Master’s Degree in Informatics Engineering
Project
Final Report

Oratio’s Classroom

João Miguel Mota Paul
jmpaul@student.dei.uc.pt

Adviser:
Maria José Marcelino

July 10, 2014

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Acknowledgments

First and foremost, this report represents the end of a five year long academical path as a student of Informatics Engineering at the University of Coimbra which would not be possible without the constant help, and sometimes patience, of my parents to whom I owe most if not all I am today - a most special and loving thanks.

To Teresa Mendes for her guidance in the entire internship proposal and for its undeniable help in making this opportunity a reality.

To my advisor: Maria José Patrício Marcelino for the continuous preoccupation, encouragement and all the help provided during the internship.

To all the jury members for giving me precious feedback during the first semester. Hopefully, the same errors will not be repeated.

To Insignio Labs for creating this opportunity, to its CEO, João Ramalheiro, for its comprehension over the course of the internship and most of all to its employees that gladly helped me whenever I had any sort of need or issue.

To my ex-girlfriend which, even though we no longer remain together kept on giving me full support and encouragement over the time of this internship.

To my parents for everything that they did for me during my entire life and for making me the person I am today.

Last but not least, to a special group of friends which not only have been present for as long as I can remember but also for letting me speak up my frustrations, always bringing a smile to my face whenever I needed it the most - thank you.

Abstract

We live in a technological era, an epoch where once fictional ideas are now possible. While some things have changed over the years, the education system still relies on many outdated technologies such as blackboards and repetitive unrewarding exercises. In order to try to gradually improve and change this cycle, Insignio Labs took on the objective of developing "Oratio's Classroom", a multifaceted platform for teaching which makes full use of the current technologies.

The main focus of the software developed during this internship is to become a main part of Insignio Labs "Oratio's Classroom" full application. During the first semester, a module, similar to Windows paint but working as an overlay of other application, was developed allowing the user to signal, take notes and even draw on top of any of any of the applications developed by Insignio Labs. The capability to add notes is of major importance for the application to have educational uses and value as it creates the possibility to completely replace the blackboard in classes where Oratio's Classroom is present by creating digital notes which could easily be shared, viewed and changed at will.

During the second semester, an application with functionality similar to the known web page "Duolingo" was planned and developed. Using a design directed to children and many pedagogical techniques coupled with a fun game-play, this application has the objective of providing a fun and easier introduction to the world of music, more specifically, to the rhythmic part of it.

This report documents all the steps taken when developing this last application, both the ones regarding programming and choices made about the technologies, and the ones which encompass user interaction and design choices.

Keywords: Education, Gamification, Oratio, Unity3D, Classroom, C#, Codecademy, Duolingo, Rocksmith, Ribbon Hero, Insignio Labs, Flow, Music, Rhythm.

Contents

List of Tables

List of Figures

Chapter 1

Acronyms

Term	Meaning
OS	Operating System
IMMA	Instituto de Música Moderna e Artes
C#	C Sharp
GUI	Graphical user interface
MIDI	Musical Instrument Digital Interface
IDE	Integrated development environment
ACK	Acknowledgment

Table 1.1: Acronyms

Chapter 2

Glossary

Term	Meaning
Git	Distributed revision control and source code management system with an emphasis on speed, data integrity and support for distributed, non-linear workflows.
Unity3D	Unity is a cross-platform game engine with a built-in IDE developed by Unity Technologies.
Asset	Any component of a Unity project which can be placed in the scene is an asset. These can go from a texture to a full scale 3D model or even a script.
Rhythmiacs	Small creatures created by Insignio Labs with the intent to represent Rhythmic figures in Oratio's world.
application	In this report, when referring to the application it is meant as Oratio's Rhythmic Village as a whole and not to a single scene.
scene	Unity3D divides a project into scenes. In a scene exists assets which can use scripts to call methods creating the game logic.
module	Oratio's Classroom is divided in modules. The already existing ones are only Oratio's Flutemaster. Four more are in development. This internship contained work from two of these, the Oratio's notes, and in the second semester Oratio's Rhythmic Village.
object	In Unity3D all assets in scene are represented by an object which can be placed during development or instantiated through code.

Table 2.1: Glossary

Chapter 3

Introduction

This report, describes the work performed over a year long internship for Master in Informatics Engineering graduation, at Insignio Labs, a newly founded software and creative design company making its first steps in the market.

The work was supervised by Maria José Marcelino, PhD, professor at the Department of Informatics Engineering of the University of Coimbra.

3.1 Document Overview

- Introduction - This section presents an overview of the internship, including contextualization, motivation, goals and risks of the current project.
- State of the art - On this section it is described a list of products and services that are, in a way or another, related to the area of gamification.
- Requirements - This section describes the features supported by the final build of the application developed and the technologies used during the development and testing process.
- Architecture - This section describes the application architecture and provides an overview of the technical decisions and choices made during the project development.
- Project Planning - This section describes the software development methodology followed along with the initial and revised project planning.
- Software Quality - This section describes the tests made to assure the application quality and to make sure that all of the defined requirements were met.

- Conclusion - This section sums up the work done during the year-long internship at Insignio Labs, the lessons learned during its course, the contributions made and the outlined work for the future.
- References - This section lists the sources used as reference in the present report.

3.2 Context

The internship took place at Insignio Labs, a "creative software company specialized in the field of education, which seeks to establish a solid commitment to professionals in education and the entire school community. Our company presents 4 departments: Film, TV and Videogame Scoring, Design and Animation, Software Engineering and Classroom Contents' Development."[1] and it is part of the Oratio's Classroom project which aims to "enable teachers to transport their students into a virtual world by turning each lesson into a unique and cheerful Learning Experience using technology".[2].

3.3 Motivation

"Games have long been used to train concentration, to improve strategy, to learn human nature, and to understand how to think interactively and situationally."[3]

The gaming industry has evolved massively in recent years. However, the evolution it has taken is not exactly the most relevant to a gamified pedagogical application as it usually relies on realism instead of the state of mind which is a requirement for learning.

Since the beginning of video games as we know them in May of 1967 with the creation of Odyssey (The first game to use a raster-scan video display), the gaming subject has been a topic of great debate and turmoil. Amongst the endless debates about the benefits and drawbacks of playing video games, the hypotheses that they could be used for education has been slowly gaining strength and nowadays is considered by many as a possible future in the world of education. [4]

In an attempt to cement their position in this rapidly expanding area, multi-national corporations and entities, such as Microsoft[5] [6] and Best Buy[7], began to develop and fund several initiatives either to improve the usage of technologies in the classroom or to develop better tools for teaching and learning.

Due to the recent introduction of all these fundings, it seems almost wrong to think that this wasn't always an alternative, but before 2002 the gamification term did not even exist[8], and before 2010 it had no popularity at all as can be seen with the trend of searches in Google.

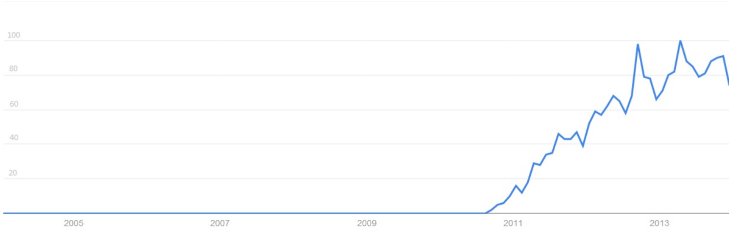


Figure 3.1: Popularity trend of "Gamification" [9]

However, even before 2002, computer based education was already thriving and was considered a booming market until around 2005 where "an explosion of new, often free technologies competing to entertain and teach children", appeared. [10]

Since then, hundreds (if not thousands) of technologies have emerged, creating a panoply of different approaches to educational software which, successfully or not, have changed the way many children learned subjects such as Math, English and Music.

Recently, taking courage from this boom, Insignio Labs current CEO, João Ramalheiro, fueled by his wish of creating a tool to help in his classes, began the development of a concept tool which tried to fulfill the needs he felt while teaching.

By making use of common tools such as Power-point and Paint, the first prototype, with which he participated in Microsoft's Partners in Learning European Forum held in Lisbon in 2012, was able to, although in an embryonic state, gained attention from the participants and also did not pass unnoticed by Microsoft, ending with a second place in the "Educator's Choice" category.

Later, and now with a team assembled, the same project was presented in Microsoft's Partners in Learning Global Forum which was held in Prague, where now, with an Alpha version of part of the project, it was able to gain a first place in the same category as before.

In the following months, a full business plan was defined and, the initially single application, began being developed in stand alone modules which could be sold individually to help fund the rest of the project.

While some of the modules are being commercialized as games, the main focus of all the applications are the gamification of a discipline or class.

During the development, a lot of thinking and prototyping was done in order to make sure of providing the best experience in several fields:

- Education - Not only the applications are thought as a way to learn while having fun, but they also have several tools for a teacher to use during the classroom, creating this way a more fluid and interesting class.
- Design - From the animations to the buttons, every single piece of design was carefully planned in order to provide a colorful and funny application with the intent of creating a world which the children may enjoy and interact with.
- Usability - All the modules and applications done by Insignio Labs are subject to classroom testing with over 200 children in order to make sure the interface is intuitive enough that even younger children can enjoy and use.

3.4 Psychological Value

The psychological effects of playing video-games are studied worldwide for their implication in the society. With the increasing amount of technology in schools, several researchers focused their time on discovering the true effects of this, ever more common, activity.

Edward Thorndike, an American Psychologist, developed in the early twentieth century a pack of rules usually referred to as the 3 laws of learning (readiness, exercise, and effect). this laws, generally applicable to the learning process, provide insight into what makes people learn most effectively. After the original three laws were defined, five additional principles have been added (primacy, recency, intensity, freedom and requirement).

This principles have commonly been presented as an explanation for why the use of games with a learning intuit can show incredible results. [11]. The reason for such relation is that the principles, present conditions which are very similar to many of the common design techniques used in games:

- **Exercise** - Common use is as a way to prolong game play. As usually said, practice makes perfect and, considering a game with any sort of challenge, the need for practice in order to surpass a level makes this a principle with great influence in the act of learning.
- **Readiness** - Greatly related to the concept of Flow (Further explained later in this report), this principle is mainly followed by the usage of techniques which reduce distractions (such as a simplistic design), a good balance of difficulty vs skill and a correlation between actions and feedback. This techniques end up increasing the motivation and positive feelings towards the activity and as such increasing the readiness to play more and with more enthusiasm.
- **Effect** - Much like readiness, this principle relies on the feelings of the user towards the activity and the relation between this feeling and the strength of the learning process.
- **Intensity** - flooring itself on the basis of emotions, immersion and engagement, this principle states that the impact created by an exercise can influence the appeal to the activity it represents. It also states that the realism can influence the learning process such as witnessing an action can be a better learning experience than reading about it. This creates an even greater need for a solid world to be created around the activities, be it real, or fictional.

Flow, is an psychology theory defined by Mihaly Csikszentmihalki as the "mental state of operation in which a person performing an activity is fully immersed in a feeling of energized focus, full involvement and enjoyment in the process of the activity".[12]

Due to the nature of most games, the audience they are aimed at is usually vast creating difficulties in designing and balancing something considered worthwhile for the entire audience.

Assuming now that the content and premise are inherently appealing to the audience, The design of a game would center not on achieving a flow state, but on keeping the player in the flow throughout its duration. To do this, the game would need to reflect the right balance of challenge and ability in order to keep players inside the Flow Zone. [13]

However, designing such a balance becomes a greater challenge as the size of the potential audience grows. By aiming at smaller audiences such as a specific age or group, the capability of keeping a player in a flow state would increase and hence, augment its capability to learn.

Other concepts exist to explain the effects of gamification such as the "Cognitive evaluation theory". This theory tries to explain the effects of external consequences on internal motivation. Not to be wrongly understood as rewarding the user, this theory focus on the idea that external events of any kind have a strong effect on the user, directly affecting his competence and autonomy, and hence, his motivation. These events do not need to be physical in any way and a simple choice to be made inside an application can give a sentiment of power and autonomy creating a pleasurable feeling for the software user. In fact, in 1995, Goudas, Biddle, Fox, and Underwood tested this hypothesis with the use of different teaching styles in a physical education class. The students reported higher levels of intrinsic motivation when their track-and-field instructor offered them a number of choices throughout the lesson rather than controlling every class decision.[14]

3.5 Goals

This project goals can be divided in two main parts.

- Internship
- Project

3.5.1 Internship

Considering the internship point of view, there are clear goals in consolidating the knowledge about Software engineering, game engines, gamification methods and development and also game planning and programing. There is also an obvious objective in gaining experience while developing software in a non-academical environment with real life clients with all the its implications and learn with the experience.

3.5.2 Project

Existent solutions for teaching using gamified exercises and/or pedagogical games present multiple limitations:

- Many are platform dependent and are not prepared for the multitude of technologies already in existence and being used in schools worldwide.
- Usually focus either on helping the teacher or in being exciting for the students. The capability to have the gaming aspect while retaining the pedagogical value is, in most cases, limited and ending up making the software a game where you can learn and not a game with whom you can teach.
- Try to do everything in a single application which usually makes the software not ideal for specific classes.
- Can be expensive and, in situation such as the previous case, won't have its potential used due to its magnitude.

In order for Oratio's Classroom to become successful, the above limitations must be solved. Moreover, it is necessary to develop a reason or motivation to keep playing after passing through each of the exercises.

During the second semester, a module, part of the Oratio's Classroom project was developed and its requirements are defined on chapter 3 of this document.

3.6 Risks

there are several reasons that can influence the outcome of a project. An initial analysis of the eventual risks and further periodical ones along the development can help mitigate unpredictable results.

The following list represent all the risks found which are worth of mention and the solutions found to overcome them:

- **Different Platform Releases**
  There is a wide variety of technologies and different devices, each one with its own hardware specifications and OS.
  - Complete mitigation of the risk is not attainable.
  - Solution: The usage of Unity3D as the game engine for the project greatly decreases the risk of the application not being supported by different OS's. The initial analysis of this risk allowed for the application to be thought considering slower devices.
    - Texture size was reduced to the maximum possible without losing quality.
    - No 3D models were used. The entire application animations are based in mathematical adjustments and sprite usage.
    - No use of static variables when possible to reduce memory usage between scenes.
    - Extreme caution when programming to avoid potentially expensive operations.
- **Microphone Usage**
  Part of the exercises existent in the application can be played by usage of a microphone as the input. In a classroom situation, it is not wrong to assume there will exist a lot of unwanted noise.
  - Complete mitigation of the risk is not attainable.
  - Solution: The assumption that when there is microphone input it is necessarily an input to be considered cannot be used in this situation. By limiting the inputs to be considered only when there is a sudden change (which may need to surpass a certain threshold) in the average noise can help reduce this risk. Furthermore, the limitation of inputs per second can further reduce the possibility of noise interfering with the exercise.
- **Repeatability**
  While not directly related to the development, for the application to attain his goals, a certain degree of repeatability must exist.
  - Complete mitigation of the risk is not attainable.
  - Solution: While not everyone will feel the solutions found are enough, while testing it was found that the application has

enough repeatability up to the point where all levels are completed with a score high enough to win a gold medal (over 90% of the maximum points). The posterior adding of achievements and on-line sharing can further optimize this.

