

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. A més, l'estudiant ha de ser consistent i entrenar regularment per tal de millorar les seves habilitats i incrementar els seus coneixements sobre la música. Aquest projecte pretén crear un videojoc, des de la seva especificació fins al disseny i la implementació, per ajudar els estudiants en el seu procés d'aprenentatge d'alguns aspectes del llenguatge musical, com ara el ritme o l'escala. Amb aquest videojoc, volem abordar el problema que els estudiants no inverteixen prou temps 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. Additionally, the learner must be consistent and train often in order to improve their abilities and increase their knowledge of music. This project aims to create a video game, from its specification to design and implementation, to help students in their process of learning some aspects of the musical language, such as rhythm or scale. With 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.

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 addition, 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 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” [8], 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. According to the study:

“The research adopted a methodological approach of single case study, observational mode, being that the conclusions obtained seem to point to an increase in the level of internal motivation in groups in which they used the multimedia materials. In fact, the Group B (with access to the multimedia materials and the associated game) has been shown to have developed quality skills in all the areas observed, as well as other adjacent domains, such as socialization and the recovery of hidden curriculum” [8, p.17]

The effects of gamification can also be seen in the student’s motivation and performance [8, p.15-17]: the group with access to gamified activities (Group B) was the one with higher motivation and better performance [ANEXO].

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

- 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

music teachers of this school.

This 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.

Resum dels apartats del report.

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* [2]. 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* [3] or *Music Tutor* [4], which share a common concept but are 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 staff. 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 the music video games that focus

in making the players use their sense of rhythm to play. Games like Harmonix's *Rock Band 4* [5], 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* [6] 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.

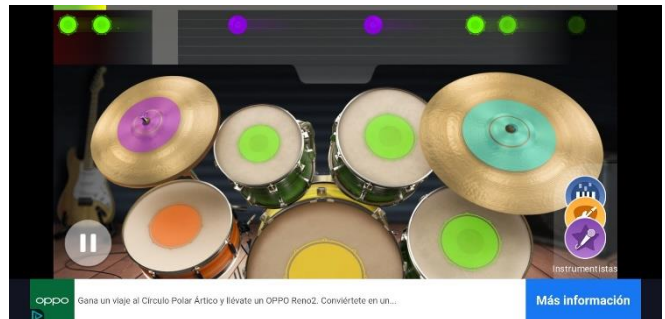


Fig. 2. Playing a song in WeDrum.

Even if this 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* [7] 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.

2.3 Gamified applications

Video games are not the only successful solution we have on our reach to make learning be 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 4). Investing time in learning a new language is always tedious, but it can be made more en-

joyable 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.

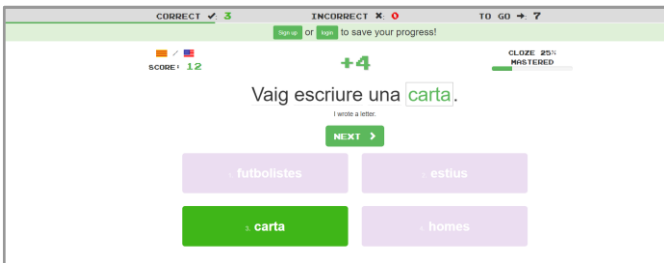


Fig. 4. Clozemaster. The user earns point every time they choose the right answer.

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 developed a means-goals tree (ANEXO). This tree and its goals have been further developed from our initial means-goals tree that we created for our Initial Report [], as the project has been further detailed during the meetings with the music teachers. This tree is based on the goals and expectations the music school, *Aula de So*, has regarding the project.

When we started the project, it was completely undefined. For this reason, we set some 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 have 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 that, due to the limited time we have for the project, are not going to be developed now.

TABLE 2
MVP'S PROJECT GOALS

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

TABLE 3
OTHER PROJECT GOALS

Goal ID	Goal Description
PG7	Giving trophies/badges to the player when they achieve certain tasks.
PG8	Automating level creation.
PG9	Having a register/log in system.
PG10	Ranking students by score in a global leaderboard.
PG11	Allowing <i>Aula de So</i> '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

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.

