Specification, design and implementation of a video game for musical language learning

Joan Cueva Moreno

Resum— Al nostre país, l'ensenyament de música als nens i nenes es realitza principalment fora de l'horari lectiu. Per tal de millorar les seves habilitats i incrementar els seus coneixements sobre llenguatge músical, l'estudiant ha de ser perseverant i entrenar regularment. Aquest projecte proposa crear un videojoc, des de la seva especificació fins al disseny i la implementació, per ajudar els estudiants d'una escola de música real en el seu procés d'aprenentatge d'alguns aspectes del llenguatge musical, com ara el ritme o l'escala. Amb aquest videojoc, volem incrementar el temps que els estudiants inverteixen en l'aprenentatge musical, involucrant-los en l'aprenentatge donant-los una eina entretinguda amb què treballar.

Paraules clau— música, ritme, escala, videojoc, *rhythm game*, joc rítmic, especificació, disseny, implementació, Unity, telèfon intel·ligent

Abstract— In our country, teaching music to children is mainly conducted outside the compulsory teaching hours. In order to improve their abilities and increase their knowledge of musical language, the learner must be persistent and train. This project aims to create a video game, from its specification to design and implementation, to help students of a real music school in their process of learning some aspects of the musical language, such as rhythm or scale. With this video game, we want to increase the time invested by students in learning music, engaging them in learning by giving them a fun tool to work with.

Index Terms— music, rhythm, scale, video game, rhythm game, specification, design, implementation, Unity, smartphone

1 Introduction

TEACHING music to children is mainly conducted as an optional activity outside the compulsory teaching hours. In order to improve their abilities and increase their knowledge of music, the learner must be persistent and train often. Like in any other subject of study, making the learning fun and entertaining always encourages the student to invest more time into it and be more passionate about it. Two different ways of achieving this is by the usage of video games or by gamifying the process of learning and working.

Gamification is defined by Growth Engineering [1], an expert organisation in applying gamification to learning environments, as "the application of gaming mechanics in non-gaming environments to make difficult tasks more palatable". Some gaming mechanics that can be applied to the process of gamification are giving the user a score based on their performance, having rewards for achieving big or difficult tasks or showing a scoreboard of all the users. All these elements contribute to make the user become more involved in learning or working.

According to "Gamification in teaching music: case study" [2], gamification is also an effective tool when teaching music. To conduct the research, they applied gamification techniques when granting access to multimedia learning material to a group of music students. Then, the researchers compared the results of this group with those of a control group without access to the gamification contents and another group with access to the multimedia material but without gamification, only as a class activity. The effects of gamification can be seen in the students' motivation and performance: the group with access to gamified activities (Group B) was the one with higher motivation and better performance, as it can be seen in section A1 of the appendix.

This *Treball Final de Grau* aims to create a system, from design phase to implementation, that helps students in their process of learning musical language. For this project we are working with *Aula de So*, a music school in Sant Cugat del Vallès, Barcelona, to create an ad hoc music video game to be used by students as an extra tool outside the classroom. The main characteristics of the project have been defined in several meetings with the music teachers of this school.

Aula de Joc, our video game, aims to assist students in their process of learning concepts of music language, such as rhythm or scale, as well as to improve their skills in these areas. With the development of this video game, we want to address the problem of students not investing enough time in music learning, engaging them in learning by giving them a fun tool to work with.

In this document, we look at examples of the current state-of-the-art video games and gamified applications in section 2. In section 3, we discuss the main goals we set

- *E-mail de contacte: joan.c.m_1998@hotmail.com*

- *Menció realitzada: Enginyeria del Software*
- *Treball tutoritzat per: Daniel Ponsa Mussarra (Ciències de la Computació)*
- *Curs 2019/20*

for this project. In section 4, we talk about the methodology and the planning of the project. In section 5, we talk about the progress done on the project and the video game: product specification, UX design and documenting the game's characteristics, as well as the necessary training in Unity. Finally, in sections 6 and 7 we discuss the results and the conclusions we draw from the product and the project.

2 State of the art

There are several systems that are being used for teaching and entertaining purposes, either as video games or as gamified applications. To have a clear idea of the current trends and software, we explored the state-of-the-art software we can find in the professional business related to music video games and gamification in education.

2.1 Music-related video games

Nowadays, we can find music-related video games whose only purpose is to teach how to play musical instruments, such as Ubisoft's *Rocksmith 2014* [3]. In this video game the user can learn how to play the guitar and the bass guitar from scratch and build their skills to mastery, starting from the initial steps of just playing some basic notes to being able to play a whole musical piece. The game achieves this by progressively increasing the difficulty of the songs the user plays and adding extra mini games. These help the user improve in different aspects of playing the guitar such as correctly positioning their fingers or moving the hand along the guitar's neck.

There are much simpler video games and applications such as *Music Crab* [4] or *Music Tutor* [5], which share a common concept but are targeted towards a different age group. These games help the user learn how to differentiate between the different notes in a scale by making them correctly identify the note that is being shown. *Music Crab* (Fig. 1) is a video game for mobile phones targeted to children where the users must press the note that corresponds to the position of the crab on the stave. The difficulty of the level increases as the game progresses, making the user learn and adapt to being able to correctly identify every note.

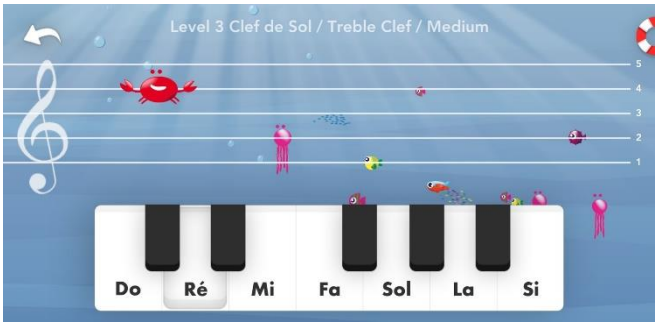

Fig. 1. In-game picture of *Music Crab*.