Chapter 4

State of the Art

"If any market takes a finite amount of resources (however small) to function, complete markets would exhaust the resources of the universe."[15]

The education nowadays still follows some guidelines from the year 1918. In the industrial revolution, when education was made mandatory, the main objective of it was to create adults that worked well within the system and never to motivate kids. Scale was more important than quality, just as it was for most industrialists.

This system, still guides many schools nowadays where kids are told to sit in straight rows, listen to the teacher and obey instructions. This ended up creating, over the years, several generations of workers which are trained to do 1920-style labor.

The problem however goes even further than that mainly because many teachers still lack awareness of an important thing; That the needs changed.

Before the Internet became an almost standard asset in education, access to information was limited. The need for remembering all the information was critical for success and as such, education emphasized this by relying on standardized examinations and data memorization as a result.

The Internet however, over the years, changed our relationship with knowledge and making the need for memorizing data largely obsolete.

In an age where we have access to more data and information than ever before, this is no longer a productive way to prepare for the working world. With such a valuable tool giving almost unlimited access to data, it is more than obvious that data is not the valuable part; the processing is what matters.[16]

This does not mean in any way that access to information is the same as education. While many schools, colleges and universities wont be able to

brag about the size of their libraries and the amount of resources they can provide their students due to this globalization of data, this also means that teachers are gaining an even more important role in the formation of the youth. Not as the owners, and passers, of the knowledge but as the guides for this new generation's way of thinking.
In order to do so, a new path must be carved in education. One not following the old, industrial age based, one not following methods that rely on memorizing books but one where creativity, awareness, commitment, improvising and most of all innovating are rewarded.

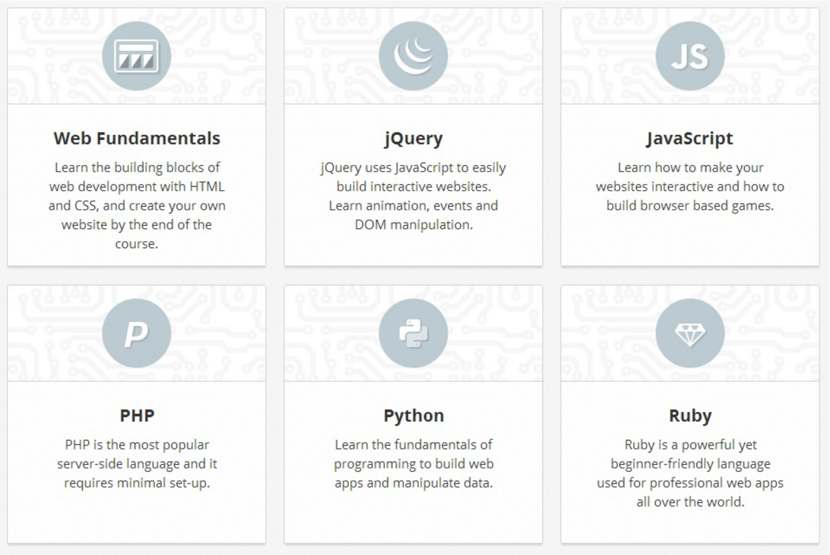
In order to take advantage if this mentality and the need for teaching tools that arises with it, many companies began to develop several pieces of software which tried to fulfill that need and over the years, the concept of gamification began to develop and cement its position in the market.

Because of the nature of the project, a common comparison was not done as the main goal of the development was in the gamification aspect of it and the capability to be used in the classroom which most of the softwares currently in the market are not prepared for. Considering that, a comparison would lack metrics which were both comparable and useful.

Some of the most prominent examples of software which directly or indirectly work as a pedagogical tool using gamification are described in the next sections of this chapter along with their relation to the project developed during the later semester of this internship.

4.1 Codecademy

The years where computer programming were known only to a small handful of people are in the past and each day that passes further in the past. Due to the spread of Internet connections in the nineties and the proliferation of personal computers even before it, the art of programming has now become in reach to anyone with some time and motivation. In order to help those, some websites such as `www.codecademy.com` began to amass great amounts on information about several programming languages in a single place.



Web Fundamentals
Learn the building blocks of web development with HTML and CSS, and create your own website by the end of the course.

jQuery
jQuery uses JavaScript to easily build interactive websites. Learn animation, events and DOM manipulation.

JavaScript
Learn how to make your websites interactive and how to build browser based games.

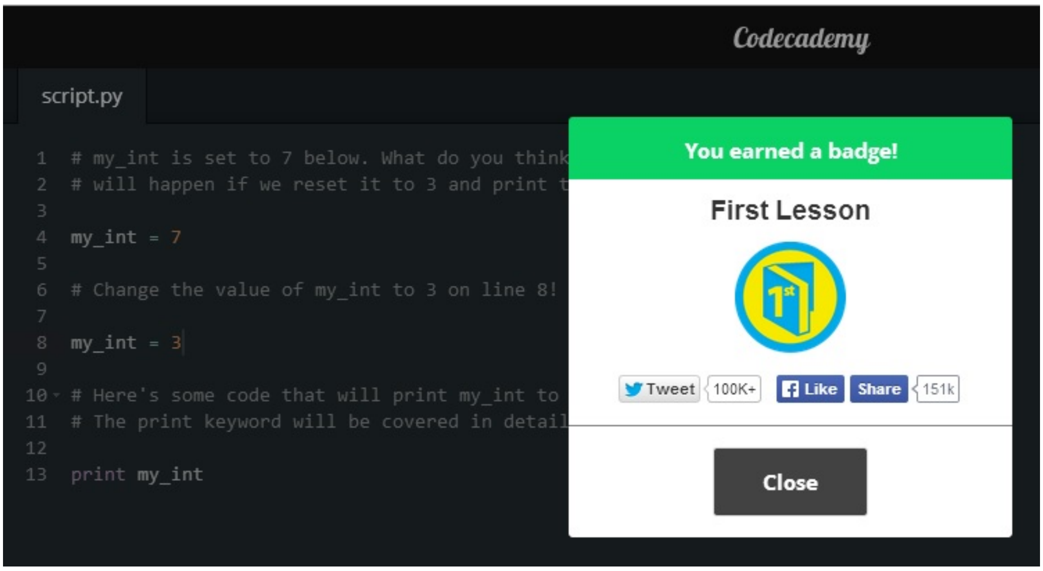
PHP
PHP is the most popular server-side language and it requires minimal set-up.

Python
Learn the fundamentals of programming to build web apps and manipulate data.

Ruby
Ruby is a powerful yet beginner-friendly language used for professional web apps all over the world.

Figure 4.1: Languages in codecademy

This webpage, in order to further encourage users to learn, uses a system of achievement badges and leveling in which, the user gets the badges by reaching certain milestones such as finishing a difficulty level on a language and will get experience by completing any exercises.



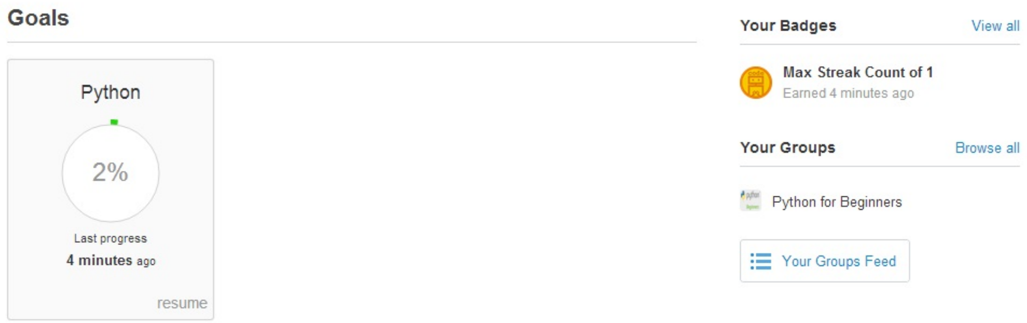
```
# my_int is set to 7 below. What do you think
# will happen if we reset it to 3 and print t

my_int = 7

# Change the value of my_int to 3 on line 8!

my_int = 3

# Here's some code that will print my_int to
# The print keyword will be covered in detail

print my_int
```

You earned a badge!

First Lesson

Figure 4.2: Badge system in codecademy

This extra however, while motivational by themselves, have an incredibly higher value when considering that Codecademy also implements an entire social network for the users.



Goals

Python

2%

Last progress
4 minutes ago

Your Badges

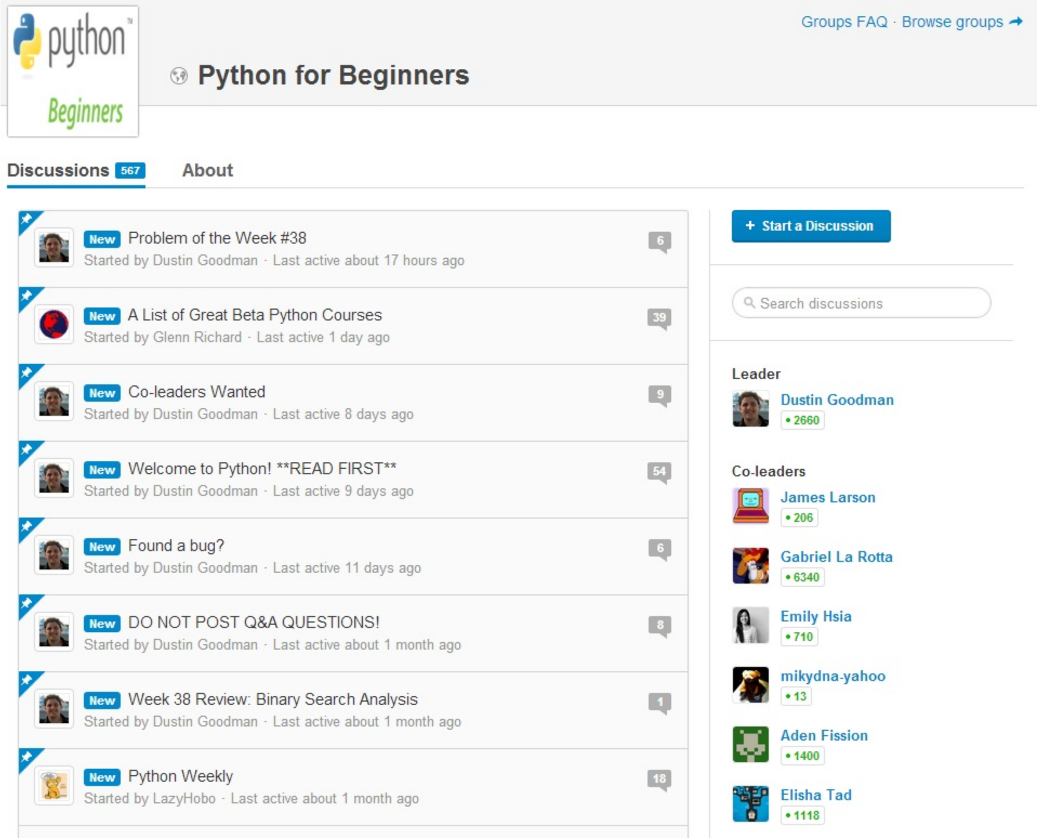
Max Streak Count of 1
Earned 4 minutes ago

Your Groups

Python for Beginners

Figure 4.3: Codecademy profile

Where an user can create its own profile, join groups dedicated to certain languages(or aspects of one), show his medals and achievements and talk to other members of the community.



Groups FAQ · Browse groups

Python for Beginners

Discussions 567 About

New Problem of the Week #38
Started by Dustin Goodman · Last active about 17 hours ago 6

New A List of Great Beta Python Courses
Started by Glenn Richard · Last active 1 day ago 39

New Co-leaders Wanted
Started by Dustin Goodman · Last active 8 days ago 9

New Welcome to Python! **READ FIRST**
Started by Dustin Goodman · Last active 9 days ago 54

New Found a bug?
Started by Dustin Goodman · Last active 11 days ago 6

New DO NOT POST Q&A QUESTIONS!
Started by Dustin Goodman · Last active about 1 month ago 8

New Week 38 Review: Binary Search Analysis
Started by Dustin Goodman · Last active about 1 month ago 1

New Python Weekly
Started by LazyHobo · Last active about 1 month ago 18

+ Start a Discussion

Search discussions

Leader

Dustin Goodman • 2660

Co-leaders

James Larson • 206

Gabriel La Rotta • 6340

Emily Hsia • 710

mikydna-yahoo • 13

Aden Fission • 1400

Elisha Tad • 1118

Figure 4.4: Codecademy groups

Much to the liking of Oratio's Rhythmic Village, this software relies on simple incentives like medals to increase the likelihood of a person to keep using the software. This is a clear attempt at grabbing the attention of the achiever type of users. While not in the project plan, Oratio's Rhythmic Village, as well as Codecademy, have a social component which can become the feature that allows the socializer type of player to actually enjoy the application after the initial period of use.

4.2 Duolingo

When talking about education blended with gamification, one cannot miss the chance to talk about Duolingo. This tool, from the developers of the known CAPTCHA and reCAPTCHA, developed to teach new languages to its users, created a gamification plan to its courses in which every single action a user performs correctly, be it a lesson, a translation or a skill strengthened, will give the user points towards a user skill level.

