GameCourseUI

Mariana Frazão Matos Saraiva

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Daniel Jorge Viegas Gonçalves
Dr. Amir Hossein Nabizadeh

Examination Committee

Chairperson: Prof. Rui Filipe Fernandes Prada
Supervisor: Prof. Daniel Jorge Viegas Gonçalves
Member of the Committee: Prof. Sandra Pereira Gama

November 2021

Acknowledgments

I would like to thank my boyfriend, Gonçalo Melo, for all the love, support, friendship, for always being there for me and for never letting me give up, even when I wanted to. All these years and specially this project would not be possible without his support.

A special thanks to my best friends Raquel and Madalena for their friendship and for cheering me up during these last months, when I needed the most.

I would like to thank my parents for their friendship, encouragement and caring over all these years, for always being there for me. I would also like to thank my grandparents, aunts, uncles and cousins for their understanding and support throughout all these years.

I would also like to acknowledge my dissertation supervisors Prof. Daniel Gonçalves and Amir Hossein Nabizadeh for their insight, support and sharing of knowledge that has made this Thesis possible.

This work was supported in part by the National Funds through the Fundação para a Ciência e a Tecnologia (FCT) under Project UIDB/50021/2020, and in part by the Project GameCourse, Portugal, under Grant PTDC/CCI-CIF/30754/2017.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

Abstract

Gamification is defined as the use of game design elements in non-game contexts and, recently, has been introduced in many fields. In education, it is used to increase students' motivation and involvement in a course or some activities. The User Interface (UI) of the system has a huge impact on how it is accepted because, even if the system fills all the requirements, its ease of use and visual appeal can lead to rejection by users. Although gamification increases overall motivation, some students do not feel engaged due to the way game elements are presented to them. GameCourse is the system used to gamify the Multimedia Content Production (MCP) course at Instituto Superior Técnico (IST), whose UI has been improved since the previous year. The purpose of this thesis is, besides improving some UI features that are already implemented, add features and mechanisms to allow customization of some views for students with different profiles, providing them with the game elements they feel more comfortable and motivated, as well as ease the configuration and management of the courses. We accomplished our goals by easing the personification of the view as well as improve the User Experience (UX) in the module configuration pages.

Keywords

Gamification; Gamification in Education; User Interface; User Experience.

Resumo

A gamificação é definida como o uso de elementos de *design* de jogo em contextos fora de jogo e, recentemente, tem sido introduzida em diversas áreas. Na educação, é usada para aumentar a motivação dos alunos e o seu envolvimento nas disciplinas ou em algumas atividades. A interface do utilizador do sistema tem um enorme impacto em como este é aceite, porque, mesmo que o sistema preencha todos os requisitos, a facilidade em usá-lo e o seu apelo visual pode levar à rejeição dos utilizadores. Apesar da gamificação aumentar a motivação no geral, alguns alunos não se sentem envolvidos devido à maneira como alguns elementos de jogo lhes são apresentados. o Gamecourse é o sistema usado para *gamificar* a unidade curricular Produção de Conteúdos Multimédia (PCM) no Instituto Superior Técnico (IST), cuja interface tem sido melhorada desde o ano passado. O objetivo deste projeto é, além de melhorar alguns aspetos da interface do Gamecourse já implementados, adicionar mecanismos que permitam a personalizar alguns elementos para alunos com diferentes perfis, mostrando-lhes elementos de jogo com os quais eles se sentem mais confortáveis e motivados, assim como facilitar a configuração e gestão das disciplinas. Conseguimos atingir os nossos objetivos ao facilitar a personalização das *views* assim como melhorar a experiência do utilizador nas páginas de configuração dos módulos.

Palavras Chave

Gamificação; Gamificação na Educação; Interface do Utilizador; Experiência do Utilizador.

Contents

List of Figures

List of Tables

Acronyms

API	Application Program Interface
CSS	Cascading Style Sheets
EL	Expression Language
IST	Instituto Superior Técnico
MCP	Multimedia Content Production
NASA-TLX	NASA Task Load Index
PCM	Produção de Conteúdos Multimédia
SUS	System Usability Score
UI	User Interface
UX	User Experience
XP	Experience Points

1

Introduction

Contents

For the past few years, the learning process has benefited from the introduction of several activities which make it more fun and engaging. Mainly in the education field, different learning processes have been employed by providing students with a more motivating system, in which they feel engaged, without losing their interest in the subject. With monotonous classes, students may lose motivation and interest, leading to them not learning as much as they should or could. Since Education is a crucial part of our lives, when we acquire the knowledge, skills, and values to face the challenges of the future, it is important to decrease the number of students that fail in school. The innovation of the learning process by taking different approaches could be helpful to make it more attractive.