Rhythm games are music video games that focus on making the players use their sense of rhythm to play. Games like Harmonix's *Rock Band 4* [6], where we can use instrument-shaped controllers to play the guitar, the bass guitar, the drums and the vocals, rely on the user playing to the rhythm of the songs.

On a smaller scale, *WeDrum* [7] is a smartphone application that allows the player to play different songs while they have to follow the song's rhythm with virtual drums (Fig. 2). This game has free and premium songs, and they are tagged by difficulty. It also has a "solo" mode where we can play the drums freely, without a song playing in the background.

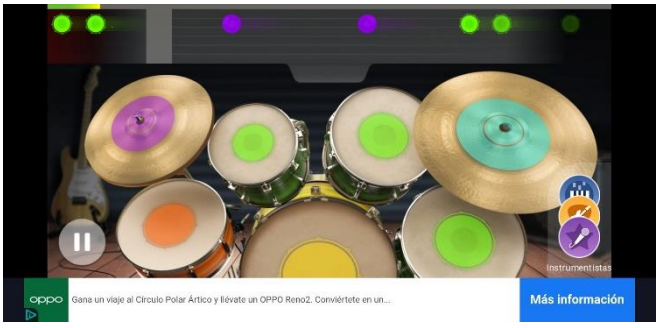

Fig. 2. Playing a song in *WeDrum*.

Even if these video games and applications are interesting options in music learning, *Aula de So* would like to have a tailor-made tool for them that would allow them to follow the evolution of their students and control the content of the application.

2.2 Other teaching video games

Although it is not related to music, some educational institutions have started to find an ally in video games and are now creating their own and incorporating them into their teaching methods. The University of New South Wales (UNSW) in Sydney, Australia, is a good example of this. *Playconomics* [8] is a video game developed by LionsHeart Studios used in UNSW's School of Business. Students have to apply the contents that are taught throughout the course to run a city and earn money while making their citizens happy, which in turn increased the productivity of the city. By doing so, they gain points and they compete among themselves in a public leaderboard.

The score and leaderboards, in addition to the fun of testing your knowledge about the subjects by playing a video game, make the students want to invest more time in the subject and contributes to making learning a less tedious task.

2.3 Gamified applications

Video games are not the only successful solution we have in our reach to make learning more fun. As stated before, we can also use gamification.

Language learning platforms like *Clozemaster* [9] often apply the concept of gamification in their applications and websites (Fig. 3). Investing time in learning a new language is always tedious, but it can be made more enjoyable if the player earns points every time they write the right answer and then compete against players from all over the world to see who has improved the most

during that month.

Fig. 3. *Clozemaster*. The user earns point every time they choose the right answer.

Gamification is also being used in universities. The Universitat Autònoma de Barcelona (UAB) has developed *TOP Enginyeria* [10], a platform where Computer Science students earn badges and points for achieving some goals in their subjects. They are then classified in a global ranking. This encourages students to invest more time in earning the badges (and thus learning the contents of the subjects) and compete among themselves.

3 Main Goals

To give us a clearer idea of the project, we have gathered *Aula de So's* main goals and expectations in a means-goals tree that can be found in section A2 of the appendix. This tree was created after the first meetings with them, where they showed us what they want to achieve with the project:

- Increasing the time invested by students in learning.
- Improving student's ability to recognize notes by pitch and position in the keyboard.
- Improving student's sense of rhythm.

This project was started from scratch. For this reason, we set some first project goals related to the analysis, design and implementation of the project. These project goals can be found in Table 1.

TABLE 1
PROJECT GOALS

Goal ID	Goal Description
PG1	Eliciting requirements for the software.
PG2	Prioritizing the most important parts of the video game in the MVP and the subsequent iterations.
PG3	Designing the architecture of the system.
PG4	Implementing the software.
PG5	Testing the results.

After the initial meetings with the music teachers, as a result of PG1 and PG2, we agreed on a minimum viable product (MVP). In Table 2 we can find the main goals related to the implementation of the characteristics of our software are.

In Table 3 we can find other interesting characteristics of our system. Due to the limited time we had for the project, these were only going to be developed if we had extra time after finishing the MVP.

TABLE 2
MVP'S PROJECT GOALS

Goal ID	Goal Description
PG6	Developing a rhythm game where we apply concepts of gamification.
PG6.1	Developing five different game levels.
PG6.2	Grouping songs by difficulty.
PG6.3	Scoring the actions of the players.

TABLE 3
ADDITIONAL PROJECT GOALS

Goal ID	Goal Description
PG7	Giving trophies/badges to the player when they achieve certain tasks.
PG8	Automating level creation so that new levels are created automatically with only adding the song.
PG9	Having a register/log in system.
PG10	Ranking students by score in a global leaderboard.
PG11	Allowing *Aula de So*'s teachers to change the contents of the video game.
PG12	Generating users/groups progression reports.
PG13	Allowing teachers to send personal notifications to the students.

4 Methodology and Planning

The project has been run in an agile-style development [11], with incremental iterations. We have decided to follow this approach since agile methodology brings us really valuable benefits, such as:

- Having a working piece of code after each iteration.
- Demonstrating functional code to the customers and getting feedback from them.
- Being able to implement changes rapidly and with less time cost, since we do not implement them at the end of the developing process but instead after the iterations are presented to the customers.
- Adapting the product to the customer needs and expectations.

Agile methodology ensures a higher product quality and customer satisfaction and reduces the risks and the cost of changes since we are working closely with the customers and we can get their feedback from actual pieces of code.

In order to plan the tasks of our project, we have created a Gantt chart that can be found in section A3 of the appendix. For the time estimation of each task, we have followed a bottom-up approach. This entails estimating each task individually and then adding all the estimations to obtain the total of each phase and the total estimation for the project but adjusting it so that it fits our schedule for the project.

In our Gantt chart, we have used colours to represent and to easily differentiate each type of task:

- Orange for progress documentation tasks.
- Blue for grouping tasks.
- Dotted green for "Study Unity", since it is a task that will be performed throughout the project.
- Green for project tasks.