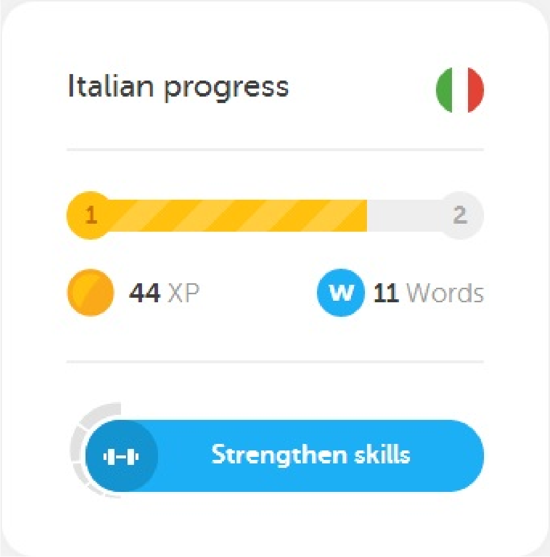
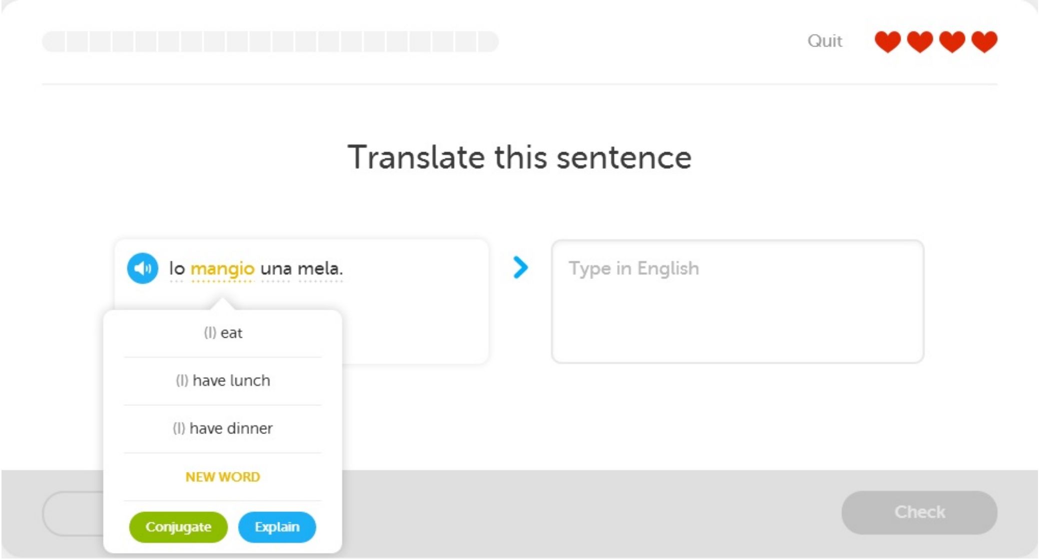


Figure 4.5: Duolingo level progress

This courses, while having the same objective, use different methods to not only teach but also keep the user entertained. Among the different exercises we can find matching images to the words they present, writing what we hear, speaking a translation with the aid of a microphone or the most common "translate this sentence" exercises.

Any word can also be hovered with the mouse to display a tooltip which will allow the user to see the meanings of the word as well as to load a explanation or even a table with a verb conjugation.



Translate this sentence

Io mangio una mela.

Figure 4.6: Duolingo exercise example

Duolingo also implements a full social experience by letting its users find friends from other social networks and sorting them through leaderboards:

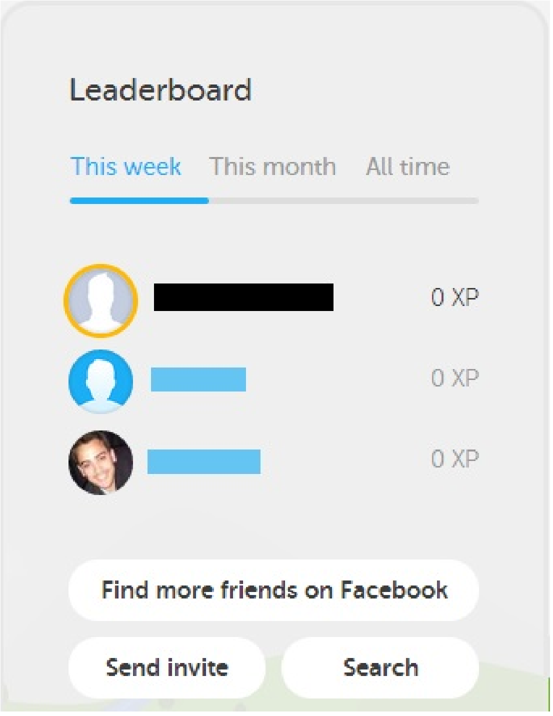
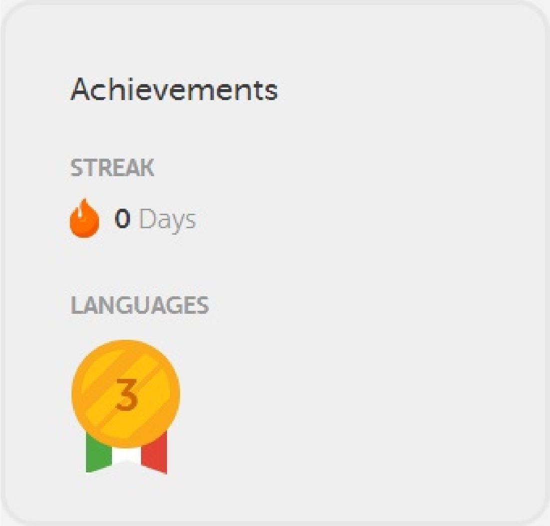


Figure 4.7: Duolingo friends leaderboards

Checking their achievements:



Achievements

STREAK

0 Days

LANGUAGES

Figure 4.8: Duolingo friend achievements

And even send messages:

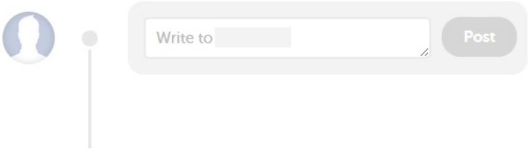


Figure 4.9: Duolingo friend messages

The way Duolingo implemented gamification, apart from the leaderboards and level up mechanic, focus not only in the exercises but also on the division of the various themes of each language.

Each of the languages was divided in several smaller topics which are considered as arts of a level (composed of several of these parts).

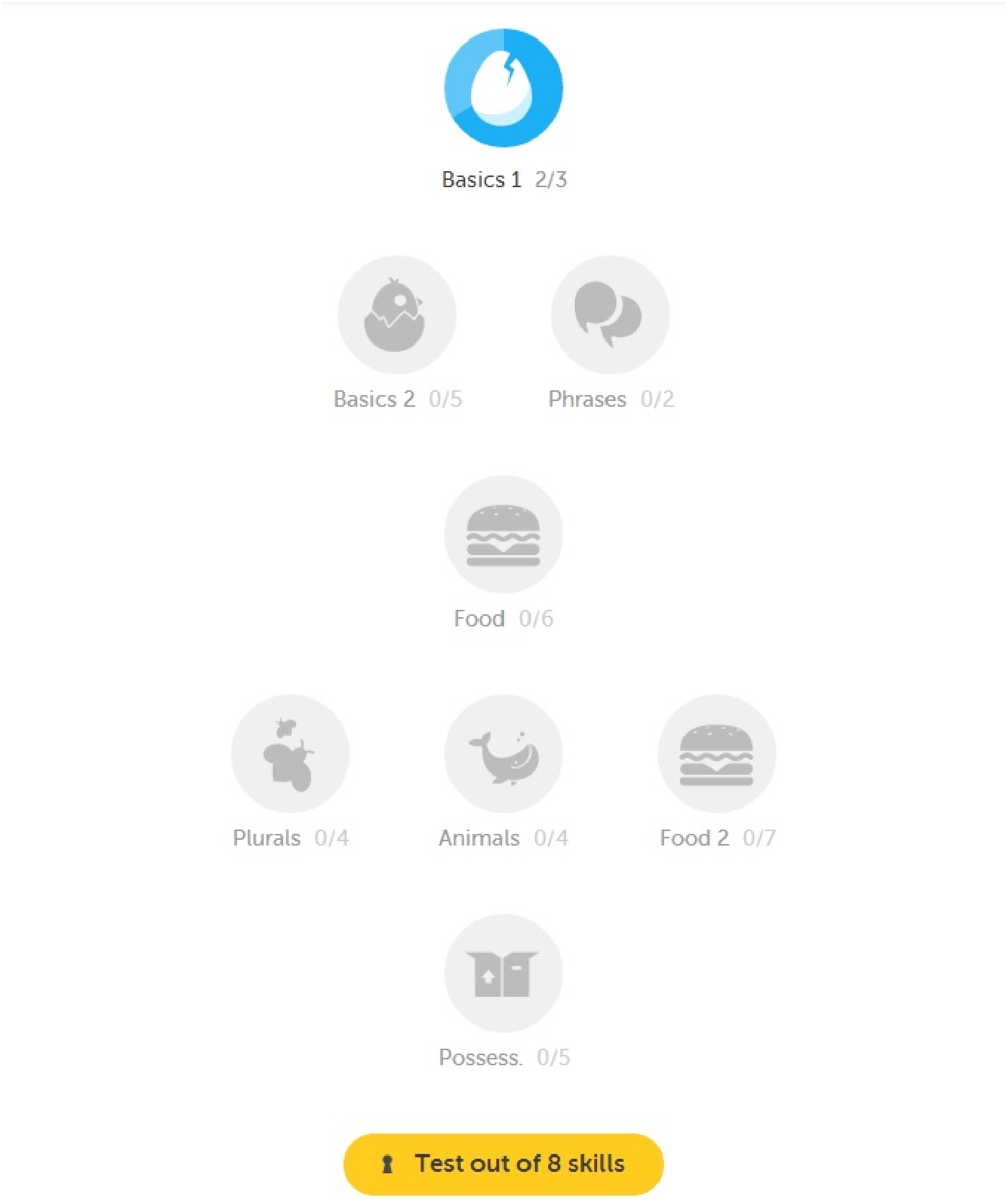
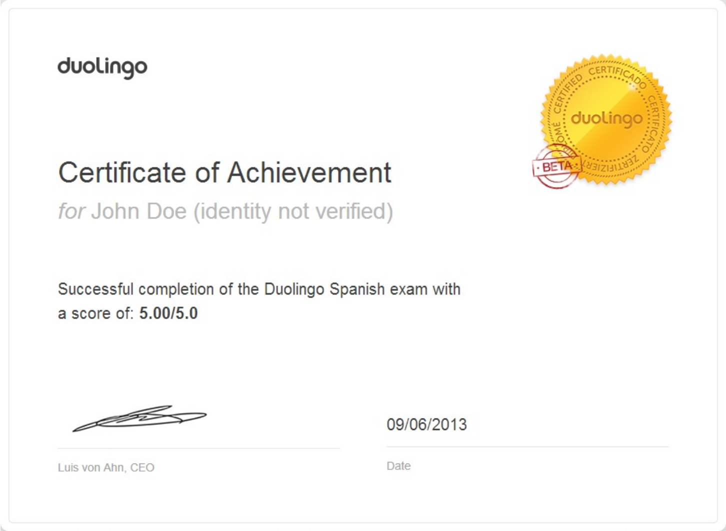


Figure 4.10: Duolingo levels tree

After each level an user can take part in a test which rewards the user with a new level if it passes.

Duolingo also offers a currency called "Lingots" which can be spent to gain access to new usage modes such as "timed practice", freeze the daily usage streak for a day, or to take a test in which the user is rewarded with a Duolingo certificate if it passes.



duolingo

Certificate of Achievement

for John Doe (identity not verified)

Successful completion of the Duolingo Spanish exam with a score of: **5.00/5.0**

Luis von Ahn, CEO

09/06/2013

Date

Figure 4.11: Duolingo certificate

Duolingo was so successful in its implementation that a study was done about its efficiency. That study concluded that the learning speed of an English speaker trying to develop Spanish skills would be faster with Duolingo than with current University methods[17].

While such a study may lack a bigger user sample and a deeper analysis of the different courses instead of only Spanish for English speakers, the results, and hence the conclusion, were not only valid but also surprising.

Similarly to Oratio's Rhythmic Village, Duolingo spreads the various different tasks which comprise the learning process into several smaller tasks with a great gamification component creating this way a more fluid experience which helps in reaching a Flow state. Both of these applications rely heavily on mechanics relying on the attention of the user creating a gap of skill between whoever is really into the game or not, hence reinforcing the importance of the previously referred state of mind.

Duolingo also allows users to repeat previously completed levels in order to remember previous lessons or to increase the scores attained. This ends up being the same train of thought applied in Oratio's Rhythmic Village and acts as an important asset in keeping player motivated allowing them to repeat levels they did not perform well or to train in easier levels before tackling the more difficult ones.

4.3 Ribbon Hero

Using a name reminiscent of the known franchise "Guitar Hero", Microsoft presented its own gamified teaching platform called Ribbon Hero. This software, which works as an overlay module of the most common Microsoft Office tools provides a learning experience for this applications where the users must perform several tasks to gain points.

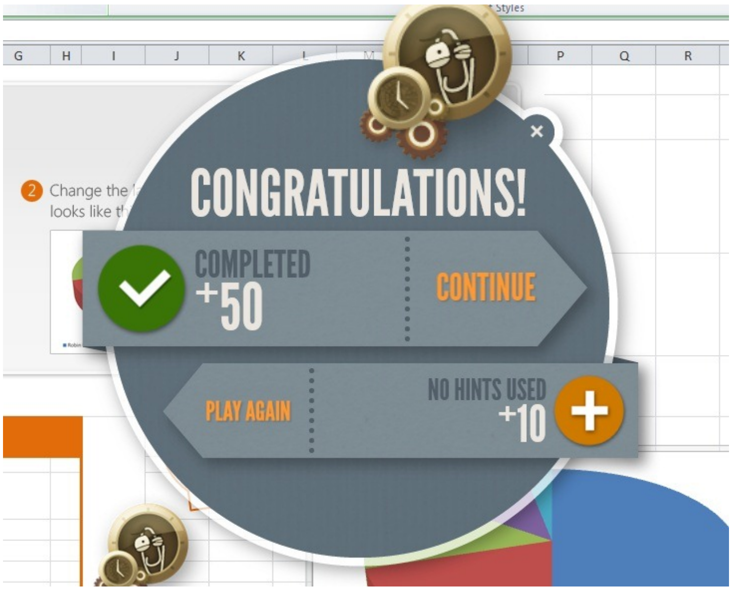


Figure 4.12: Ribbon Hero completed level screen

This tasks could consist in editing documents, formating, inserting graphs or pictures or many others.

While mechanically different from Oratio's Rhythmic Village, this platform shares many characteristics with it such as social integration through Facebook, ordered levels with increasing difficulty and distinct scores, different types of exercises in order to increase variety and also tries to create a fantasy world in attempting to achieve a better user experience.

