INITIAL REPORT

INTRODUCTION

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

Related to music, we can find video games whose only purpose is to teach how to play instruments, such as Ubisoft's *Rocksmith*. In this video game the user can learn how to play the 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.

Much simpler video games and applications such as *Music Crab* or *Music Tutor*, each one targeted towards a different age group. These 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 have to press the note that corresponds to the position of the crab on the stave. As the game progresses, the difficulty of the level increases.

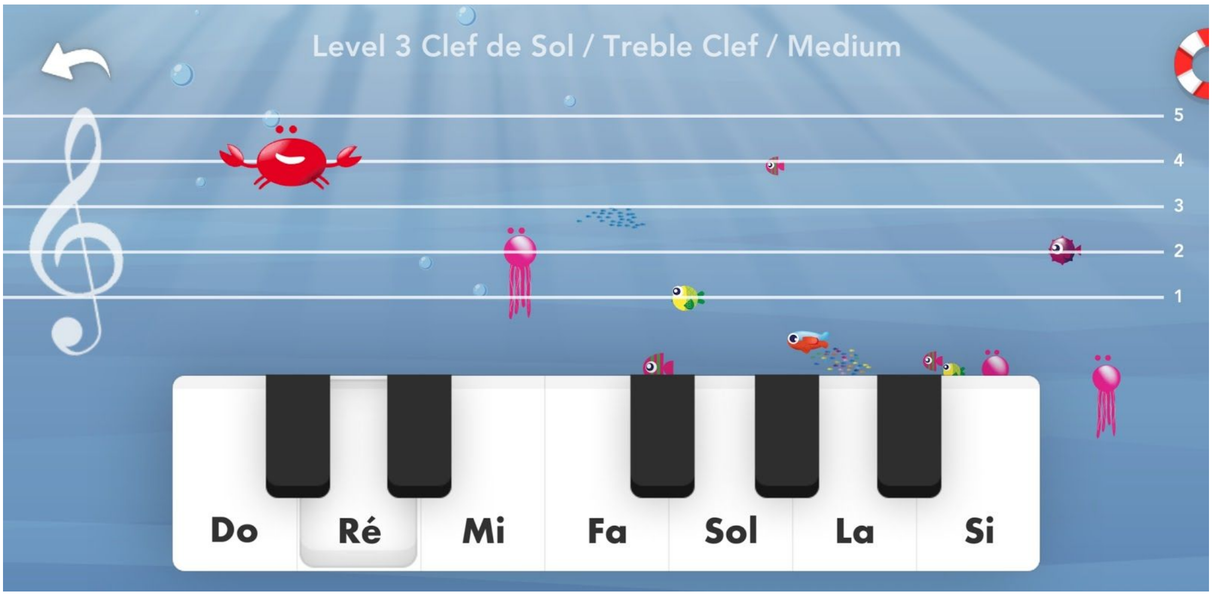
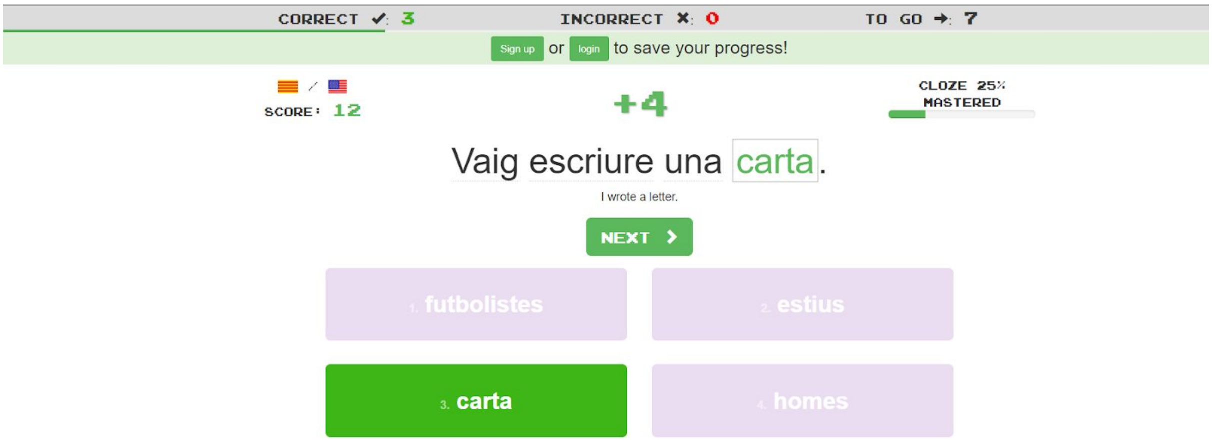


Fig 1. Music Crab.

Nowadays, 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* 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. This feature, added to the fun of testing your knowledge about the subjects and getting some extra marks by playing a video game instead of by taking tests and writing reports, makes the students want to invest more time in the subject and contributes to making learning a less tedious task.