We have run an initial 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. These iterations could not be completed due to an underestimation of the time it could take to complete each task.

After showing the product to the customers, we got their feedback [referencia a lista de feedback] on different parts of the video game and we worked on implementing those necessary changes.

During the developing of the project, we will be using GitHub, a cloud storage platform that allows version control. We will store code files and documents like the interviews' outcomes and reports made. If there is anything that we cannot upload to GitHub, we will store it in Google Drive. To do the version control, we will be using GitHub Desktop, a software tool that checks if there have been any changes to the code and allow us to push and commit the changes to our GitHub repository.

For the implementation of the video game itself, we decided to use Unity []. 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 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 PLANNING

In order to plan the tasks of our project, we have created a Gantt Chart [ANEXO]. 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 get 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 make it easy to 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 done throughout the project.
- Green for project tasks.

6 PROGRESS AND RESULTS

6.1 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

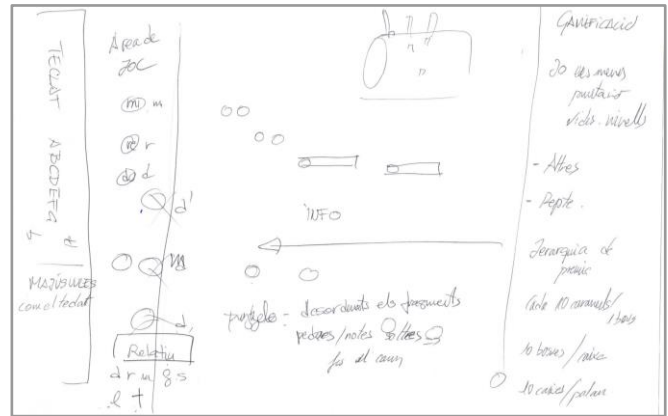


Fig. 7. Initial idea as sketched by *Aula de So* teachers.

sketched an initial idea of what they wanted the video game to look and behave like (fig 7), 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.

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

From the Story Map, we have created a list of Use Cases [] and a Use Case Diagram [ANEXO]. To create the Use Case Diagram, we have used the draw.io designing tool for UML. The current Use Case Diagram received some changes after the feedback from the tutor:

- Teacher Use Cases:
 - "Rearrange contents" renamed to "Rearrange songs".
 - "Add content" renamed to "Add song".
- Player Use Cases:
 - "Update application" deleted since it is something done externally (using Google's Play Store or Apple's App Store).
 - "Update contents" renamed to "Update songs".
- MVP's Use Cases' outline made thicker to make them easier to recognize.

Once we had an idea of the functionalities of our application and we knew exactly what our users will be allowed to do, we went ahead in the designing of our program and created an initial Class Diagram that can be found in our Progress Report I [14, p.5]. For this chart we also used draw.io's tools for UML. However, we had to modify this diagram to include changes and feedback from the third meeting, the tutor and the actual pro-

gramming of the video game:

List changes made to the class diagram.

The current version can be found in [ANEXO].

6.2 Game's characteristics

During the analysis phase, we have been specifying 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.

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:

- 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 hit note gives the user a point, and every fail (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 normal version of the song.
- 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 relying 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 main notes 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 a 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 then 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.

The presentation is played before the user access to the level they choose.

After each level, the user gets feedback from 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 everytime 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 unblock when the score of the user is 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.

Parlar del color de les notes.