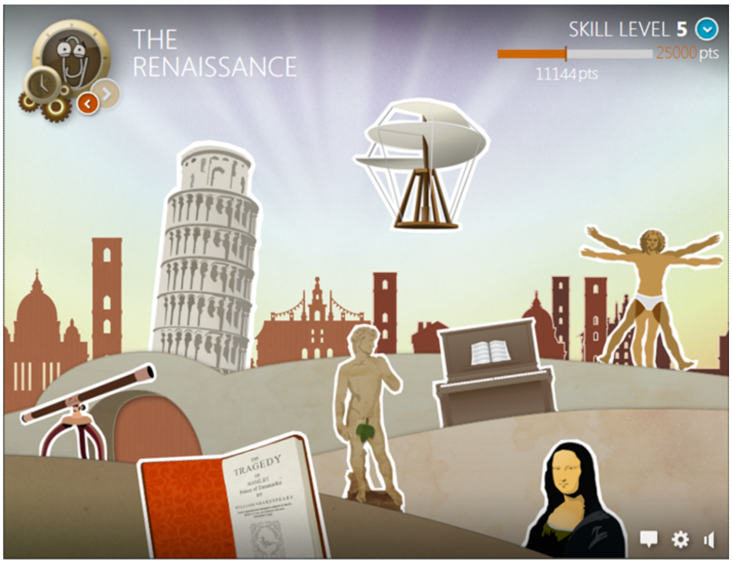


Figure 4.13: Ribbon Hero game screen

A global skill level acts as a meter of the player progression through the game and also as an incentive to play more in order to reach the highest level. Oratio's Rhythmic Village applies a concept similar to this one by dividing its levels in "chapters" where in each one a new type of rhythmic figure is unlocked, again, with the intent of prolonging the time spent by the user playing.

4.4 RockSmith

No state of the art about education aided by technology, gamification or even games by itself should be completed without referring RockSmith. RockSmith, is a software which plays similarly to Guitar Hero but instead of giving the player a toy guitar with 5 colored buttons, it uses a real electric Guitar or Bass connected directly to the device.

The game area is simple, showing a lane split into color coded strings and frets (the finger positions) starting at one on the left side of the screen. Like the fretboard itself, there are dots to help guide your fingers to the right place much like in most guitars.

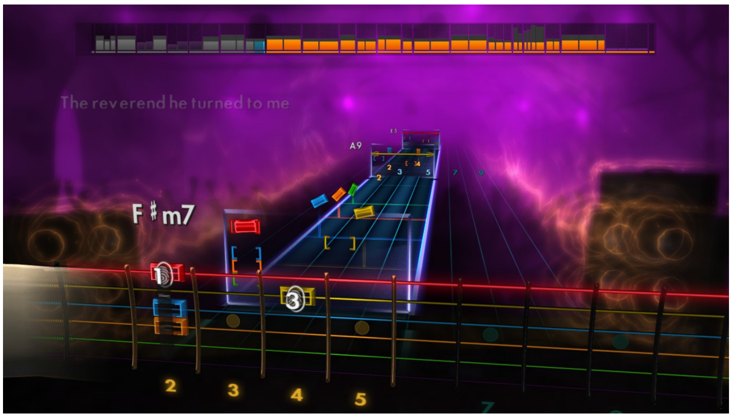
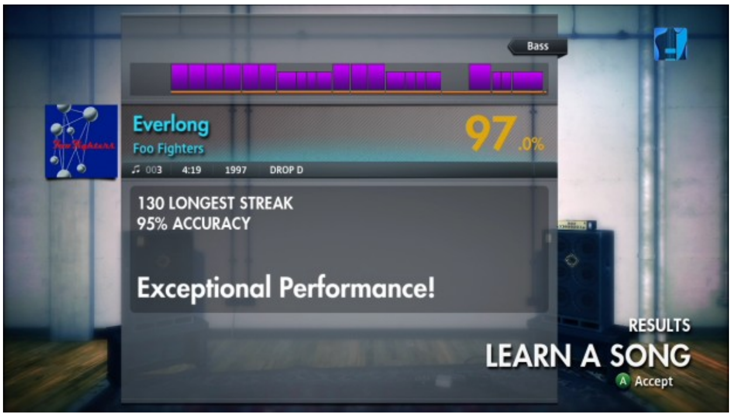


Figure 4.14: Rocksmith gameplay

This game is somewhat different from the most common ones as the player does not choose the difficulty. Difficulty in this game is based upon two variables, the song chosen as there are different degrees of difficulty in the song list and also in a variable called mastery. Mastery is song dependent and always starts at zero while gradually going up along with the skill of the player. The game also prepares the user for later difficulties over time, as can be seen in the previous image, there are even times when an user need only to play a single note but the game already shows that the note is part of a chord giving the opportunity for the player to correctly place the fingers even if for the current difficulty it was not required.

By providing the user with scores not based on the amount of notes (contrary to games like Rock Band), the player is given a sense of accomplishment by winning even if the difficulty set would only show him a tenth of the real notes.



Everlong
Foo Fighters
97 .0%
003 4:19 1997 DROP D

130 LONGEST STREAK
95% ACCURACY

Exceptional Performance!

Figure 4.15: Rocksmith scores menu

In order to provide an easier and more fluid gameplay, a decent amount of mini-games were included which, much like the exercises in Oratio's Rhythmic Village focus on a certain part of the whole creating a better environment to train and develop certain skills.

By providing local and on-line multiplayer, RockSmith also allow for a great degree of social interaction. Allowing two players to play together while the game manages the difficulty differently for each one. allowing for a much better gameplay and hence an increase in the learning value. While not in the planning, Insignio Labs expects to create multiplayer capabilities in Oratio's Rhythmic Village.

Chapter 5

Requirements

5.1 Introduction

"Without requirements or design, programming is the art of adding bugs to an empty text file."[18]

The first step in a software development project is the definition of its requirements. These are to be used as a map or guideline for the development of the project.

This requirements were defined right from the beginning by considering the usual user and the needs the software would try to fulfill. Over the course of the development, part of the requirements were changed, new ones were added and some were forfeited. All these choices were made after carefully analyzing each situation in meetings done for that sole reason.

5.2 Oratio's Rhythmic Village

Oratio's Rhythmic Village is the name given to the module developed during the internship. It is part of Insignio Labs greater project, Oratio's Classroom, and its objective is providing a learning platform for the rhythmic portion of musical education.

5.2.1 Functional Requirements

These requirements are descriptive of the features the application should have and correspond to an analysis made by considering the possible use-cases of the application.

They can be divided into the following groups:

- Profile Authentication
- Game
- Global
- Randomness
- Scoring

5.2.1.1 Profile Authentication

Some of the functionalities present on the application can only be accessed after a valid profile is loaded or created.

Name	Requirement
Get Profiles	The application must be able to read all the valid profiles existent in the default profiles directory.
Load Profile	The application must be able to load the currently selected profile.
Save Profile	The application must be able to Save new Scores, game completion and options to the correct profile when prompted to do so.
Create Profile	The application must be able to create new profiles at will as long as there isn't another profile with the same name.
Delete Profile	The application must be able to delete a profile following an user action mandating it to do so.

Table 5.1: Authentication requirements

5.2.1.2 Game

The application is comprised of 5 different kinds of exercises. All of these must act independent of one another while utilizing the same profile and options.

Name	Requirement
Load Exercise	Disregarding the type of exercise, the application should load the expected assets and rules receiving only a midi file. (Exceptions are made for the exercise not related to a sound file.)
Game Mechanics	The application should, regardless of the exercise, be able to mark the user inputs as wrongs or rights and act accordingly.
Ingame Sound	Every valid user input should be met with the correct sound FX.
Continuity	Once an exercise is finished, the application must update it's status so that the next level can be loaded right away.
Ending	The application must be able to have one of the exercises being terminated at any time. In the case of an exercise being finished, the ending should be different depending on the outcome (correct or wrong).

Table 5.2: Game requirements

5.2.1.3 Global

Some of the capabilities of the application should function regardless of the scene and situation. These are requirements which are usually solved by the existence of an object which will not be destroyed in each scene load.

Name	Requirement
Transitions	Transitioning from one scene to another must be done seamlessly. The sound should not stop and the loading times should be unnoticeable.
Theme	The theme music in the application should be controlled in a way that it can stop and resume depending on the scene. When stopping and resuming, the volume should change overtime instead of being changed instantly.
Ratio Handler	The application should be able to handle any kind of resolution and even changes of it during gameplay should be handled even though the application does not allow changes after starting. (This was requested by taking in consideration the full project implications.)

Table 5.3: Global requirements

5.2.1.4 Randomness

Some parts of the application rely on a degree of randomness. Exercises can have its difficulty increased by not having the exact same choices every time they are played and some design and sound randomness can help in creating a more credible and interesting world.

Name	Requirement
Randomization Variation	Any randomization mechanism in the application must create a unique value/object from the ones previously generated in the scene.
Randomize Sounds	The in-game audio representing the Rhythmiacs voices should be chosen randomly from a pool of possible sounds.
Randomize Exercises	Some exercises rely on choices or shuffles. These should be done randomly by the application providing an extra layer or re-playability.
Randomize Design	Some of the design elements of the application must be defined randomly. These include: • Clouds in the background shape, speed and position • Totems in the level selection screen shape and rotation • Eyes movement in all Rhythmiacs

Table 5.4: Randomness requirements

5.2.1.5 Scoring

Being a game, a scoring system is needed for the application. This scoring should only apply to the exercises in which it is relevant.

Name	Requirement
Score Counter	In the exercises that allow it, a score should be updated based on the user input.
High Score	The application should be able to save the highest score attained in each exercise. This creates a possibility for re-playability
Rank System	In the exercises that allow scoring, a rank should be awarded based on the proportion between the highest score and the maximum possible score. This should change the aspect of the totem representative of the level in question.)
Scoring Not Possible	In exercises such as the multiple choice, it was decided that scoring would be disabled. The application however should still save information about if the exercise was completed or not successfully.
Scores By Profile	Each profile must have its own high scores completely independent of the other profiles.

Table 5.5: Scoring requirements

5.2.2 Quality Requirements

This section focus on the quality requirements for the application. These exist to make sure system constraints and properties are taken in consideration. They are grouped in the following categories:

- Performance
- Compatibility
- Reliability
- Robustness

5.2.2.1 Performance

This section covers quality requirements related with performance.

Name	Requirement
Method-Response-Time	Every method which rely on some degree of randomness (and considering the need for it to be different from the rest) must have a limit of the time it takes to return its value. In case this time is spent a new way to generate the value/object must be considered by the application even if it means reducing the degree of randomization.
Computational Cost	The application should be able to run in most of the devices on the market without great performance variations.

Table 5.6: Performance requirements

5.2.2.2 Compatibility

This section covers quality requirements regarding OS compatibility.

Name	Requirement
OS Target	The application being developed first for the Windows OS ust be compatible with all the latest Windows versions (Vista, 7, 8 and 8.1)

Table 5.7: Compatibility requirements

5.2.2.3 Reliability

This section covers quality requirements related with reliability.

Name	Requirement
Check Save File	The application must be able to check if the save is not corrupted in any way to prevent malfunctioning.
Activity Flow And User Actions	The activity flow and the user actions should remain constant across the entire application.

Table 5.8: Reliability requirements

5.2.2.4 Robustness

This section covers quality requirements related with robustness.

Name	Requirement
Fault-Tolerant	The application should be completely fault tolerant. If by any reason or event some of the data is invalid, this should be handled by the application without any impact on the user experience.

Table 5.9: Robustness requirements

5.3 Tools and Technologies

The following sections describe the technologies used during the project development.

5.3.1 Unity3D

The game engine, previously used by Insignio Labs and the one expected to be used was Unity3D, an engine best known for the ease of portability of its builds and its easy to approach free license. Along with Unity3D, a programming language was chosen from the three possibilities allowed. C#, UnityScript and Boo, and several Add-ons were also used to ease the usage of Unity3D.

5.3.1.1 C#

C# was the language chosen to develop with for 2 main reasons. the first one was the previous interaction and current familiarity with Java, another programming language similar in syntax to C#. Secondly, C# was the language previously used in other applications by Insignio Labs. Choosing C# would only make it easier to later integrate each of the softwares into the full Oratio's Classroom project.

Furthermore, and now considering the possibility of using more than one language, it was found 2 reasons to stay with a single one. A single language allows for an easier organization and although some capabilities were in one of the languages and not on the others, there are usually workarounds making the usage of a second language more irrelevant than if there was functionalities unique to each of them.

5.3.1.2 NGui

NGui is an Unity3D Add-on which attempts to fully replace the Unity3D native GUI. Due to a later discovery of this tool most of the project GUI was done from the start, without using the native GUI system due to several limitations. After the discovery of NGui most of the project was already finished and as such only some parts of the software use this Add-on.

5.3.1.3 Visualizer Studio

Visualizer Studio, is another Unity3D Add-on that is usually used to procedurally generate worlds based on audio. Some of its functions however were proven indispensable in creating the functionality which allows some of the exercises to be played with the input from the microphone.

5.3.1.4 NAudio

NAudio is a free-source library compatible with Unity that allows for great control over most audio requirements. In this poject, the capability to load and save MIDI files was the most important characteristic.

5.3.1.5 ITween

ITween is a library tailor-made for Unity which allows dozens of tweens to be done to a game object. a Tween is a series of states created by interpolation. These states can be positions, volume values, scales or many other different attributes.

5.3.2 Adobe Photoshop

While all the design work was done by Insignio Labs design team, some times, simple modifications like a resize of an asset were needed. Instead of passing through the entire process of asking for the change, waiting for it to be done and then receiving the new asset, it was easier to just do the change without asking the team. This was specially helpful in the later stages of the development when trying to reduce the memory usage and required disk space in order to allow the software to run in slower devices.

5.3.3 Microsoft Visual Studio 2013

While Unity allows a build to be created for all Windows versions, In order for an application to work with the new Metro system from Windows 8 and 8.1, the usage of Visual Studio was a must. By building the solution in Unity and later creating the app package in Visual studio, all the functionalities existent in the Metro system were usable which includes the possibility to later add in-app purchases and to publish the application in the Windows 8 store.

5.3.4 Dropbox

Dropbox, even though it is not a development tool, proved invaluable as a way to easily share assets between all the Insignio Labs team members.

5.3.5 Audacity

Audacity was a tool used mainly for audio prototyping and correcting slight audio assets issues, much like Photoshop was for the design assets.

5.3.6 TeXstudio

TeXstudio, along with MiKTeX, a free LaTeX implementation, was the tool of choice for writing the reports for this internship allowing greater/easier control over formatting than the usual word processors.

Chapter 6

Product Vision

6.1 Introduction

This document was done with the sole intent of further explaining the functioning and the aesthetic of the application developed during the second semester of the internship. In order to understand the application it is necessary to know that what was developed was not the final version of the application. The software in question is but a small part of a larger software called Oratio's Musical Adventure which, in turn, is a single module of a gargantuan project entitled Oratio's Classroom. Oratio being the mascot and main character of this entire franchise.