These new learning processes include introducing new elements, mainly game elements, that keep students motivated. This is called gamification, defined as “the use of game design elements in non-game contexts” [Deterding et al., 2011]. These game elements, like leaderboards and badges, give a sense of fun and entertainment to students while doing their tasks. Even so, game elements have different effects on different people. Thus, it is important to evaluate which ones are better for each context by considering the types of players we have on it. When planning gamification, there are many factors we have to be careful about, namely, the context, because, in a classroom scenario, it will affect the learning process of the students. Hence, a well-planned and successful gamified classroom can bring more interest to students in the acquirement of knowledge as well as motivate them to study.

The impact of the User Interface (UI) on students can change the way they accept gamification. The visual appeal of the interface and its ease of use, which it influences, are determinants for users to continue using it and, thus, in keeping them engaged within the application [Pengnate and Sarathy, 2017]. Therefore, in the education context, gamification can be better received by students if the interface has a good visual appeal and is easy to use. In addition, the interface and the elements presented on it can influence the gaming experience since students have perceived the game differently depending on how it is shown to them. For example, a leaderboard gives students feedback on their performance, as well as the performance of their peers. However, it is possible to display the same information in different ways, to mitigate the negative impact leaderboard might have, by properly designing how the leaderboard shows itself.

Furthermore, the fact that students have different learning abilities should be taken into account for the learning process. If it is equal for everyone, students with more abilities tend to be disinterested and the ones with fewer skills tend to give up and fail [Brühlmann, 2013]. The introduction of some fun along with elements that fit each student’s flow zone can increase the students’ engagement [Brühlmann, 2013]. While learning, they feel deeply immersed in the activity, creating a balance between their skills and the challenges. Moreover, the abstraction from the serious environment allows them to enjoy and have fun during the activities. Thus, when applying gamification, these differences between students have to be taken into account, to give the best experience to all of them. In fact, with different user interfaces suitable to each student profile, students may feel the environment more attractive insofar they realize that the game elements were chosen for them, spending more time in the platform [Monterrat et al., 2015].

In the Master's degree in Computer Science and Engineering at Instituto Superior Técnico (IST), there is a successfully gamified course - Multimedia Content Production (MCP), where elements such as a leaderboard, badges, and points are used to create an engaging environment, resorting to the GameCourse system. Currently, this system needs improvements to its UI, and also the ability to provide differentiated versions of UI to different students, in order to give the best experience to everyone.

1.1 Objective

The goal of this thesis is to **improve the current GameCourse interface and, particularly, create mechanisms to build different views for students with different profiles and ease the configuration and management of the courses**. The interface that deals with the creation of views will be more flexible and it will make it easier for teachers to create and manage the views. It will have the capability to build different interfaces for different types of students. To achieve this, we took into consideration the concept of role, hierarchical, while building the views, in order to show the users the right view according to their role. In addition, we needed to modify the current view editor to allow creating different versions of each view for the different roles.

By implementing these changes, we aim to arouse interest in other teachers, not only from the Department of Computer Science and Engineering but from the whole IST, and perhaps from other universities.

1.2 Document Structure

In the second section, we will explore related work, where we define gamification, explore important elements that a gamified context should include, present different taxonomies to classify players according to their personalities and, in the end, provide some examples. After that, we explore the employment of gamification in the education field, including in the MCP course and, finally, we discuss different ways to personalize the gamification in this field. Then, in the third section, we describe the current state of the GameCourse. The fourth section describes what we have changed in the system as well as all the new features we added. In the Chapter 5, we analyse the results of the evaluation performed on the system. The last section includes the conclusion of this work as well as the future work.

2

Related Work

Contents

This section will present research about gamification and its applications. Firstly, we define the concept of gamification, and present the most recent well-known taxonomies regarding different types of players. Afterwards, we show different examples of gamification. Secondly, we will approach how gamification is applied to education along with some examples. We will then focus on the UI and its impact in the employment of gamification, finishing with the explanation of how the MCP gamification system currently works.

2.1 Gamification

Defined as "the use of game design elements in non-game contexts" [Deterding et al., 2011], gamification has become extremely popular in recent years. Its benefits in increasing intrinsic motivation [Mekler et al., 2013] and the ability to add an entertaining dimension to monotonous tasks has led to its usage in a wide range of settings. Namely, great success has been achieved by the employment of gamification in health care [Johnson et al., 2016], traffic management, and learning applications [Barata et al., 2013a].