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 will have different notes at once and will require the players 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 close 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

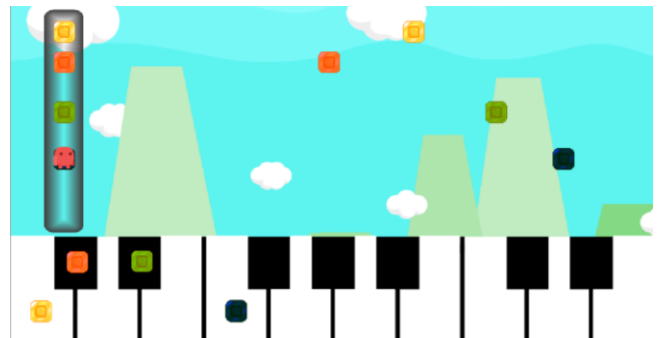


Fig. 1. Initial idea as sketched by *Aula de So* teachers.

the notes, while in *WeDrum* the user can play all the levels relying only on the visual aspect of the game.

Another reason 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.

6.3 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 they 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 adapt it to the screen of a smartphone, since *Aula de So's* initial design was thought only to 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 him, 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 10) to show them how the actual game could end up looking like.

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

6.4 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" [16] have been very useful. Even if our game behaves in a very different way (as our video game is going to be played in 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 [17].

Outside those tutorials, we have basically been using Unity's User Manual [18] to learn and understand the behaviour and how to work with each of the elements that we needed.

6.5 Developed software

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.

The "PLAY" option of the menu takes the user to the song selection menu, where songs are grouped in three difficulties: easy, medium and hard. In this menu, the user can scroll through the available songs in each group, and they can select which song to play.

As agreed in the MVP, one song has been developed for the video game, "Tinc un gos", that is classified as an easy song.

For the selected song, we have developed the five levels described before.

We have manually placed the notes in the screen, opening the MIDI file of the song to have a reference of the position of each note and adjusting them to correctly match with the song's real notes.

To change the tempo of the song (the speed at which the notes move), we have a script that takes a "tempo" parameter. The "tempo" parameter is used later to change the position of the notes at that speed. This is used to modify the speed of the notes from level one to level two.

For levels three (fig 17) 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.

We have developed a feedback screen (fig 18), where the user receives their final score after playing each level. The score is represented in stars and depends on the number of notes hit and the fails they have done.

In this feedback screen, the user can choose between:

“SELECTING A SONG” option, which takes you to the song selection menu; “PLAY AGAIN” option, that starts the same level again, and “NEXT LEVEL” option (if they got at least 1.5 out of 3 stars), that takes you to the next level. In case the user is in the last level of the song, the “NEXT LEVEL” would take them to the first level of the next available song. This functionality has not yet been implemented because there is only one song in the game.

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.

For all the levels, we added a pause button at the upper right corner of the screen, that allows the user to resume the level, restart it or go to the song selection menu.

After this had been implemented, we met with *Aula de So’s* music teachers to

Parlar de que s’ha fet tot el MVP d’aord amb les especificacions.

Secció del progress report II + test amb l’escola + implementació dels canvis.

Imatges de la APP al annex.

7 CONCLUSIONS

Parlar dels objectius del projecte.

Objectius del MVP. Completats/no.

% de objectius de projecte+MVP assolits.

Altres objectius.

7.1 Future work

Ens agradaria continuar treballant en la implementació d’altres aspectes del videojoc que no han entrat al MVP, com la creació d’un leaderboard, la possibilitat dels professors de controlar dinàmicament el contingut de la APP o la creació d’informes del progress dels alumnes.

AGRAÏMENTS

Agrair a la Escola d’enginyeria per la oportunitat de treballar en un projecte real per uns clients reals.

Al tutor per tota l’ajuda durant el projecte.

A Aula de So per l’ajuda i totes les facilitats que m’han donat.

BIBLIOGRAFIA

[1] Referència 1

[2] Referència 2

[3] Etc.

S’ha de treballar en reordenar les referències i la bibliografia pel contingut del report.

ANNEXE

Figures que em hagut de referenciar aquí pel seu tamany.

APÈNDIX

A1. SECCIÓ D’APÈNDIX

.....

.....

.....

.....

A2. SECCIÓ D’APÈNDIX

.....

.....

.....

.....