That being said, Oratio's Rhythmic Village is by no means a small application. Counting 5 different types of exercises, randomized gameplay, fully animated and completely hand drawn, Oratio's Rhythmic Village finds itself being given the last touches before the commercialization.

6.2 Overview

This application is comprised of 6 main scenes which is where most of the gameplay time will be spent. These are the level selection menu and the 5 different exercises. There is also a few more scenes such as tutorials for all the exercises, 2 different cut-scenes which represent the Introduction to Oratio's Musical Adventure and Oratio's arrival in the Rhythmic Village.

In the final application, Oratio will be able to travel between the Rhythmic Village, the Melody City (still not developed by Insignio Labs) and the Magical Composer (a musical editor already being developed by Insignio Labs which will enable the users to create their own exercises and music).

Also to notice that the names used in this document are not final and only used inside Insignio Labs as the final ones were not decided at the time of this document's writing.

6.3 Scenes

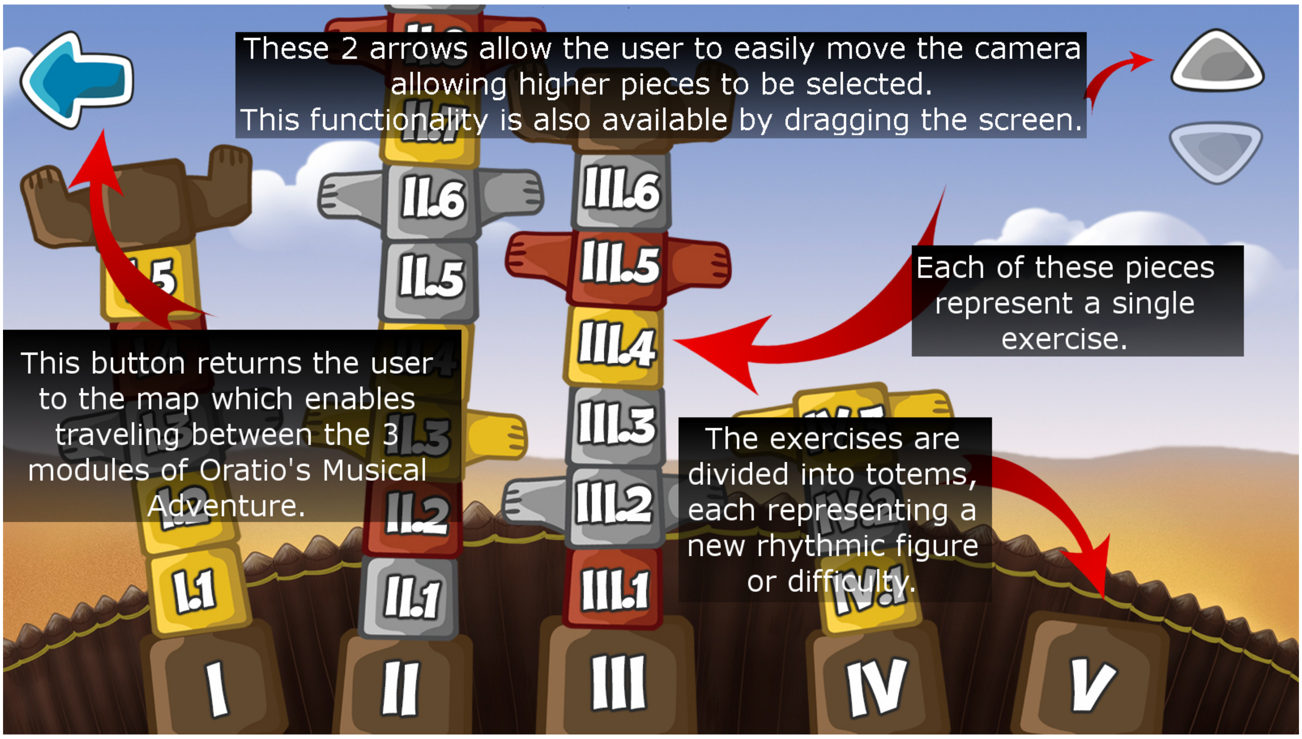
This section thoroughly explains the elements of each of the scenes in the application and the functioning of it. The final design (which was not completely integrated with the application at the time) is also shown. Again, the application developed is still unfinished and suffering changes (mainly in the design department) and what is shown here might suffer changes in the future.

6.3.1 Level Selection

The level selection scene was designed thinking mostly in the native American totem poles usually comprised of several levels. Likewise, each of the pieces of a totem represents a single exercise. by completing the exercise, different colors will unlock in the totem representing how well did the user performed. These colors include bronze, silver and gold as well as a wooden totem for the exercises still not completed.

A single button allows the user to return to the previous scene where a map allowing to travel between the modules exists.

To avoid future problems when adding new exercises, 2 arrows allow the user to drag the scene up and down making higher totems visible. These are of course disabled when not needed. This functionality can also be used by simply dragging the mouse in any pat of the screen.



These 2 arrows allow the user to easily move the camera allowing higher pieces to be selected.
This functionality is also available by dragging the screen.

Each of these pieces represent a single exercise.

This button returns the user to the map which enables traveling between the 3 modules of Oratio's Musical Adventure.

The exercises are divided into totems, each representing a new rhythmic figure or difficulty.

Figure 6.1: Level Selection Menu Explained

6.3.1.1 post-internship

After the internship, this scene, mainly due to the feedback received from the tester as well as many limitations identified, is suffering a major overhaul which will allow the users to travel between the levels without the need for any sort of screen dragging. This will be done by redoing the totems so that each one opens a single window with all the possible levels. This will also allow for an easier identification of the different types of exercises.

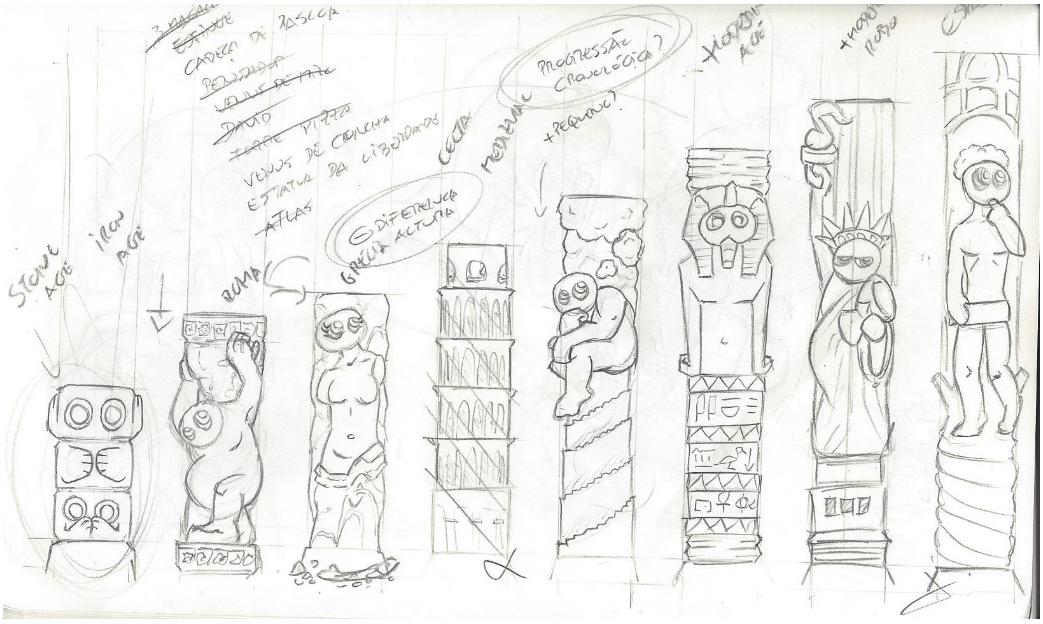


Figure 6.2: Level Selection Remake First Ideas

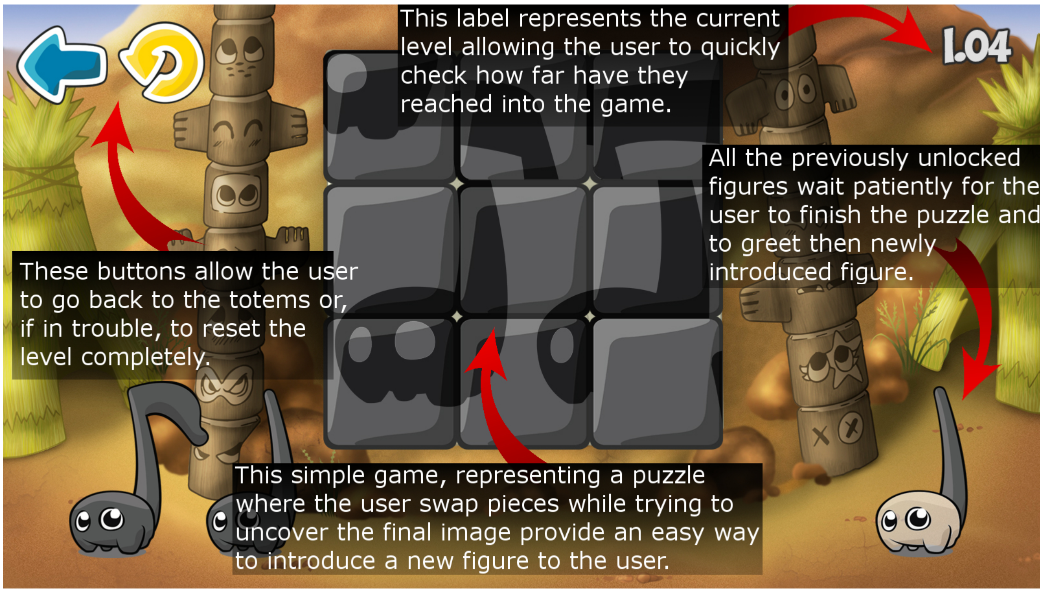
6.3.2 Exercises

In this section, a small explanation on how each exercise work will be provided along with an image representing the exercise and when possible the changes expected to happen post-internship.

6.3.2.1 Swap Puzzle

While not an actual exercise, this small scene with a simple puzzle allows for the user to make the first contact with the Rhythmiacs and each different rhythmic figure.

After the puzzle is completed, the Rhythmiac representing the rhythmic figure in the puzzle appears granting the opportunity to hear him further exploring these tiny creatures rhythmic capabilities.



This label represents the current level allowing the user to quickly check how far have they reached into the game.

All the previously unlocked figures wait patiently for the user to finish the puzzle and to greet then newly introduced figure.

These buttons allow the user to go back to the totems or, if in trouble, to reset the level completely.

This simple game, representing a puzzle where the user swap pieces while trying to uncover the final image provide an easy way to introduce a new figure to the user.

Figure 6.3: Swap Puzzle Explained

The design was later re-done while keeping the entire functionality the same.

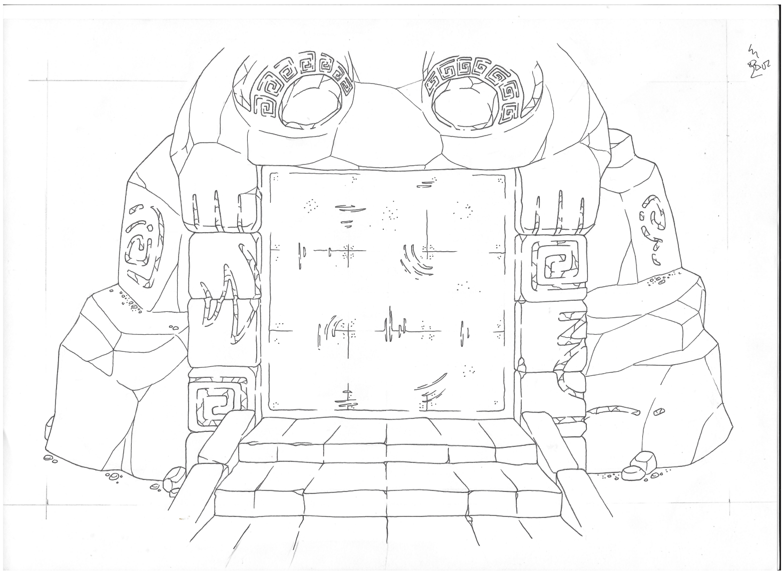


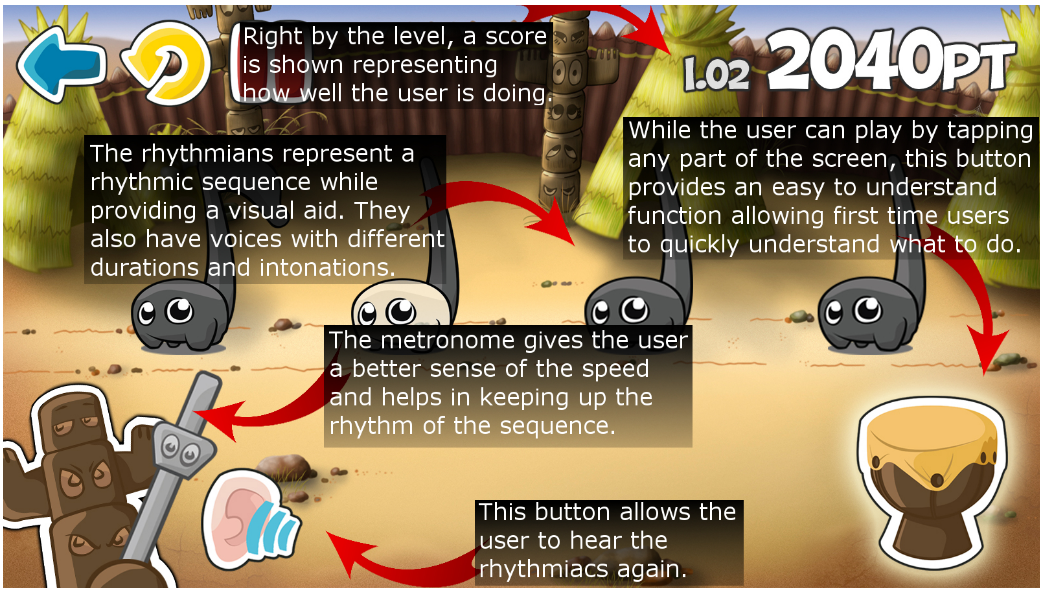
Figure 6.4: Swap Puzzle New Design

6.3.2.2 Drum Master

This exercise makes use of a visual and audio aid to explain the basics of the rhythm. By using a sequence of Rhythmiacs and playing each of their sounds at the same time of the metronome the rhythm composed becomes easily distinguishable.