This definition introduced by Deterding is fully explored in his paper. The term "Gamification" is clarified using the two-dimensions of playing/gaming and parts/whole to differentiate from serious games, toys and playful design. Games and serious games can be distinguished from gamification through the parts/whole dimension since gamification only uses game elements (parts of a game). In turn, gamification is distinguished from playful design and toys through the playing/gaming axis, where gamification belongs to the gaming quadrant because gamification implies structured activities and well-defined rules while playing requires unstructured activities.

Gamification is a strategy to improve user engagement and is highly determined by user experience. To optimize the user's motivation, Ning et al. designed a gamified system that fulfils the users' requirements - a three-level gamification system design method based on goal, rules, and tools [Ning, 2018]. The goal is based on the real need of users, giving them "value-added experience of emotional experience". Rules are the framework of the gamification system. The tools are the external appearance, give feedback to the users and show the results of their behaviour. These three elements create a gamification system where each element is directly connected to User Experience (UX) elements (Tools – UI; Rules – Procedure; Goal - Nature). Its design method aims to enhance the UX, using characteristics of games and design as well. The gamification design affects the gameful experience, according to Thomas et al., by stimulating user motivation. In turn, this enhances their engagement making the experience more pleasant and enjoyable [Leclercq et al., 2020].

Although gamification is defined as the use of game elements in non-game contexts, this is not enough. We cannot simply apply these elements, call it gamification, and hope it will work and return the results we want, regarding user engagement. Gamification gets unsatisfactory results because the priority is its components such as leaderboard, badges and points instead of the balancing and the design of the game. This leads to a failure in engaging and motivating players [Deterding, 2012]. To build a successful gamification system, it requires understanding the users and their needs. Besides, the gamified activity must already have some intrinsic value to increase users' engagement and their desire to participate.

Another determinant point that makes gamification successful is feedback. Most of the time, gamification is associated with experiences of affective feedback [Hassan et al., 2019]. Affective feedback is the positive/negative emotional return experienced from an intrinsic or extrinsic stimulus. The users' behaviour is directly influenced by this feedback as well as their motivation for the concerned activity. According to Hassan et al., gamification is associated with experiences of affective feedback, which are directly associated with both perceived benefits by users and the intention to continue using the system [Hassan et al., 2019].

User's behaviour depends on their motivation to participate in some activity. The importance of intrinsic motivation in managing user behaviour is explored on "Self Determination Theory", developed by Richard Ryan and Edward Deci [Ryan and Deci, 2000]. The three innate psychological needs of the SDT that support motivation are autonomy, competence and relatedness.

Autonomy is related to the human perception of ownership of their own life. Autonomy is important to increase motivation since people keep motivated if they know that their effort caused their success.
Competence is related to a sense of control and experience dominance. The seek for approval from significant others is also present in Competence. The feeling of Competence can be aroused by the feedback provided which will increase the motivation to accomplish that task.
Relatedness is the need to be connected to others and to interact with them.

The concept of flow, introduced by Mihaly Csikszentmihalyi, is highly related to competence since flow's properties can lead to a feeling of competence [Brühlmann, 2013]. Flow describes a mental state reached when a person is entirely immersed in an activity [Amaral, 2013].

This is experienced when someone's perceived skills are balanced with the perceived challenges. When these two things are balanced, you are in the Flow Zone. Different people have different flow zones for each task/game. If you are overqualified for an activity, you will end up in a boredom state because there are no challenges and the activity is too easy for you. On the other hand, if the challenge is much bigger than your perceived skills, you lose the state of flow and you go to an anxiety state. A well-designed game maintains players in their Flow Zones, and that is why those games are successful - the players enjoy and have fun while playing. In fact, during the game, players improve their skills. For example, while playing board games, children develop many skills such as, attention, thinking and memory – cognitive skills –, cooperation and listening – social skills – and the ability to deal with disappointment – emotional skills [Strode, 2019]. Integration in the game contributes to a positive attitude and enhanced motivation in the learning process.

Just as motivation - provided by the elements of the game - is a fundamental point for gamification, the flow experience must also be, providing a more fun experience while balancing users' skills with challenges.

2.1.1 Player Types

Although gamification has been successful, sometimes it fails for some people and they don't get engaged. To manage this problem, the personality of each player must be taken into account to create a gameful experience for any type of player. The same game element may not have the same effects for players of two different types, that is, it could have positive results for one and negative for the other one.