We have run an initial requirements analysis phase where we have been working closely with our customers, the music teachers, in order to describe and specify the project and its components. From this phase we have agreed on a MVP, that is the minimum features that ensures a working product that can satisfy the customers.

Each of the following iterations have developed a functional piece of software. During the first iteration, we focused on implementing the software defined at the MVP to ensure that the software is both functional and useful for the main goal.

The second and third iteration were planned to work on the automation of the level creation and the graphical aspect of the video game. However, due to an underestimation of the time it could take to complete each task, we had to use the second iteration and part of the third one to finish the tasks planned for the first iteration.

During the developing of the project, we have used GitHub, a cloud storage platform that allows version control. We have stored code files and documents like the diagrams and reports made. Files like the application's APK, that cannot be uploaded to GitHub due to its size, have been stored in Google Drive. To do the version control, we have used GitHub Desktop, a software tool that checks if there have been any changes to the code and allows us to push and commit the changes to our GitHub repository.

For the implementation of the video game itself, we decided to use *Unity* [12]. *Unity* is a cross-platform game engine that can be used to create video games in three dimensions, two dimensions, virtual reality, etc. We chose *Unity* since it allowed us to create our video game for Android and in 2D. Additionally, it is a very popular game engine, so it is easier to find documentation and tutorials to learn how to use it. Another reason why we used *Unity* is because it has a free version for individuals and small companies, called *Unity Personal*.

5 PROGRESS

5.1 Product specification

Since the start of the project, we have carried out the three meetings with the customers that were planned for the Analysis Phase. These meetings were useful to design and specify the video game they would like and agree on the characteristics of the software and the MVP.

The first meeting gave us a clear idea of what *Aula de So* wanted for the video game. The music teachers sketched an initial idea of what they wanted the video game to look and behave like (Fig. 4), providing us with the source material that we have been working with. They also specified aspects of the video game like its behaviour, its structure and the target audience, giving us a clear and basic understanding of the video game. The information we got was useful to prepare the materials that we needed for the second meeting.

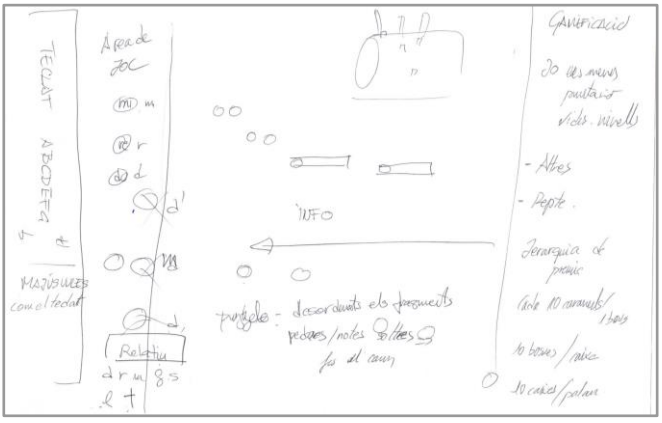

Fig. 4. Initial idea as sketched by *Aula de So* teachers.

The second meeting was focused on creating a Story Map [13] to have an even clearer idea of the requirements of the application. We also agreed on a MVP, which consists of the video game with one song, a presentation of the song (where the user sees the notes scrolling with the music playing, to allow them to get used to the song) and five game levels (that give the user feedback after being played, with three stars representing the score they got). Both the User Story Map and the MVP can be found in [14].

From the Story Map, we have created a Use Case Diagram that can be found in section A4 of the appendix. To document the Use Case Diagram, we have used the draw.io designing tool for UML.

Once we had an idea of the functionalities of our application and we knew exactly what our users can do, we went ahead in the designing of our program and created a Class Diagram that can be found in section A5 of the appendix. For this chart we also used draw.io's tools for UML.

5.2 UX Design

During the third meeting we showed the music teachers paper prototypes [15] that we had created to give them a first contact with what we were going to end up developing and make all the necessary changes to it so to make it adequate to what they want and need. It was also an opportunity to check if the product the wanted had been correctly specified.

This paper prototype includes the menu, the song-selection screen and the levels of it, as well as the feedback the player would receive after playing each of the levels.

The design of the video game follows the initial idea sketched by *Aula de So*'s music teachers, but with some necessary changes to make it work:

- The keyboard was moved from the left side of the

screen to the bottom, since having it to the left of the screen would give no space to place all the keys. It would have also made it extremely difficult and uncomfortable to play, since the user would have probably pressed more than one key with their finger.
- The information column to the right of the screen was deleted to give more space to the actual information of the game. Having the column there would have taken a lot of space from the screen, giving the player less time to react. It would have also distracted the user when playing the game.
- The score information that was in the information column to the right of the screen has been moved to the left. It has also been changed to a score bar to make it easier to understand.

One of the reasons why we moved some elements and deleted others was to adjust it to the screen of a smartphone, since *Aula de So*'s initial design only fit the paper.

During the third meeting, the customers played through the paper prototype and proposed some changes to it:

- In the song selection menu, showing the song's number but also its name with it, to make it easier to identify the song.
- In the song selection menu, having a different screen for each difficulty group.
- In the song selection menu, allowing the user to go directly to the levels that are unblocked for them, instead of forcing them to start from level 1.
- Specifying how the user's score is displayed: with a score bar to the left of the screen.

After presenting them the paper prototype of the video game, we created a higher detailed prototype (Fig. 5) to show them how the actual game could end up looking like.

Fig. 5. Higher detail prototype of the video game's levels.

Aula de So agreed to the design of it and we got their acceptance to start implementing the video game.

5.3 Game's characteristics

During the analysis phase, we have specified what is defined as a rhythm game. Even if there are already several good rhythm games in the market, *Aula de So* asked us to develop a video game because they wanted some specific characteristics that are not found in other games.

