables producían una caída de frames constantes en la ejecución de la aplicación además de que los hologramas temblaban llegando a producir mareos. Otras versiones más estables no tenían estos problemas, pero no eran compatibles con la librería de Vuforia porque eran demasiado antiguas. Finalmente se escogió la versión 2017.2.0f3 la cual si es compatible con Vuforia.

- Cuando se utiliza el portal de las HoloLens para retransmitir la vista del usuario en vivo, además de bajar la frecuencia de frames, puede crear conflicto con las librerías que acceden a la cámara web de las HoloLens. Esto obliga a parar la ejecución del vídeo y luego reanudarla cuando se termine de escanear el QR o el Piano, en el caso de este sistema.

- Otra dificultad fue cómo enfocar la representación de las notas a través de la realidad mixta. La idea inicial era que la nota representara de manera intuitiva a que tecla iba a estar asociada. Para contrastar ideas, se investigaron otros sistemas parecidos. La mayoría sistemas utilizaban líneas verticales unidas a las teclas. Esta solución era poco escalable ya que, si hubiera muchas notas a la vez cerca crearía confusión. También, otros sistemas desplegaban un teclado encima del real que se utilizaba como ejemplo en la canción. Esta solución también era confusa y el modelo del piano podría obstaculizar la información. Finalmente, se optó por utilizar unas esferas como representación de las notas que descienden hacia la tecla que hay que pulsar. Para aclarar que tecla es, se utiliza una columna que sigue la nota al descender (ver figura 7.4) .

9.3 Trabajo futuro

A continuación, se listará una serie de líneas de trabajo que podrían realizarse para añadir nuevas mejoras y contenido al sistema:

- Otra forma de poder añadir más niveles consistiría en crear un **portal web** en internet a modo de curso. Siguiendo un formato parecido al de los **cursos online**, se explicaría un ejercicio y a continuación se generaría el código QR con el enlace a la partitura XML. Usando el sistema de puntos, se podría poner como requisito para pasar a la siguiente lección obtener un mínimo de puntos.

- **Nuevas mecánicas de juego**. Actualmente el sistema utiliza unas esferas que representan las notas cayendo en dirección a las teclas del piano. Al ser un sistema de realidad mixta se podrían crear distintas mecánicas para así hacer al sistema más variado. Por ejemplo, un personaje que tuviera que ir sorteando obstáculos mientras corre. La velocidad del personaje representaría el ritmo o ritmo a de la partitura. Los obstáculos tendrían una señal con la nota a la que corresponde. Si la nota del cartel es pulsada, el muñeco saltaría. En caso contrario se chocaría indicando que se ha tocado mal. De esta forma se podrían introducir las notas, ya que cada obstáculo representa una nota que hay que saber para poder ser saltado.

- La mayoría de los pianos digitales que se fabrican hoy en día cuentan con la función de conectarse por Bluetooth a otros dispositivos para enviar los datos del teclado. Una de las posibles integraciones en el futuro sería la de añadir a la capa de comunicaciones un nuevo **módulo de Bluetooth**. Con esto, además de conseguir llegar a más dispositivos de distintas maneras, se podría reducir el número de sistemas involucrados en el proceso de comunicación. Teniendo únicamente dos dispositivos, el piano electrónico y las HoloLens.
- Para poder sacar mayor partido al importar niveles, se podría incluir la **opción de guardarlos en las HoloLens** para no tener que descargarlos cada vez que se quiera importar uno distinto. Como consecuencia, también habría que añadir un menú de selección de los niveles descargados.
- En este prototipo solo se usan un total de 13 teclas, desde el DO central (C4) hasta el (C5). Para poder **incluir partituras con más notas** habría que definir estos valores en el sistema para que los entienda. Como el sistema se diseñó pensando en la escalabilidad, no es una tarea complicada de completar.
- Actualmente el sistema solo incluye la opción de mutear el sonido de las notas. Se podría incluir una **opción para bajar o subir el volumen de los efectos**.
- **Mejorar el sistema de recompensas**. Sería posible añadir nuevos logros específicos los cuales se obtendrían al completar acciones determinadas. Además, sería posible utilizar las estrellas para desbloquear los siguientes niveles generando así mayor utilidad al sistema de estrellas.