Sometimes, even if we want to make learning be more fun, a video game may not be the best solution. We can then use gamification. According to Growth Engineering, an organisation expert in applying gamification to learning environments, gamification can be defined as "the application of gaming mechanics in non-gaming environments to make difficult tasks more palatable". Language learning platforms like *Clozemaster* (Fig 2) often apply this concept in their applications and websites. 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 of all over the world to see who has improved the most during that month.



Sign up or login to save your progress!

Vaig escriure una carta.

I wrote a letter.

Fig 2: Clozemaster. The user earns points every time they choose the right answer.

Gamification is also being used in universities. The Universitat Autònoma de Barcelona (UAB) has developed *TOP Enginyeria*, 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.

MAIN GOALS AND ANALYSIS OF THE PROJECT

A music school wants us to develop an application that aims to help music students to learn concepts relevant to the subject, such as rhythm, scale or music history, as well as to improve their skills. The application is to be used as an extra tool by the students outside the classroom. To get them to invest their time and have them improve their skills, we want it to be as fun and entertaining as possible. For this reason, we will be developing a music video game. The main characteristics of it, such as the genre, visuals and game structure, will be defined in future meetings with music teachers of the school.

To give us a clearer idea of the project, we have developed a means-goals tree (Fig 3). This is only an initial tree and it will be further developed after the future meetings with the music teachers.

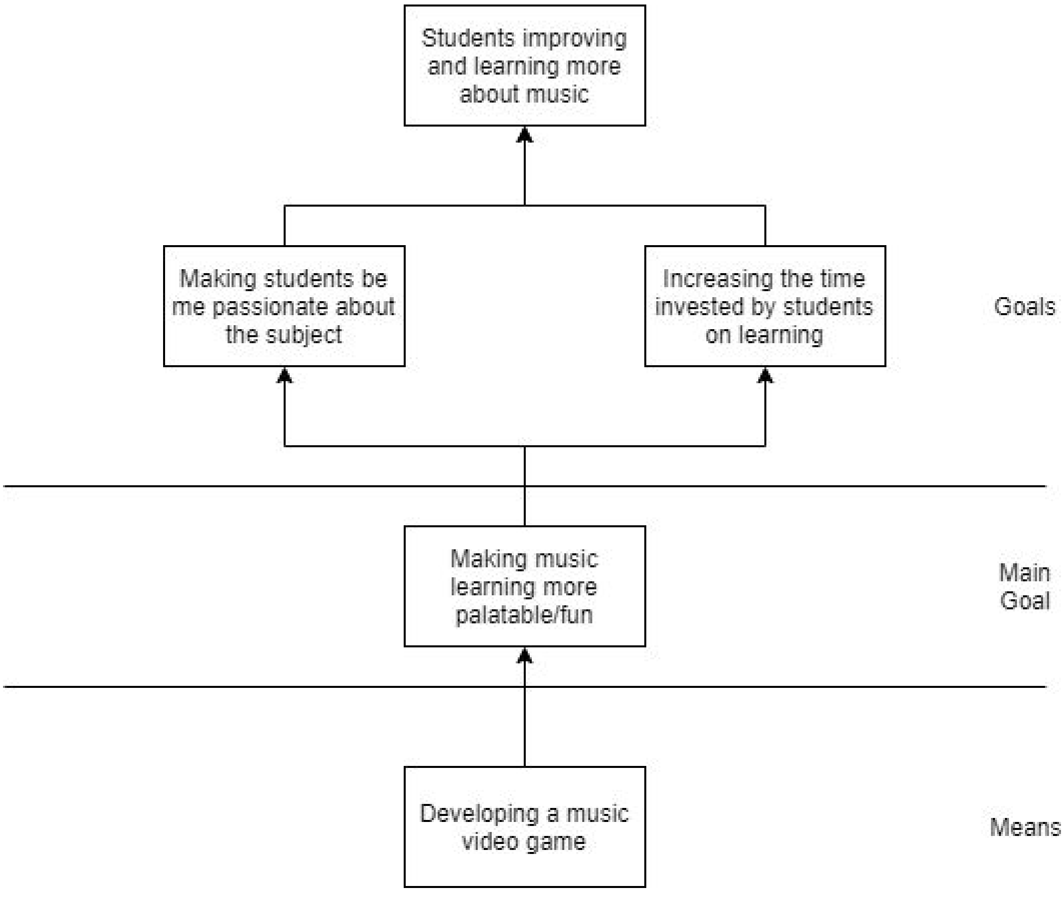

Students improving and learning more about music
Making students be me passionate about the subject
Increasing the time invested by students on learning
Goals
Making music learning more palatable/fun
Main Goal
Developing a music video game
Means

Fig 3: Means-goal tree of the context of our project.

With the means-goals tree we can more clearly see what we are trying to achieve with the project and why. It also helps us understand through which means we can achieve those goals.

We want to analyse the project from four different perspectives: subject, usage, IT system and development.