One of the reasons why *Aula de So* wanted a video game of their own is to be able to control the songs included in the game, as that will allow the students to practice the same songs they do during the class. In the future, they will also want to keep track of the performance of the students in an automatic way, getting automatically generated reports from the application.

There is not a standard for colouring musical notes. However, *Aula de So* wanted us to use the colour scheme in section A6 of the appendix designed by Boomwhackers [16], since it is the same colour scheme the use with their students at class.

Our video game implements a layout similar to *Music Crab*'s one: there is a keyboard to the bottom of the screen and the user reacts to some notes (in *Music Crab* the note is the crab itself). *Music Crab* only plays with the position of the crab, thus making the player focus on only one note at a time and giving the user a large timespan to react. Conversely, our video game has different notes at once and requires the player to react exactly when the note is played, making them focus not only on the note itself but also on the rhythm.

WeDrum also has a layout similar to what we have developed. In this case, the main difference comes from the musical instrument the user plays with. While in *WeDrum* the user plays with a drum kit and they have to play a different drum depending on the "note", in our video game the user has a keyboard and has to press the right key depending on the note. In our game we also want the user to play with the audio and learn the song, forcing them to play without the visual help of watching the notes, while in *WeDrum* the user can play all the levels relying only on the visual aspect of the game.

The game has a song selection menu that groups the songs by difficulty. Each song has a presentation, where the user can listen to the song before playing. Unlike many other rhythm games, where songs are used only once, in our video game each song will be divided in five sub-levels with different mechanics, as we want our players to get familiarised with and learn the song. The levels, as initially specified, were:

- Level 1: A slower version of the song is played, and the user has to press the corresponding key on the keyboard when the note arrives to the "reference notes" (the notes that show when the user has to press the corresponding key) to the left of the screen. Each note hit gives the user a point, and every mistake (missed note or wrong note hit) takes one point from the user's score. The maximum score is the total number of notes in the song. We can see a small character moving to the position of each note when the player hits it.
- Level 2: Same as level 1, but with the song played at the normal tempo.
- Level 3: Same as level 2, but now some obstacles appear on the screen to make it more difficult and to force the user to play with the sound, instead of rely-

ing on the visuals of it. The obstacles make it difficult to see when the note gets to the "reference note" to the left, but the user can still see when the notes come from the right side of the screen.
- Level 4: Same as level 3, but now the obstacles cover all the screen.
- Level 5: A version of the song without its melody is played, and now the user does not see any note in the screen nor a reference note to know when they have to press the keys. The user will have to press the right keys on the keyboard when the note is supposed to be played. The keyboard makes sound when played. The character now moves from the left side of the screen to the right side, moving a step up when the user hits the right note and moving a step down when the user fails. The Y position of the character represents the current score of the user, while its X position changes with the progression of the song. To the right side of the screen we have three stars, to let the user know what the Y position of its character means in terms of score.

However, after the product validation with *Aula de So*, some changes were made to the video game, as outlined later in our results section.

A presentation level, where users do not need to play but only listen to the song, is played before the user can access to the level they choose.

After each level, the user gets feedback for their performance in the form of stars, with three being the maximum score. It is possible to get half stars. The number of stars depends on the percentage of points gained. However, we grant the three stars every time the user gets 95% of the points or more, to make it a bit easier. If this score is greater than the current one, we save the newest score.

By default, all the levels but the first are blocked. They are unlocked when the score of the user scores 50% or more (at least 1.5 out of 3 stars). Every level has its own individual score. The song's score is calculated as the mean of the each of its levels' scores.

5.4 Learning *Unity*

Before the start of the project, we had a very limited and basic understanding of *Unity*. To be able to develop the project, we have been studying and learning how Unity works and how to implement the concepts that we need for the video game.

For the core of the game, we had to learn how to make a rhythm game. For this purpose, *gamesplusjames'* videos on "How to Make a Rhythm Game" [17] have been very useful. Even if our game behaves in a very different way (as our video game is going to be played on a smartphone and we have to work with the user clicking on the elements instead of pressing buttons), we have used his videos to get a basic idea of the skeleton of our software.

For the menus, we have followed the tutorials in Brackey's YouTube channel on how to create menus in *Unity* [18].

Outside those tutorials, we have been using *Unity*'s User Manual [19] to learn and understand the behaviour of the elements that we needed and how to work with them.

6 Results

The software we have developed meets the characteristics agreed with *Aula de So* in the MVP. We have created a main menu that allows the user to navigate through the game. In this main menu, the user can find buttons to access the different screens of the video game, both the screens that have been developed in this project and the ones that were planned to be developed in the future. However, the screens that have not been developed are currently not accessible. We show this by making the text lighter.

From the main menu, the user can access the song selection menu, where songs are grouped in three difficulties: easy, medium and hard. In this screen, the user can scroll through the available songs in each group, and they can select which song to play. As agreed upon in the MVP, one song has been developed for the video game, "Tinc un gos" [20], that has been classified as an easy song. For the selected song, we have developed the five levels described in section 5.3 Game's characteristics.

For each level, we have manually placed the notes on the screen, adjusting them to correctly match with the song's real notes. To change the speed at which the notes move from level 1 to level 2, we have a script that takes a "tempo" parameter and moves the notes at the screen at such speed. For levels three and four, we have included the obstacles needed to make the user rely on the audio instead of in the visual aspect of the game. These obstacles appear slowly at the beginning of the song. To create level five, we made the character move from the "reference notes" to the stars at the right side of the screen. The character also moves up when hitting the right note, and down when the user fails. In all the levels, the user can find a pause button at the upper right corner of the screen that allows them the resume the level, restart it or go back to the song selection menu.

After finishing each level, the user goes to a feedback screen where they receive their final score for the level, as specified before in the game's characteristics section. In this feedback screen, the user can choose between going back to the song selection menu, playing the same level again and, if they got at least 1.5 out of 3 stars, advance to the next level.