APPENDICES

Appendix A

Class diagram of the HoloLens system

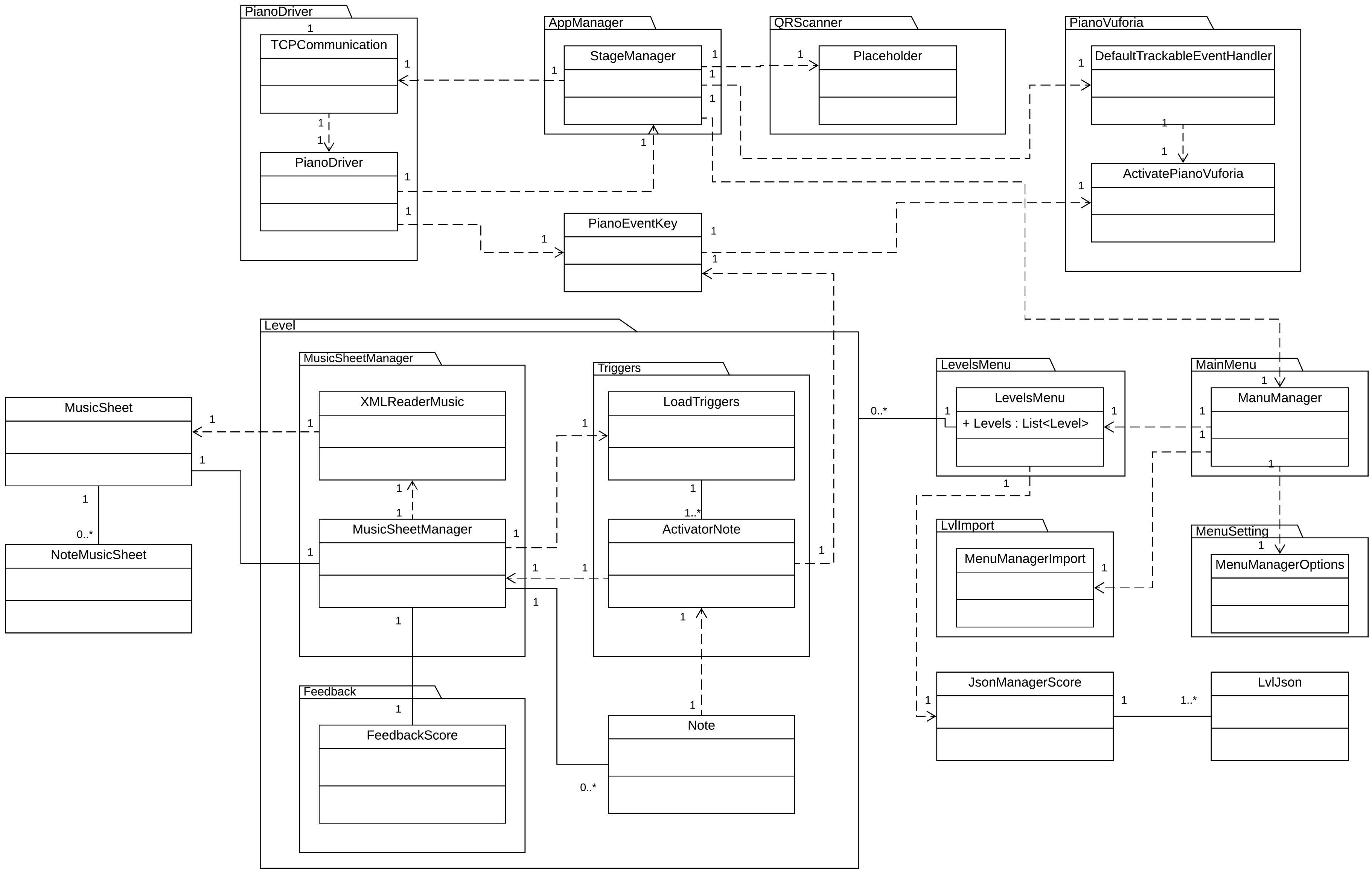


Figure A.1: Class diagram of the entire HoloLens system

Appendix B

Image references

- **Figure 4.2**
  - *a)* en.wikipedia.org/wiki/Sensorama
  - *b)* www.pinterest.es/pin/568649890426092848
- **Figure 4.3**
  - *a)* altvr.com/htc-vive-demo-gdc
  - *b)* www.bbc.com/storyworks/future/future-ready/future-proof?utm_source=nav
- **Figure 4.4** vr.google.com/cardboard/
- **Figure 4.5** www.arreverie.com/blogs/types-of-augmented-reality/
- **Figure 4.6** generated from www.barcode.tec-it.com
- **Figure 4.7**
  - *a)* www.wikitude.com/blog-top-travel-apps-enjoy-summer/
  - *b)* www.hipertextual.com/archivo/2011/08/junaio-la-realidad-aumentada-en-contra\ -de-la-publicidad-en-la-via-publica/
- **Figure 4.8**
  - *a)* www.netzspannung.org/cat/servlet/CatServlet?cmd=netzkollektor&subCommand=showEntry&entryId=130986&lang=de
  - *b)* www.onmsft.com/news/holotour-app-briefly-hits-windows-store-runs-hololens
- **Figure 4.13**
  - *a)* www.voltaico.lavozdegalicia.es/2017/12/holosurg-sistema-realidad-mixta-medicina/
  - *b)* University of South Florida, College of Nursing www.health.usf.edu/nursing/
- **Figure 4.14**
  - *a)* www.arblog.inglobetechnologies.com/?p=446
  - *b)* www.windowscentral.com/microsoft-hololens-app-team-reaches-halfway-point-\ making-galaxy-explorer