After this aid, the user must input the same sequence by tapping the mouse at the correct time. This exercise also allows the user to use the microphone instead of the mouse which allows for classroom use.

This is the easiest exercise and also the first one to become unlocked and the main focus is to train the perception of the duration of each rhythmic figure.



Right by the level, a score is shown representing how well the user is doing.

The rhythmians represent a rhythmic sequence while providing a visual aid. They also have voices with different durations and intonations.

While the user can play by tapping any part of the screen, this button provides an easy to understand function allowing first time users to quickly understand what to do.

The metronome gives the user a better sense of the speed and helps in keeping up the rhythm of the sequence.

This button allows the user to hear the rhythmiacs again.

Figure 6.5: Drum Master Explained

This exercise was also restructured in terms of design while keeping the same functionality. Below is the final scenario (while still missing the Rhythmiacs and the entire GUI).

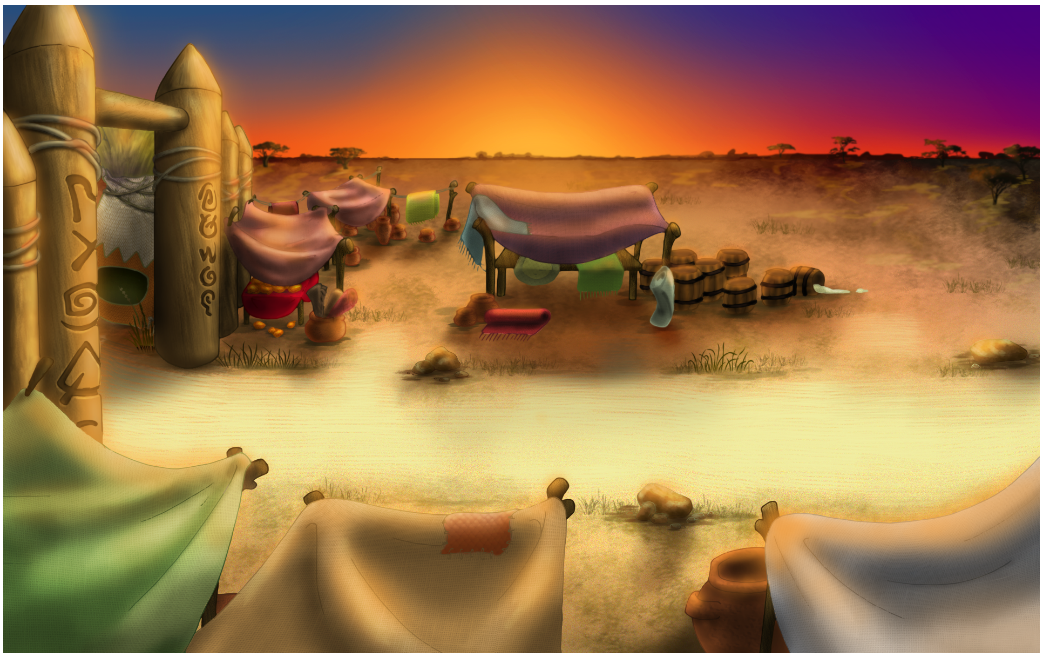
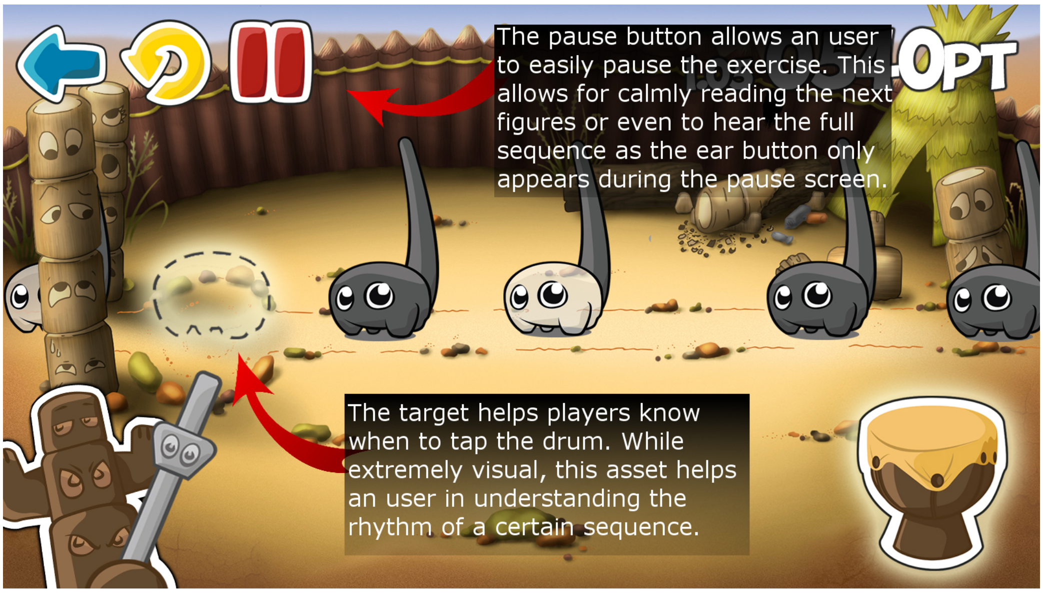


Figure 6.6: Drum Master New Design

6.3.2.3 Rhythm Master

This exercise, being the second unlocked does not try to bring something completely different to the table. Instead, a similar situation to the Drum Master exercise appears but the Rhythmiacs cannot be seen since the beginning. By making each Rhythmiac run to a certain target a sense of rhythm is created while disallowing simple memorization.

This exercise also helps the user in keeping the rhythm. Different from the Drum Master, this time the aid is purely visual which ends up working as a simple exercise for sheet reading.



The pause button allows an user to easily pause the exercise. This allows for calmly reading the next figures or even to hear the full sequence as the ear button only appears during the pause screen.

The target helps players know when to tap the drum. While extremely visual, this asset helps an user in understanding the rhythm of a certain sequence.

Figure 6.7: Rhythm Master Explained

As before, the design of this exercise is being redone by the art team and the current prototype can be seen below as a beach scenario where the Rhythmiacs must be helped to jump over the fireplace.

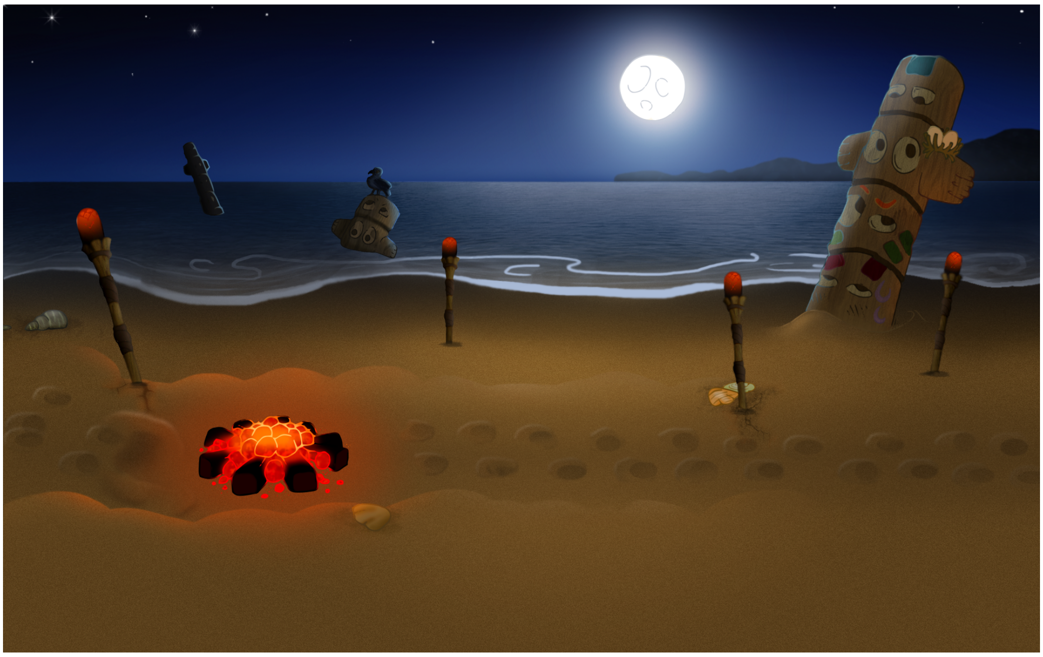
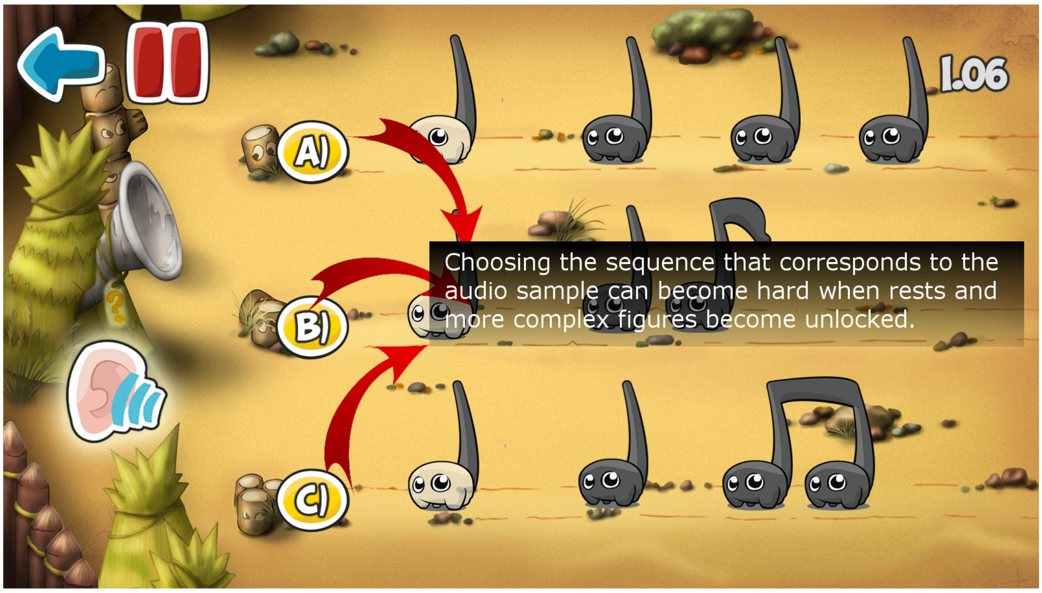


Figure 6.8: Rhythm Master New Design

6.3.2.4 Multiple Choice

This exercise is easily recognized by most users due to the similarity to several real-life situations. In short, the user is played a rhythm sequence which then must be identified from among 3 different options.

Not only is this exercise extremely important to train the hearing but it also works as a reading exercise, requiring the user to know exactly how each sequence would sound.



Choosing the sequence that corresponds to the audio sample can become hard when rests and more complex figures become unlocked.

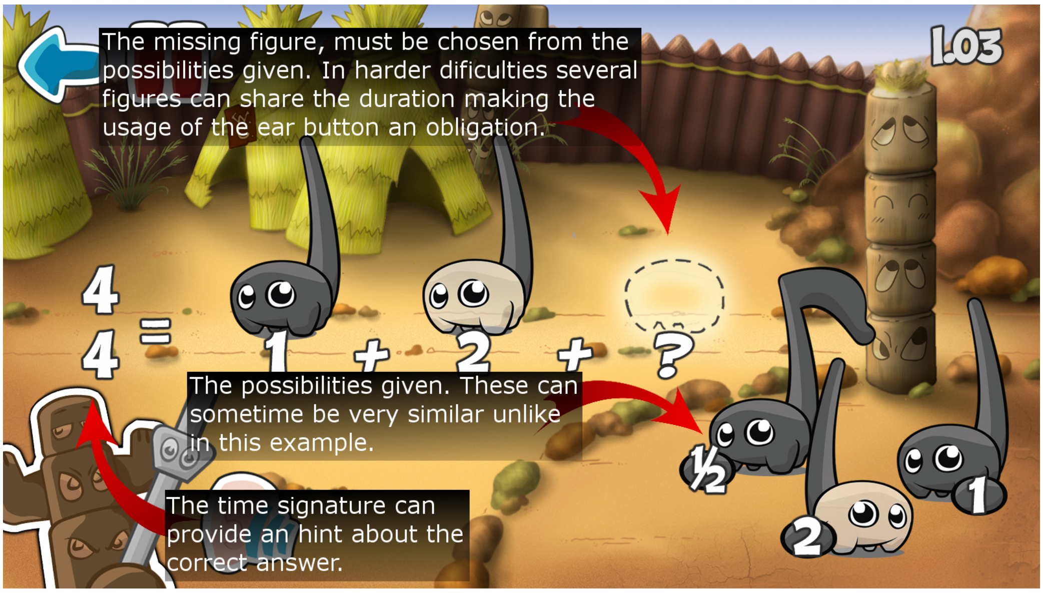
Figure 6.9: Multiple Choice Explained

6.3.2.5 Math Music

The last but probably hardest exercise requires the user to do a choice much like the previous exercise but of only one single Rhythmiac.

While the choice is less, the difficulty is greatly increased as the sequence shown is already completely right apart from the missing note making the choice much harder.

When different Rhythmiacs become unlocked and harder sequences become available, there are times when the difference be of only a quarter of a note which makes distinguishing them incredibly challenging.



The missing figure, must be chosen from the possibilities given. In harder dificulties several figures can share the duration making the usage of the ear button an obligation.

The possibilities given. These can sometime be very similar unlike in this example.

The time signature can provide an hint about the correct answer.

Figure 6.10: Math Music Explained

6.4 Future Work

While not much can be exposed as certain, there are a few extra exercises being though of. Among them, an exercise which would require the user to play different rhythms at the same time is the most promising one for being developed as it would train a part of the rhythm not available yet in the application.

There are also plans for a reverse multiple choice where the user has a single sequence and must choose among 3 different sound clips and even plans for a jamming scene where the user can freely play various percussion instruments without caring for objectives and hence training the capacity for improvising.

While the current software is a strong candidate for the beta release once the art team redesigns the scenarios, there is much more to be done in this application and hopefully, along with the rest of the Oratio's Classroom project, it can be developed into something that helps the various users worldwide to learn about such a beautiful thing as music.

Chapter 7

Architecture

"Programming without an overall architecture or design in mind is like exploring a cave with only a flashlight: You don't know where you've been, you don?t know where you?re going, and you don?t know quite where you are."[19]

Software architecture is essential in the development of any project. It addresses the software components and the interactions among them providing a solid guideline towards its implementation.

This information can help a developer to understand how the system works from a different points of view. Only then it becomes possible for said developer to extend or upgrade a specific feature.