To get a clearer view of the screens and the transitions between them, a UI flow diagram of the video game is attached at section A7 of the appendix. For a clearer view on how the game levels work and their execution, we have recorded a playthrough of the video game, found at [21].

6.1 Product validation

We met with *Aula de So*'s music teachers to show them the video game we had developed and receive their feedback on it. For that, we run a test with them where we asked them to realize some tasks with the software:

- Play the first song without score.
- Go directly to the third level of the first "hard song".
- While inside a level, pause it and return to the menu.

They were able to realize correctly all the tasks we planned for them.

We also asked them for their opinion on the video game and if they had any feedback to give or any changes they wanted to make to it. From that information, we made the following changes to the video game:

- Added clearer visual feedback when correctly hitting/failing a note.
- **Deleted the character that appeared in the levels since it was hard to see.** They decided that it was not necessary and rather not have it anymore.
- Users allowed to listen to the presentation for as many times as they need before starting to play the actual game levels.
- Level 3 changed to make the obstacles appear throughout the level, instead of at the beginning of it.
- Level 5 changed to play the same song as the rest, with the song's melody, since the sound of the piano had a bit of delay and they found it distracting rather than useful.

With the implementation of these changes, *Aula de So* agreed to the final product we have now.

7 Conclusions

The software developed in this project follows the specifications that we have documented. The video game follows the structure agreed with *Aula de So* for the songs and its game levels.

In Table 4 we can see how the work we have done during the project relates to the initial goals we set for the project and the development of the MVP.

We can see how we have completed all the goals related to the implementation of the MVP. Because of this, we can ensure that the video game we have developed meets the goals that *Aula de So* set for the project.

7.1 Future work

We would like to keep working in the implementation of the additional goals we set for the project in Table 3, like automating level creation with only adding the song, allowing *Aula de So* to control directly the songs included in the game or the implementation of a leaderboard with all the users playing.

Additionally, we would like to work on the creation of a "playlist mode" where users can listen to the songs they have played or the creation of more game levels with different mechanics. These options would allow the users to work different aspects of the musical language and would encourage them to keep working and training to improve their musical skills.

TABLE 4
PROJECT AND MVP GOALS

Goal ID	Goal Description	Completion
PG1	Eliciting requirements for the software.	100%
PG2	Prioritizing the most important parts of the video game in the MVP and the subsequent iterations.	100%
PG3	Designing the architecture of the system.	100%
PG4	Implementing the software.	100%
PG5	Testing the results.	100%
PG6	Developing a rhythm game where we apply concepts of gamification.	100%
PG6.1	Developing five different game levels.	100%
PG6.2	Grouping songs by difficulty.	100%
PG6.3	Scoring the actions of the players.	100%

Acknowledgments

I would like to thank UAB's *Escola d'Enginyeria* for the opportunity to work for a real customer.

To Daniel Ponsa Mussarra for all the help provided during the project.

To *Aula de So* for the help with musical concepts and their gratitude.

And to my family and friends for the support during the project.

Bibliography

[1] Growth Engineering, "What is the Definition of Gamification and What Does it Mean?", Growth Engineering, Nov. 13, 2018. Accessed on: Oct. 1, 2019. [Online]. Available: https://www.growthengineering.co.uk/definition-of-gamification/Etc.

[2] C. Gomes, M. J. Guerreiro Figueiredo and J. Bidarra, "Gamification in Teaching Music: Case Study", EduRe 14, Universitat Politècnica de València, València, Spain, 2014. Accessed on: Dec. 19, 2019. [Online] Available: http://hdl.handle.net/10400.2/3478

[3] Ubisoft San Francisco, San Francisco, CA, USA, *Rocksmith 2014*, Oct. 22, 2013. Accessed on: Sep. 30, 2019. [Online] Available: https://rocksmith.ubisoft.com/rocksmith/en-us/home/

[4] E. Zorgniotti, *Music Crab - Notes de musique*, ver. 1.4.2. Accessed on: Sep. 30, 2019. [Online] Available: http://www.musiccrab.com/

[5] JSpalsh Studios, Bangalore, Karnataka, India, *Music Tutor (Sight Reading)*, ver. 2.16. Accessed on: Sep. 30, 2019. [Online] Available: https://musictutorapp.com/

[6] Harmonix Music Systems, Cambridge, MA, USA, *Rock Band 4*, Oct. 6, 2015. Accessed on: Dec. 19, 2019. [Online] Available: https://www.rockband4.com/

[7] Gismart, London, UK, *WeDrum*, ver. 3.22.0. Accessed on: Sep. 30, 2019. [Online] Available: https://gismart.com/featured/new-game-wedrum/

[8] LionsHeart Studios, Sydney, NSW, Australia, *Playconomics*. Accessed on: Sep. 30, 2019. [Online] Available: https://playconomics.com/

[9] Language Innovation LLC, *Clozemaster*, ver. 2.5.4. Accessed on: Oct. 1, 2019. [Online] Available: https://www.clozemaster.com/

[10] Grup d'Interès en Noves Tecnologies Aplicades a la Docència, Universitat Autònoma de Barcelona, Catalonia, Spain, *Top Enginyeria*, 2016. Accessed on: Oct. 5, 2019. [Online] Available: http://top.uab.cat/DF

[11] A. Stellman, J. Greene, *Learning Agile*, Sebastopol, CA, USA, 2015.

[12] Unity Technologies, San Francisco, CA, USA, *Unity*, ver. 2019.1. Accessed on: Sep. 30, 2019. [Online] Available: https://unity.com/

[13] J. Patton, *User Story Mapping*, Sebastopol, CA, USA, 2014.