- **Figure 4.9**
  - *a)* www.elsnorkel.com/2012/05/simuladores-para-submarinos-solucion.html
  - *b)* store.steampowered.com/app/517160/Richies_Plank_Experience/
- **Figure 4.11**
  - *a)* www.itunes.apple.com/us/app/ikea-place/id1279244498?mt=8
  - *b)* www.breecurry.wordpress.com/advertising-within-the-magazine-industry/
- **Figure 4.12** www.youtube.com/watch?v=Cp03UcY7DoY
- **Figure 4.10** www.digitaltrends.com/virtual-reality/deloitte-survey-shows-augmented\-reality-widespread/
- **Figure 4.15** www.midaslab.ca/portfolio-item/microsoft-hololens/
- **Figure 4.16a** www.docs.microsoft.com/en-us/windows/mixed-reality/gestures
- **Figure 4.16b** www.wired.com/2015/01/microsoft-nadella//
- **Figure 4.17** www.c-sharpcorner.com/news/hololens-emulator-will-be-available-on-march-30
- **Figure 4.18** docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping
- **Figure 4.22** www.cameronvetter.com/2017/01/03/hololens-tutorial-finalize-spatial-understanding/
- **Figure 4.19** blog.jayway.com/2016/05/28/writing-app-hololen-emulator
- **Figure 4.23** mindofmangler.wordpress.com/2014/07/14/theres-no-circle-of-thirds-because\-number-theory/
- **Figure 4.24** www.quora.com/How-many-beats-does-a-quarter-note-receive
- **Figure 4.25** [Kor05]
- **Figure 4.27** [Kor05]
- **Figure 4.28** www.tomobag.blogspot.com/2012/05
- **Figure 4.31**
  - *a)* [CFAW13]
  - *b)* www.youtube.com/watch?v=xlZL9paQEnY
- **Figure 4.32**
  - *a)* www.music-everywhere.co
  - *b)* www.youtube.com/watch?v=aovJh2SxDYU
- **Figure 6.9** newt.phys.unsw.edu.au/jw/notes.html
- **Figure 7.1** Icon made by Elena Delgado Velasco

References

[ABB+01] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[AG15] Dhiraj Amin and Sharvari Govilkar. Comparative study of augmented reality sdk's. *International Journal on Computational Science & Applications*, 5(1): 11–26, 2015.

[AGPB07] Cristina Botella Arbona, Azucena García-Palacios, and Rosa María Baños. Realidad virtual y tratamientos psicológicos. *Cuadernos de medicina psicosomática y psiquiatría de enlace*, 82:17–31, 2007.

[AL99] David Ascher and Mark Lutz. *Learning Python*. O'Reilly, 1999.

[BGP06] J Peter Burkholder, Donald Jay Grout, and Claude V Palisca. *Historia de la música occidental*. Number Sirsi) i9788420691459. Madrid Alianza, 2006.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[BOE+07] X Basogain, M Olabe, K Espinosa, C Rouèche, and JC Olabe. Realidad aumentada en la educación: una tecnología emergente. *Escuela Superior de Ingeniería de Bilbao, EHU. Recuperado de http://bit. ly/2hpZokY*, 2007.

[CFAW13] Jonathan Chow, Haoyang Feng, Robert Amor, and Burkhard C Wünsche. Music education using augmented reality with a head mounted display. In *Proceedings of the Fourteenth Australasian User Interface Conference-Volume 139*, pages 73–79. Australian Computer Society, Inc., University of Auckland, New Zealand, 2013.

[Cho15] Yu-kai Chou. *Actionable gamification:Beyond Points, Badges, and Leaderboards*. Createspace Independent Publishing Platform, 2015.

[DSJ+90] Roger B Dannenberg, Marta Sanchez, Annabel Joseph, Peter Capell, Robert Joseph, and Ronald Saul. An expert system for teaching piano to novices.

In *ICMC*. 1900 International Computer Music Conference, Glasgow, Scotland U.K, 1990.

[FS11] Kevin R Fall and W Richard Stevens. *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley, 2011.