7.1 Overview

- **Oratio's Rhythmic Village Overview**
  Provides an introduction and small overview of the project architecture.
- **Project Integration** Provides an overview of the input sources used in the application, be them user inputs or system ones.
- **Implementation Choices** Describes part of the choices made during the development of the application and the reasoning for it.
- **Logical view** Describes the functionalities that the system provides to end-users.
- **Scenarios** Describes the possible scenarios from an user perspective.

7.1.1 Oratio's Rhythmic Village Overview

- Oratio's rhythmic village was designed with weaker devices in mind. All the Scenes in the application were developed with an emphasis in minimizing the computational power required to run the scene. this however was a two-edged blade as it requires most assets to be loaded on the beginning of the scene but it was considered that was the best of the two options.
- Like referred before, all actions inside the application which might take more than the considered reasonable time have an exit case which will then fall-back to a method with a similar return value. This is used for example in situations where a randomized sequence of rhythmic figures is being processed and by simple coincidence the randomized sequence ends up being equal to a previous one. In this specific case, in case of a fall-back, a completely new sequence will be made without taking in consideration the original exercise(which usually is taken as the base of a new sequence and only small modifications are performed).
- there is also an option for microphone assisted game-play which requires the device to have a default recording-device defined. In the case that no microphone was detected, the application is able ot identify the situation and promptly warns the user about it while denying to toggle the microphone assisted game-play.
- A simple script which re-dimensions the camera in order to allow the various possible resolutions to be used is also included in the final build. This script creates black bars where needed to allow for the aspect ratio to stay constant independent of the resolution.

to sum up, any external factor, be it the device used or even simple chance is handled by the application trying to reduce its impact as much as possible while keeping the experience as seamless and invisible to the user as possible.

7.1.2 Project Integration

Oratio's Rhythmic Village was created taking in consideration classroom limitations. It allows the usage of virtually any device to which Unity3D can deploy by using generic system calls to the peripherals of the device.

These peripherals are:

- **View-port input** - Any device containing any sort of view-port input (mouse or touch) can be used to play Oratio's Rhythmic Village. This was able to be done due to Unity's native input manager.
- **Microphone** - The microphone assisted gameplay follow a similar pattern by having the software identify the default microphone and using it. A system to recognize all the possible microphones was also developed but was not included in the first version of the software due to interface limitations. These were eliminated later on.
- **Keyboard** - A keyboard is needed to make use of the profile creation screen. If the application is run in a devices which lacks the possibility to connect a physical keyboard, it will recognize it as such and open the virtual keyboard. (Such devices may include smart-phones, tablets and hybrid computers).

In order to avoid reinventing the wheel, Insignio Labs encourages the usage of open source libraries and API's. As such, four, previously acquired by Insignio Labs, libraries ended up being used to speed up the work:

- **ITween** - A simple animation library which makes use of mathematical formulas to create animation such as translations, alpha fading and scale changes. This was used mainly to avoid the creation of several .anim files (Unity's animation file) for the simplest animations such as the scale change when the mouse hovers a button.
- **Visualizer Studio** - This library allows the triggering of events when certain inputs are detected in the microphone. By using Unity's message system, this library also allowed easy debugging and fine-tuning of the microphone capabilities.
- **NAudio** - While not used in its whole, NAudio MIDI reading and saving capabilities were extensively used in the application. A simple "editor" was made with this library to allow the content creation easily. This code, while not in the release application for obvious reasons is still included in the documentation as it was fundamental or the development. Most of the exercises in the software also make use of this library to read a MIDI and generate the level based on it.
- **NGUI** - While Unity provides a GUI capability, it is considered extremely outdated and limited (Unity's next version, v4.6 will include

a new GUI system as the main headlight) as such, an alternative was found in NGUI which enables most of the functionalities expected from a GUI system.

7.1.3 Implementation Choices

While there are of course over the course of the project several times when there was the need to do an implementation choice, the ones that were considered crucial were only two as those defined most of the architecture of the application.

7.1.3.1 Generated content or static content

Two of the exercises rely on a choice by the user. While one of them requires a choice between possible figures, which can be easily solved by a single randomized number, the second one requires an entire sequence of figures which must follow certain criteria:

- No figures must be used that were not unlocked yet.
- The full time of the sequence shown must be the same as the correct answer.
- The final figure must always be a quarter or a half note.
- The amount of figures should if possible be maintained.

Because of this constraints, the usage of a simple randomization was not possible which left two possible choices. The creation of a randomization algorithm that follows these criteria or the creation of the wrong alternatives previously.

In order to increase the replay value of the application, reduce the work required to create new exercises and to allow a "random exercise" capability, an algorithm was created.

While the problem would usually be very complex, due to pedagogic situations explained by João Ramalheiro there was an order of priority to the modifications which the new alternatives should have making the algorithm easier by not relying on recursion.

7.1.3.2 Adaptive scenario or separated ones

Unity's scenes is arguably the greatest feature of Unity. While some can use them to create different levels, others can load them in the background during gameplay, creating the notion of a giant world without relying on big loading times and memory usage.

This was a doubt that was created during development, should the application exercises be one big scenario loaded and managed depending of the current exercise or define different scenes and only loading the needed one.

While keeping everything under the same scenario introduces simplicity in terms of asset management and reduces the code needed, it also increases the amount of data loaded. By dividing in several scenes completely unrelated to each other, some code duplication was inevitable and there would be difficulties in managing the assets.

The choice ended up to be the usage of separated scenarios. This however led to several problems that needed to be solved:

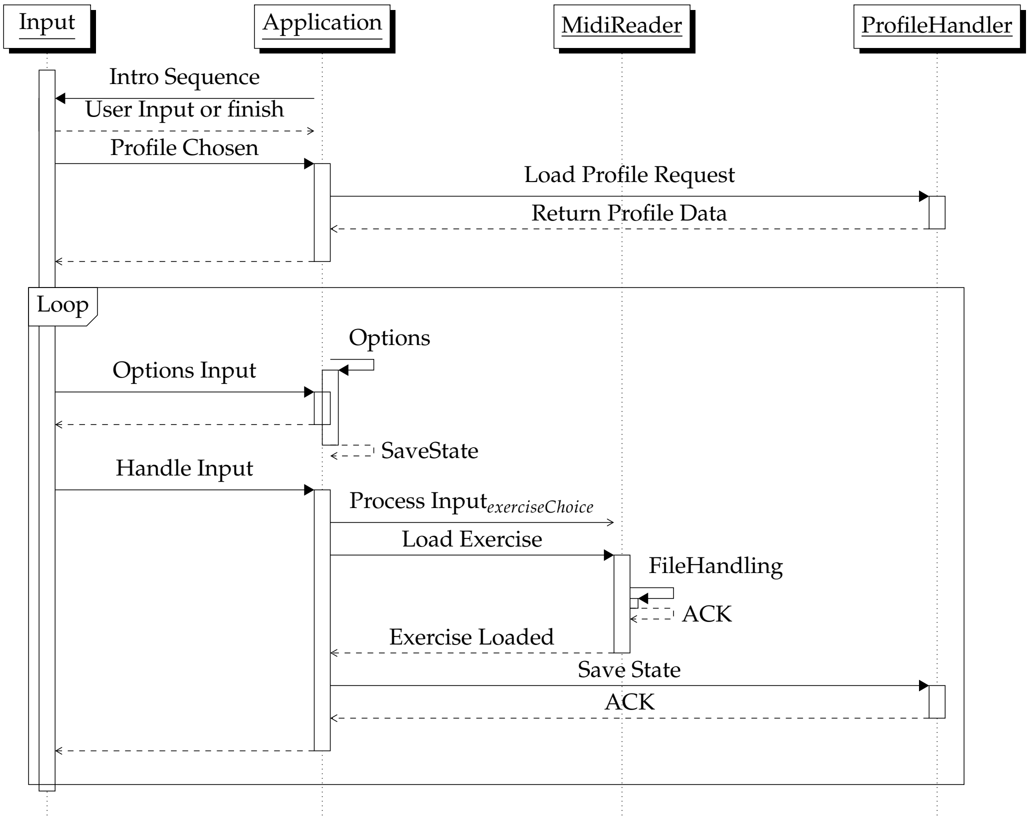
- **Level selection** - With this change, the level selection had to know which scene to load before the exercise was actually loaded. This was solved by the usage of a single letter in the beginning of the name of the exercise file representing the type of exercise to load.

- **Positioning** - While simpler than in a unique scene, having several different scenes did not remove the need for asset positioning in the scenes which could not be completed through Unity's IDE. This problem was mitigated by using several empty game objects representing placeholders for the positions which then would be accessed through code.

- **Asset management** - Management of the assets used in the scenes was completely mitigated by using Unity's capability of creating prefabs. These allow for a single asset to be changed and replicating that change to all the scenes where the object is present. This was mainly used for the GUI of the application and many objects which were to be repeated several times.

7.1.4 Logical view

7.1.4.1 Sequence Diagram



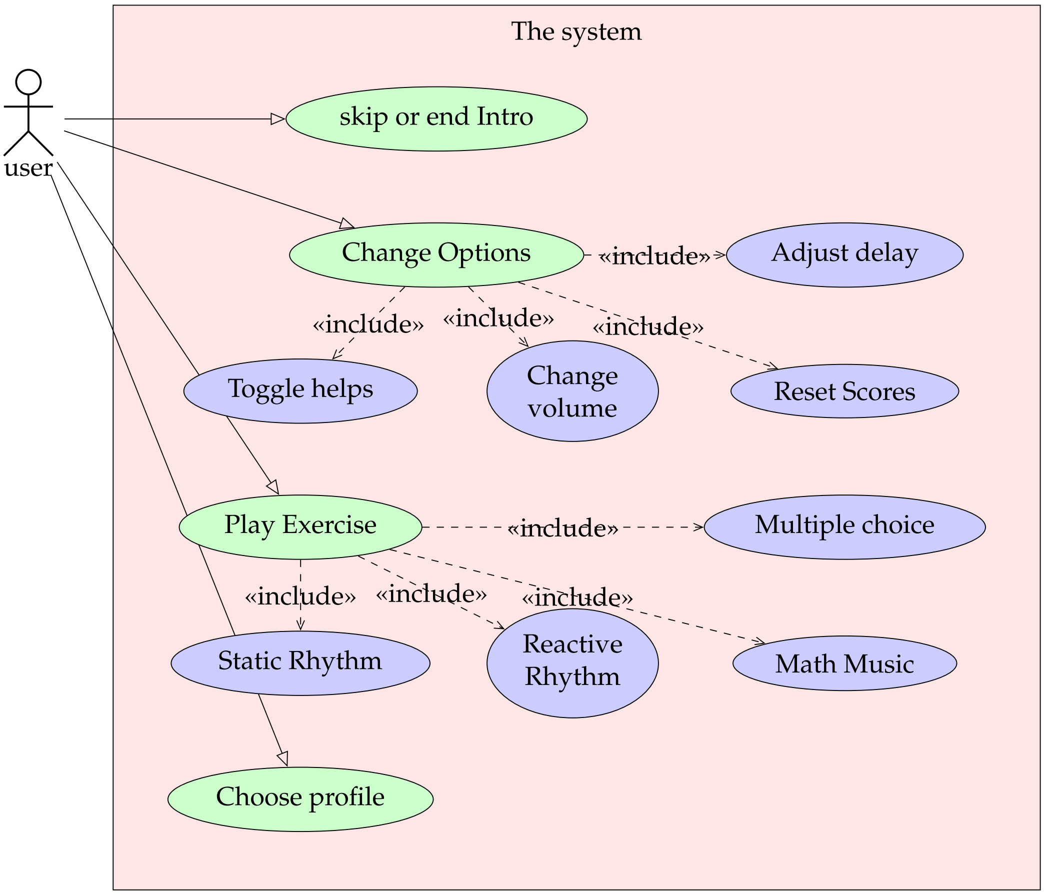
7.1.4.2 Class Diagram

Due to the nature of Unity3D, it is common for scripts to inherit directly from the MonoBehavior base class. There is also the common technique of accessing variables from different scripts by using the built in inspector in Unity3D making inheritance unneeded in most cases. There are however several cases where a class structure must be defined. Due to the size of the full classes diagram and the documented methods an external tool was used to generate such content. These can be accessed in the annex folder by opening the file index.html.
(Relative location is Annex_code_documentation\html\index.html)

7.1.5 Scenarios

There are four main possible usage scenarios which all have connections to each others and none disallows the existence of another.