[14] J. Cueva Moreno, "User Story Map", Miro, 2019. Accessed on: Dec. 22, 2019. [Online] Available: https://miro.com/app/board/o9J_kwTJB1A=/

[15] J. Cueva Moreno, "Paper Prototype v2", Repository TFG-Aso, Specification, 2019. Accessed on: Dec. 22, 2019. [Online] Available: https://github.com/joancm1998/TFG-ASo/blob/master/Specification/Paper%20Prototype%20v2.pdf

[16] Boomwhackers, "About Boomwhackers", Boomwhackers. Accessed on: Feb. 1, 2020. [Online] Available: https://boomwhackers.com/pages/about-boomwhackers

[17] Gamesplusjames, "How to Make a Rhythm Game", Gamesplusjames, YouTube, Nov. 7, 2018. Accessed on: Dec. 22, 2019. [Online] Available: https://www.youtube.com/user/gamesplusjames/search?query=how+to+make+a+rhythm+game

[18] Brackeys, "Menus in Unity", Brackeys, YouTube, Nov. 29, 2017. Accessed on: Dec. 22, 2019. [Online] Available: https://www.youtube.com/playlist?list=PLPV2KyIb3jR4JsOygkHOd2q0CFoslwZOZ

[19] Unity, Unity User Manual (2019.2), Unity Documentation, ver. 2019.2. Accessed on: Dec. 22. [Online] Available: https://docs.unity3d.com/Manual/index.html

[20] Víctor Blasco Blasco, "CICLE INICIAL – Treballem el Sol i el MI", El blog del "Víctor de música", Apr. 16, 2018. Accessed on: Feb. 8, 2020. [Online] Available: http://victordemusica.blogspot.com/2018/04/cicle-inicial-treballem-el-sol-i-el-mi.html

[21] J. Cueva Moreno, "UAB Treball Final de Grau - Aula de Joc", YouTube, Feb. 9, 2020. Accessed on: Feb. 9, 2020. [Online] Available: https://youtu.be/Rb1Jp8xgRyQ

APPENDIX

A1. GAMIFICATION CHARTS

Motivation and performance charts with and without the effects of gamification. The motivation is measured in students conducting extracurricular activities at their own initiative and performances is measured in students with evaluation results above 50%.

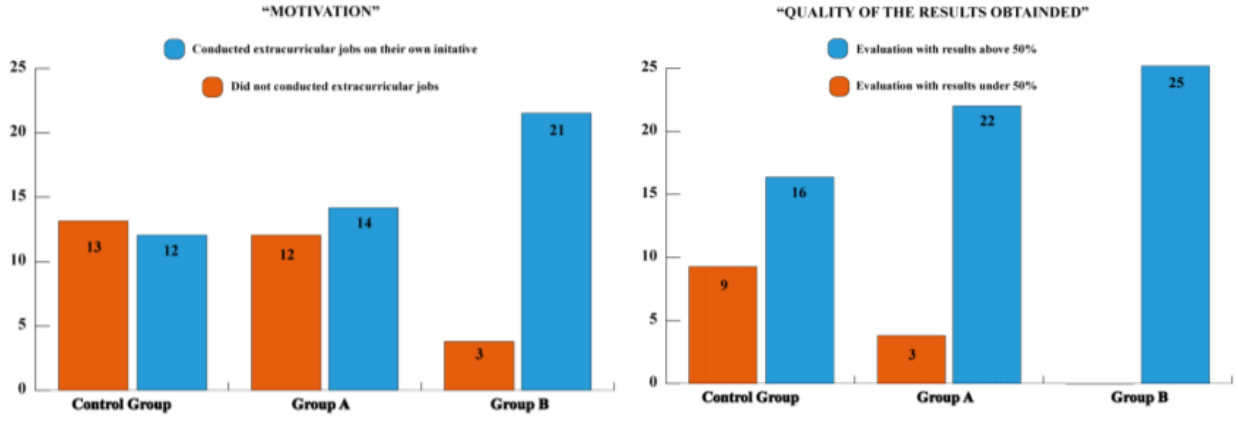

Fig. 6. Motivation and performance charts.

A2. MEANS-GOALS TREE

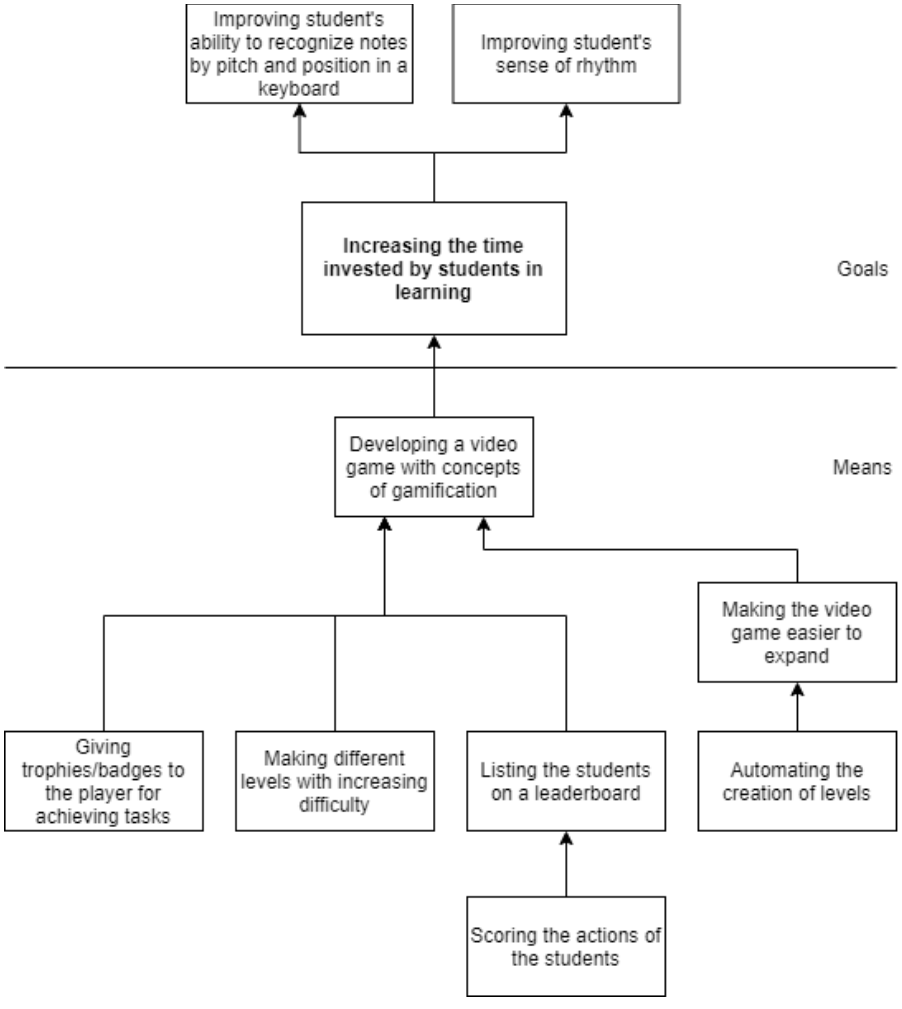

Fig. 7. Means-goals tree.