[GOP12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12 (9):11734–11753, 2012.

[Gre14] Jason Gregory. *Game engine architecture*. AK Peters/CRC Press, 2014.

[GT94] Walter J Greenleaf and María A Tovar. Augmenting reality in rehabilitation medicine. *Artificial intelligence in medicine*, 6(4):289–299, 1994.

[HT17] Stefan Hall and Ryo Takahashi. Augmented and virtual reality: the promise and peril of immersive technologies. `https://www.weforum.org/agenda/2017/09/augmented-and-virtual-reality-will-change-how-we-create-and-consume-and-bring-new-risks/`, 2017. Developer Vuforia Portal.

[Inc11] DENSO Wave Incorporated. Qr code essentials. `http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802`, 2011.

[Inc18] PTC Inc. Developing vuforia apps for hololens. `https://library.vuforia.com/articles/Training/Developing-Vuforia-Apps-for-HoloLens`, 2018. Developer Vuforia Portal.

[KB99] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.

[KHM08] Umar F Khan, Shafqat Hameed, and Tim Macintyre. Tcp/ip over bluetooth. *Advances in Computer and Information Sciences and Engineering*, pages 479–484, 2008.

[Kor05] Jono Kornfeld. *Music Notation and Theory for Intelligent Beginners*. 2005.

[Lar04] Craig Larman. *Agile and iterative development: a manager's guide*. Addison-Wesley Professional, 2004.

[LB03] Craig Larman and Victor R Basili. Iterative and incremental developments. a brief history. *Computer*, 36(6):47–56, 2003.

[Mal80] Thomas W. Malone. What makes thins fun to learn?. a study of intrinsically motivating computer games. Technical report, XEROX. Palo Alto Research Center.Cognitive and Istructional Sciences, 1980.

[Mar08] Angel Pastor Martín. Matemáticas en la música. *Ritmo*, 2008.

[Mer17] Tim Merel. The reality of vr/ar growth. `https://techcrunch.com/2017/01/11/the-reality-of-vrar-growth/`, 2017.

[MG96] Tomasz Mazuryk and Michael Gervautz. Virtual reality-history, applications, technology and future. *Institute of Computer Graphics Vienna University of Technology, Austria*, 1996.

[MK94] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[MM01] James E Melzer and K Moffit. Head-mounted displays. *Digital Avionics Handbook*, 2001.

[Oli14] José Luis Leiva Olivencia. *Realidad aumentada bajo tecnología móvil basada en el contexto aplicada a destinos turísticos*. PhD thesis, Universidad de Málaga, 2014.

[Pap08] Marina Papastergiou. Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Elsevier*, 52(1):1–12, June 2008.

[Pos80] Jon Postel. User datagram protocol. Technical report, USC/Information Sciences Institute, 1980.

[RDPS12] Susana Ruiz Davis and David Polo Serrano. La realidad aumentada como nuevo concepto de la publicidad online a través de los smartphones. *Razón y palabra*, 17(80), 2012.

[RM12] Francis Rumsey and Tim McCormick. *Sound and recording: an introduction*. CRC Press, 2012. Chapter 13: MIDI.

[Ste92] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.

[SWK95] Stephen W Smoliar, John A Waterworth, and Peter R Kellock. pianoforte: A system for piano education beyond notation literacy. In *Proceedings of the third ACM international conference on Multimedia*, pages 457–465. ACM, 1995.

[Tau16] Mike Taulty. Windows 10, uwp, qr code scanning with zxing and hololens. mtaulty.com/2016/12/28/windows-10-uwp-qr-code-scanning-with-zxing-and-hololens, 2016.

[Tec18] Unity Technologies. Unity user manual. www.docs.unity3d.com, 2018. Unity Documentation.

[VKP10] DWF Van Krevelen and Ronald Poelman. A survey of augmented reality technologies, applications and limitations. *International journal of virtual reality*, 9 (2):1, 2010.

[Vro17] Dennis Vroegop. *Microsoft HoloLens Developer's Guide, Create stunning and highly immersive 3D apps for your Microsoft HoloLens*. Packt Publishing Ltd., 2017.

[YASV11] C Manresa Yee, María José Abásolo, R Más Sansó, and Marcelo Vénere. Realidad virtual y realidad aumentada. interfaces avanzadas. *National University of La Plata, La Plata*, 2011.

[YWH05] Jun Yin, Ye Wang, and David Hsu. Digital violin tutor: an integrated system for beginning violin learners. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 976–985. ACM, 2005.

Este documento fue editado y tipografiado con LaTeX empleando
la clase **esi-tfg** (versión 0.20180628) que se puede encontrar en:
`https://bitbucket.org/arco_group/esi-tfg`