Other scenarios are possible but they are either sub-usages of these or junctions fo these and as such they were not included:



Chapter 8

Project Planning

"Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law." [20]

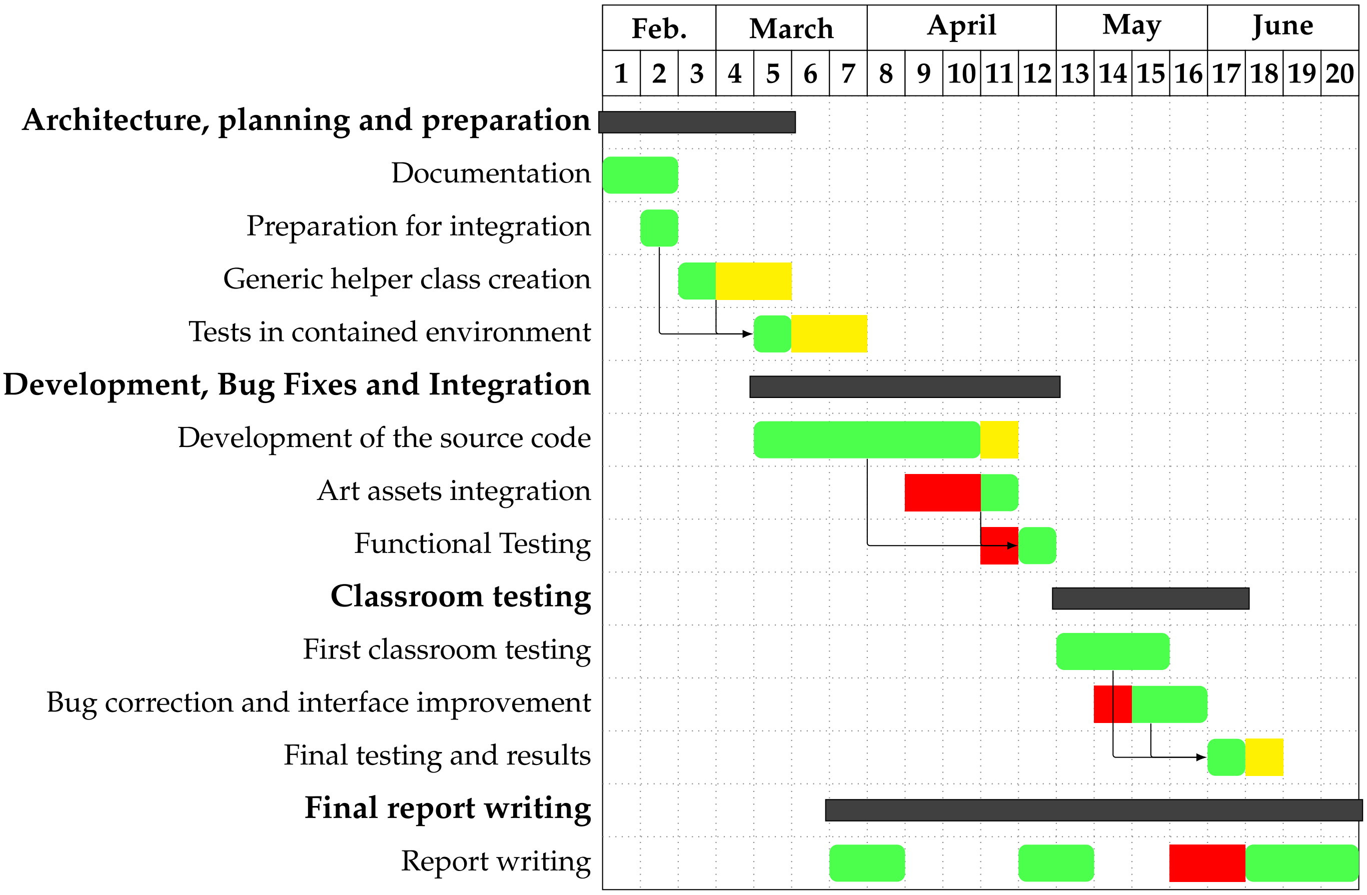
Due to the internship planning change that took place in the end of the first semester, the planning shown in the first semester was, as expected, not valid. As such, this semester's planning was defined at the beginning of the second semester and is shown later on this report. The expected planning and the final usage of time, mainly due to the good preparation and documentation created was not that different. The main differences are however shown in the Gantt diagrams shown later in this chapter.

During the second semester, a much smaller time was used in research and a greater time went into preparation, development and testing. The planning ended up comprising three different development parts and a fourth one related completely to this report writing.

- Architecture, planning and preparation - In this phase, several documents were created to make sure of the requisites of the application, eventual problems and how to mitigate them and even a document comprising every single design piece that would be needed. Later on, a paper prototype was created to further improve the interface. Some deal of time was also taken to prepare the integration of Oratio's Rhythmic Village into Oratio's Classroom such as the creation of a document where all the standards used in the company were compiled. To finish, about 4 weeks were used to create and extensively test several helper classes such as animation managers, audio managers, input event handlers and even wrappers for functions not supported by Unity natively.

- Development, bug fixes and integration - During this phase, the main development of the application was done. This passed by the creation of the entire source code, the full integration with the design material provided by Insignio Labs team and all the tests needed in order to make sure that no game-breaking bugs were left out. This phase was also a period of learning as the need for external libraries required an understanding of the underneath technologies used.

- Classroom testing - For about 3 weeks, the application underwent extensive testing at the hands of nearly 100 possible final consumers. This was possible due to the relation between Insignio Labs and IMMA, both belonging to João Ramalheiro, which began using the application in the younger student's classes and thus providing valuable feedback. This feedback was used to correct smaller bugs which passed undetected by the previous phase of testing, to further improve the quality of some design assets and to fine tune some values in the source code such as the thresholds of the microphone input.



🟩 -Final work done
🟨 - Planned but not used
🟥 - Used but not planned

Figure 8.1: Gantt diagram showing the work plan for the 2nd semester

8.1 Methodology

"Weeks of programming can save you hours of planning." - *Unknown*

The first step one must take to develop a software successfully is to adopt a methodology for the process. It is obvious that the software industry keeps on evolving and to be able to keep up it is a need to adapt an agile methodology which allows a flexible environment. Not only will this provide the team with the capability to free themselves of the restrictions of a top-down approach but also give room for improvement in previous steps whenever it is possible.

Due to the more complex demands that software nowadays faces compared to a few years back(various OS's, mobile or desktop, dozens of different requisites (many of them incompatible natively) among others), it became common for new issues to arise during the implementation phase. To top it off, the less demanding deadlines of an agile development allow for more contribution by the team, be it with new ideas or new solutions towards the project.

The methodology followed during the realization of this project does not follow any of the most common agile methods but instead used several features belonging to different practices:

- Meetings whenever any new problem arise and once a week in an informal way.
- A new working build was published every week to the Insignio Labs servers. This allowed for constant testing of the newer features.
- Unless noted in a weekly meeting, a monthly formal meeting would be performed to discuss in more detail the issues and updates of the latest builds.
- Continuous integration was used to give the design team the capability to change the art assets at any time. This was done via the usage of GIT.

Chapter 9

Software Quality

"Just because you have counted all the trees does not mean you have seen the forest." - *Anonymous*

9.1 Build Environment

All the project builds were done with Unity3D v4.3.3 and later, with the arrival of Unity3D v4.5 and the need for the project to be upgraded, the builds began being made with Unity3D v4.5.1.

9.2 Functional Tests

9.2.1 Acceptance Tests

Acceptance tests were done over the course of the implementation and as such no metrics were recorded for most of the cases as the bugs were usually revealed in the weekly meeting and promptly corrected.

9.3 Quality Tests

9.3.1 Software Performance

In order to test possible optimizations in the application. An extenuating test was done to discover possible bottlenecks. A failure was considered when either the application stopped responding or the execution time of the tests exceeded the 10 seconds. All the tests were reproduced 10 times in each scene to assure the results are valid. To note however that these tests exploit the application by stress testing it in a way not usually possible and as such the results might not interfere with the common user usage of the application. The following tests were all performed in a "Magalhães" computer and to do so, all the scenes in the application were subject to the following procedure:

- **Re-loading several times** - This was accomplished by loading the entire scene assets and as soon as it ends, re-loading the scene from the start.

Number of loads	Sucess	Failed
100	130/130	0/130
500	130/130	0/130
1000	46/130	84/130

13 scenes with 10 tries each.

Table 9.1: Scene Loading

- **Randomized inputs** - By simulating mouse clicks in random positions of the screen the input methods and logics are tested to the limit. A failure was considered when the application would become unresponsive or if an obvious lack of response was obtained from the inputs. Again, all the scenes were tested 10 times

Number of clicks per second	Sucess	Failed
100	130/130	0/130
500	130/130	0/130
1000	130/130	0/130

Table 9.2: Randomized Input

- **Extreme case usage** - Since the software is supposed to accept any type of exercises, some extreme cases were performed such as exercises with more than 100 notes. A failure was considered if any issue was detected because of the usage of this extreme cases. For this test only the scenes which allow it were used.

Number of notes	Sucess	Failed
100	40/40	0/40
200	40/40	0/40
300	40/40	0/40

Table 9.3: Extreme exercises cases

9.3.2 Usability tests

The usability tests were performed by a total of 87 students aged from 5 to 8 taking classes at IMMA. All these tests were performed using a black-box approach, meaning no feedback or help was given to the user during the testing.

These tests were conducted considering seven different use cases and the results are shown in the next table where failure means one or more cases were not able to be performed by a tester:

Status	Total	Percentage
Passed	86	98.85
Failed	1	1.15

Table 9.4: Usability tests

Due to this result and considering a less than optimal result as a failure, the design team rethought some of the GUI elements of the application, unfortunately no tests are done as of now for the new GUI which the art was not finished before the internship time ended.

9.3.3 Installation tests

These tests are short duration tests that involved installing the software in several different platforms and OS's.

Device	OS	Status
Magalhães	Windows 7	Passed
Android Galaxy S2	Android 4.1.2	Passed
IPhone 4	IOS 7.0.1	Passed
Macintosh air	Mac OS X	Passed
Clevo laptop	Windows 8.1	Passed

Table 9.5: Instalation test results

Chapter 10

Conclusion

"Enough research will tend to support your conclusions." - Arthur Bloch

10.1 Work Done

From a goal perspective, the goals that were defined by the new internship plan were concluded successfully. The following is a list of the most important tasks accomplished over the course of this internship appart from the programming itself and the writing of this report:

- State of the art Analysis - A research regarding the usage of games and new technology in education was performed during the first semester and later, on the second semester, reviewed. This research was made as an analysis of features, limitations and the value these products bring to the final consumer.

- Requirement analysis - In both semesters, due to the very different nature of the internship planning, a requirement analysis was performed and documented.

- Domain area knowledge - In order to understand the requisites of the application and the problems it tries to solve, an extensive background research about several subjects was needed, namely music and education.

- Testing - During the development, several tests were performed by many members of the team and later on by an external group of people. These tests results were always used as feedback in order to further improve the application.

10.2 Contributions

During the internship, mainly during the development, some other contributions were made in the context of the internship:

- **Insignio Labs payment interface** - The usage of PayPal instant payment notification system required the creation of a PHP script which would handle the incoming messages and send back to the buyer a confirmation e-mail which was done during this internship.
- **Random rhythm creator** - In an attempt to create more content for the application with less work, a small interface which allows an user to create midi files based on a few small rules was created. This was later used by Insignio Labs to create all the exercises in the first version of Oratio's Rhythmic Village.
- **Prototyping** - Several prototypes were created in order to try new types of exercises. All these were either further developed to become one of the current exercises or were discarded as ideas not suitable to be part of the software.

10.3 Future Work

While the application is currently in its first stable version, several plans are already in motion before it will become commercialized.

Among these, a new design is planned along with a few new features such as the possibility to play a survival type of exercise where an user would just keep playing aiming for the highest score possible.

The small lack of intuitiveness on the interface is also a problem being mitigated at this moment which leads me to believe the commercialized version will indeed have a perfect score when testing with the younger users.

The generic scripts created are also being used in another of Insignio Labs projects as they were considered valuable in reducing the development time of some common tasks like loading new scenes or changing aspect of an object in certain events.

10.4 Lessons Learned

Contrary to the rest of the report this section represents the author's opinion and it will be written as so. It is redacted in the first person and in a less formal language.

Although many times during these internship I found myself wondering why I was wasting time in trivial problems like changing the volume of an audio asset in a specific moment of the application, I also found myself enjoying these small specs of programing where all the preparation done was fruitless as it gave me time to think outside of the box and create new and original alternatives to my coding.

While I do not feel the current application is in its final form, which it isn't according to Insignio Labs CEO, I do believe it has enough capabilities to be used in the classroom as it is with a beneficial effect on the users learning. I can only hope the final version will indeed revolutionize the way of learning music.

Working with Unity was also an incredible experience, the integration of so many different aspects to create a final software was a new experience for me and I expect it to be useful in the future.

In the end, i can't help but look back and check the last years work. I found myself integrated in a great team, learning everyday how to make better software and in the end, created something i can be proud of.

Bibliography

[1] Insignio Labs. *Insignio Labs*. 2014. URL: http://www.insigniolabs.com/en/about/ (visited on 06/30/2014).

[2] Insignio Labs. *Insignio Labs*. 2014. URL: http://insigniolabs.com/en/products/ (visited on 06/30/2014).

[3] Cathy Davidson. *Now You See It: How the Brain Science of Attention Will Transform the Way We Live, Work, and Learn*. Ed. by Penguin Group USA. .2. June 21, 2011. Chap. 5, p. 86. ISBN: 978-0-670-02282-3.

[4] M. De Aguilera and A. Mendiz. "Video games and education:(Education in the Face of a Parallel School)". In: *Computers in Entertainment*. Ed. by CIE. 1st. Vol. 1. 2003, pp. 10–23.

[5] David Nagel. *Microsoft Commits $250 Million to Partners in Learning*. Nov. 29, 2012. URL: http://thejournal.com/articles/2012/11/29/microsoft-commits-250-million-to-partners-in-learning.aspx (visited on 06/30/2014).

[6] Sarah Perez. *Microsoft Invests $75 Million To Expand Digital Education In Africa*. Dec. 3, 2012. URL: http://techcrunch.com/2012/12/03/microsoft-invests-75-million-to-expand-digital-education-in-africa (visited on 06/30/2014).

[7] Robert Stephens. *Ananth Pai: Gamifying the Classroom*. 2011. URL: http://fora.tv/2011/09/16/Ananth_Pai_Gamifying_the_Classroom (visited on 06/30/2014).

[8] Marczewski and Andrzej. *Gamification: A Simple Introduction*. 1st. Apr. 2012, p. 46.

[9] Google. *Gamification search term trend*. Dec. 1, 2014. URL: http://www.google.com/trends/explore#q=gamification (visited on 06/30/2014).

[10] Matt Richtel. "Once a Booming Market, Educational Software for the PC Takes a Nose Dive". In: *The New York Times* (Aug. 22, 2005). URL: http://select.nytimes.com/gst/abstract.html?res=

F60D1EFE3A5A0C718EDDA10894DD404482&fta=y&incamp=archive:article_related (visited on 06/30/2014).

[11] Curtiss Murphy. "Why Games Work and the Science of Learning". In: *Alion Science and Technology* (2011). URL: http://www.goodgamesbydesign.com/Files/WhyGamesWork_TheScienceOfLearning_CMurphy_2011.pdf (visited on 06/30/2014).

[12] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Ed. by Harper & Row. 1990. ISBN: 978-0-06-016253-5.

[13] J. Chen. "Flow in Games". In: *Master's Thesis* (2007).

[14] Goudas M. et al. "It ain't what you do, it's the way you do it!" In: *The Sport Psychologist* (9 1995), pp. 254–264.

[15] Willem Buiter Chief Economist at Citigroup. "debate on the microfoundations of macroeconomics at economic academia". In: (Feb. 2009).

[16] Seth Godin. *Stop Stealing Dreams (What is school for?)* 2012. URL: http://www.sethgodin.com/sg/docs/stopstealingdreamsscreen.pdf (visited on 06/30/2014).

[17] John Grego PhD and Roumen Vesselinov PhD. "Duolingo Effectiveness Study". In: *journaltitle* (Dec. 2012).

[18] Louis Srygley. "Live Conference". In: (2004).

[19] Danny Thorpe. *The Guru's Guide to Transact-SQL*. Ed. by Addison-Wesley Professional. Jan. 6, 2002, p. 303. ISBN: 978-0201700466.

[20] Douglas Hofstadter. *Gődel, Escher, Bach: an Eternal Golden Braid*. 20th anniversary ed. 1979, p. 152. ISBN: 0-465-02656-7.