SUBJECT

The video game will be further defined in future meetings with the music teachers that are interested in it. Until then, we predict that the video game will be composed of different mini games. These mini games will be related to the musical aspects that the teachers think are more important, such as rhythm, scale, pitch, note value and more.

Depending on the mini game, we could want it to be dynamic and adapt to different songs and music sheets that the teachers would upload.

Probably, we will also want a leaderboard so that the students can compete among themselves and have a feeling of improvement. This means that we will have to store information about the users such as a name or nickname and their score. We will also want to store the score of every game they play.

USAGE

The application will mainly be used by the music students. Most of their time using the application will be dedicated to playing the mini games. They will need to give a name or nickname to store their score under that name. Their score will be displayed in a global ranking that will always be accessible.

Music teachers will also want to access the game. They will be able to play the current mini games, update some of them with different music sheets and access the global ranking to see which student is doing better.

IT SYSTEM SUBJECT

The video game will be developed using Unity. We want it to run in smartphones and tablets if possible. For the teachers to update the mini games with new music sheets and for the students to download them, we will need the application to connect to the Internet. This music sheets will need to be stored in a database. This database will also need to store the information displayed at the leaderboard.

The game itself will be published in the app store, where we will also upload any future updates.

DEVELOPMENT

To develop this game we will need access to Unity. According to Unity’s Terms and Conditions, we can use Unity for the development of this game because the company or the developer (in this case, me) does not have more than $100.000 of budget a year. We will likely also need to work with additional libraries that may not be free.

We will need access to an Integrated Development Environment (IDE) for working in databases with SQL. Oracle Corporation provides with a free IDE called Oracle SQL Developer.

To work with all these tools we will need a computer. We will also need a smartphone to test the software we develop.

If we want the video game to be fully developed, it would be necessary to hire a graphic designer to work on its art.

PLANIFICATION AND METHODOLOGY

We have created a Gantt Chart (Fig 4) in order to planify the project. As the project is spread all throughout the semester, we have decided to estimate each task in hours without a start date. For the estimation, we have followed a bottom-up approach; estimating each task individually and then adding all the estimations to get the total of each phase. The total estimated time for the completion of the project is of 252 hours.

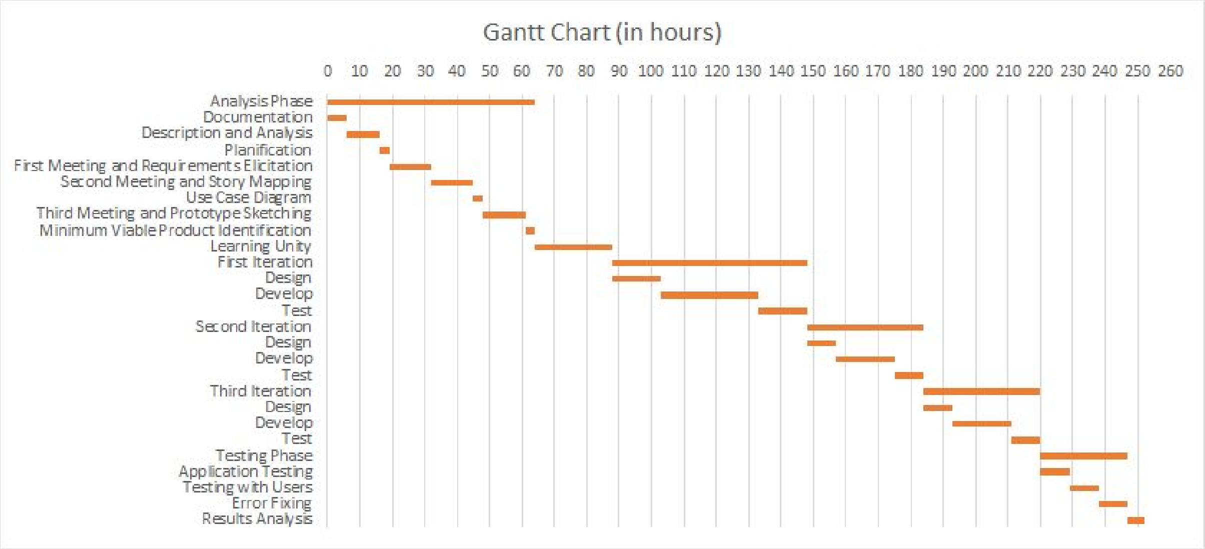


Fig 4: Gantt Chart of the project.

The project will we run in an agile-style development, with incremental iterations. Each iteration will develop a functional piece of software. We intend to develop the Minimum Viable Product during the first iteration to ensure that the software is both functional and useful for the main goal. Because we will be following an agile approach, any extra functionality will be developed during the second and third iterations.

We will be using cloud storage platforms to save the documents, figures and code of the project. For better optimization, we will store code files in a GitHub repository and the rest of the files in a Google Drive folder, so that they are accessible from everywhere.