A3. GANTT CHART

The complete Gantt chart for the project and the Gantt chart split in three charts to make it easier to read.

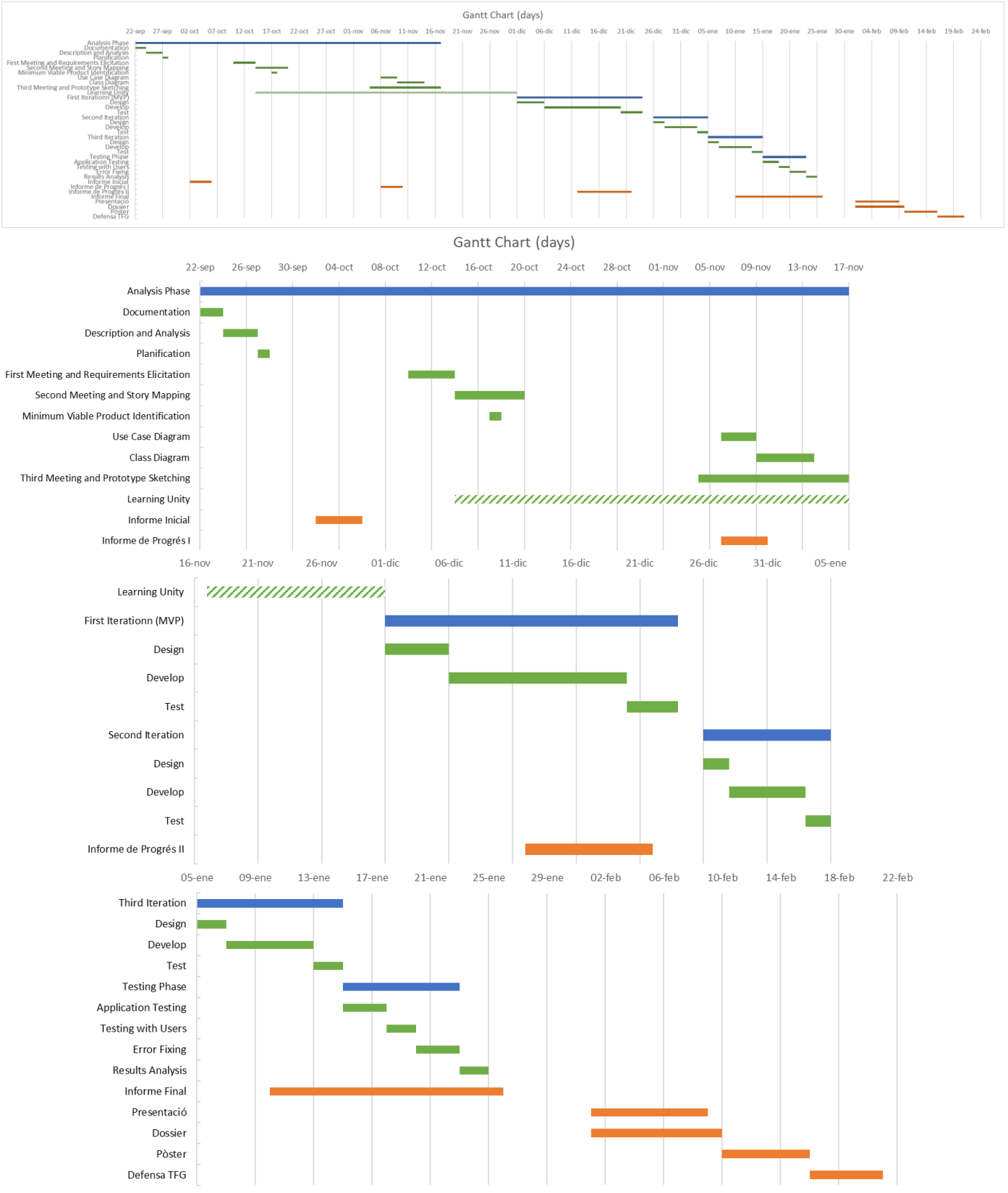

Fig. 8. Gantt chart.

A4. USE CASE DIAGRAM

Use case diagram with the use cases in a bolder outline.

Fig. 9. Use case diagram.

A5. CLASS DIAGRAM

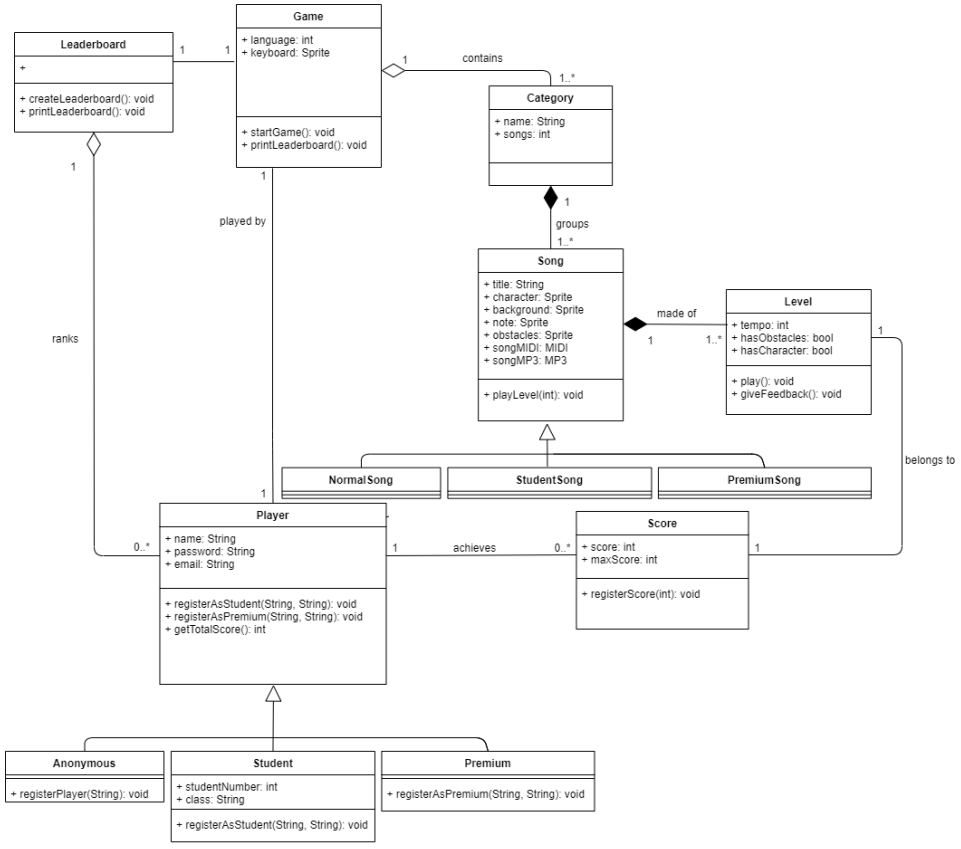

Fig. 10. Class diagram.

A6. COLOUR SCHEME

Fig. 11. Colour scheme used for the notes of the game, with the colour on top of each key of the game's keyboard.

A7. UI FLOW DIAGRAM

UI flow diagram showing the different screens of the video game and the interactions between them.

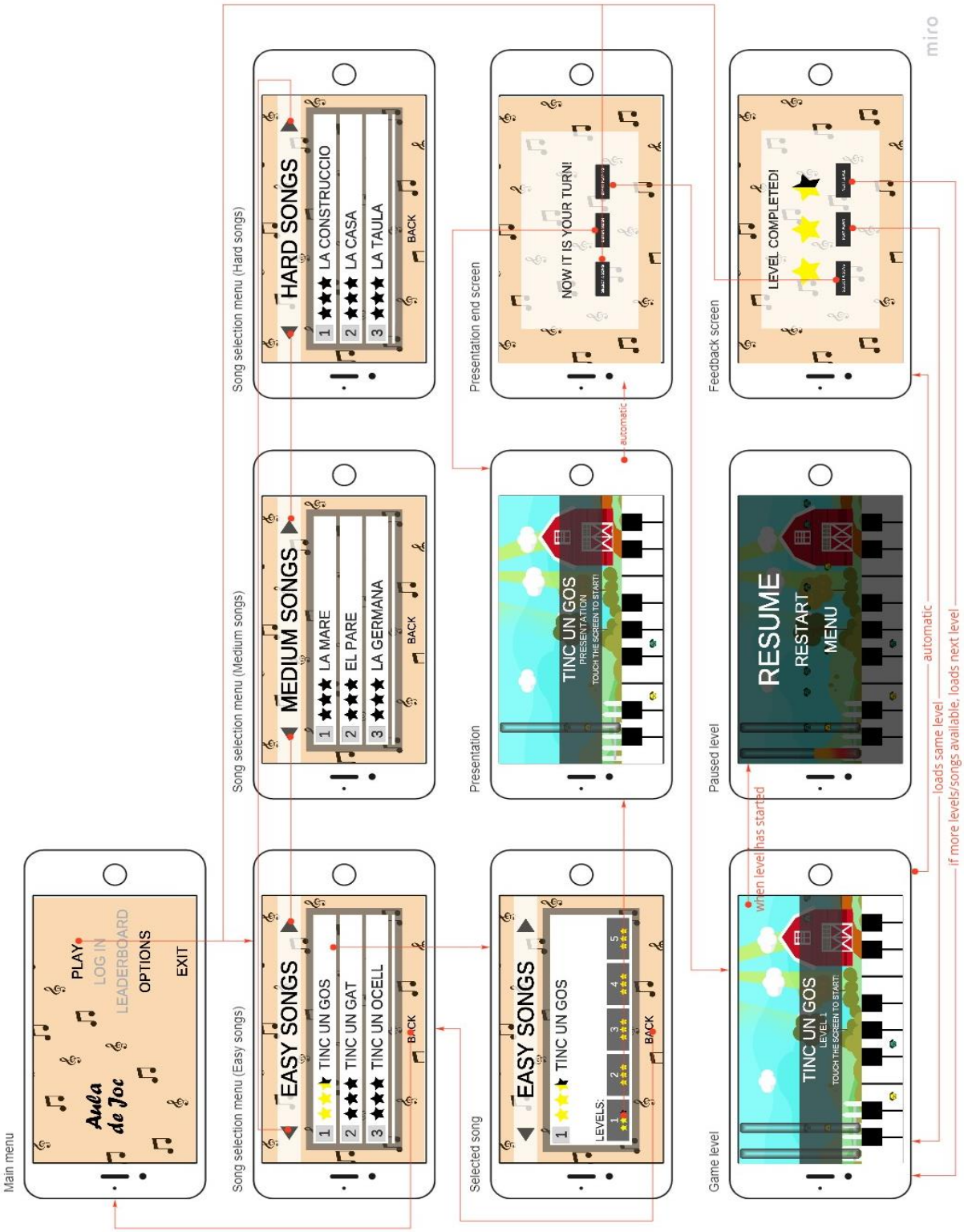

Fig. 12. Screen map.