A very well-known taxonomy regarding the player types was presented by Bartle, in 1996 [Bartle, 1996]. He identified four types of players.

Achievers are players whose main goals are accumulating points and their status - rising in levels. They like to show their progress and collect badges/medals. In short, their engagement basis on extrinsic motivation. **Explorers** like to discover new places and secrets, looking for interesting features and figuring out how things work. To stimulate explorers, there should be unpredictable mechanisms, such as random rewards, hidden areas, "easter eggs" (small bonus within a game). **Socializers** are the type of players that collaborate with other players to get better results than they could achieve just by themselves. Their main reason to play is to develop inter-player relationships because they are more interested in people. Chats and gifting are two elements that keep socializers playing. **Killers** simply want to be noteworthy imposing themselves on others. Their joy is based on killing other players' avatar and see them outplayed. Bigger the damage, the better.

These four types can be displayed in a two-dimension of players/world and interacting/acting, as we can see in Figure 2.1. On the one hand, Achievers and Explorers are attentive to the world - achievers act in the game's world while explorers enjoy interacting with the game's world. On the other hand, Socializers and Killers are concerned about the players - socializers are the ones who interact with players, while killers act on the players.

Even though this is a very well-known classification of player types, there are some more recent surveys regarding this topic. In 2013, Nacke et al. introduced a model - BrainHex - based on neurobiological findings in which seven archetypes of players are presented [Nacke et al., 2013].

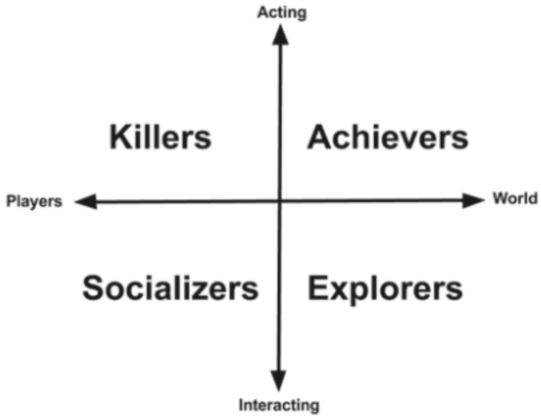


Figure 2.1: Bartle's four Player Types

Seekers are curious about the game world and the interest mechanism is what drives them, which is related to the part of the brain that processes sensory information and the memory association area. **Survivors** enjoy the feeling of fear and the intensity of a terrifying experience, which leads them to an adrenaline rush. **Daredevil** likes to play on the edge and have the excitement of risk-taking. A Daredevil enjoys rushing at high speed while still in control. For Daredevils and Survivors, the neurotransmitter that stimulates their excitement is epinephrine, which is related to the amygdala (fear centre). **Masterminds** are focused on solving puzzles as well as facing problems that require strategy, being sure that they have the most efficient solution. They depend on the decision centre of the brain. **Conqueror** players are motivated by anger (controlled by the hypothalamus), they want to feel challenged with adversities since they do not like easy-winning, they like difficulties. For the conquerors, they have to struggle until they achieve victory. **Socializers** like talking with people, helping others, and be around people they trust. Because they are people-oriented, they get angry at those who abuse their trust. This behaviour is related to oxytocin, a neurotransmitter connected with trust. **Achievers** are goal-oriented players who like to complete their achievements. The sense of satisfaction in attaining goals comes from the release of dopamine.

One year later, Marczewski proposed another model – User Type Hexad [Marczewski, 2015]. Marczewski decided to create his model since he believes that Bartle's types are not relevant when building gamified systems. Initially, instead of only observing user behaviour, he created a model where each type was related to one of our four intrinsic motivation. Other two types were introduced, with other related motivations. This yielded the Hexad model, identifying 6 types. **Socialisers**, motivated by Relatedness, want to interact with other players. **Free Spirits**, known as the ones who want to explore, are motivated by Autonomy. **Achievers**, looking for new challenges and want to learn new skills, are motivated by Mastery. **Philanthropists**, motivated by Purpose and Meaning, enjoy making better other people lives with no intention of reward. These four types are motivated by intrinsic motivations. **Disruptors** want to disrupt the system to force changes in the system, either positive or negative, and are motivated by Change. Finally, the **Players** only want to collect rewards, so they work just for it, that is, they are motivated by extrinsic Rewards. Each of the latter two types can be subdivided into four types. For example, Players can be subdivided into four sub-types, as there are many ways for Players to collect their rewards and, therefore, there may be different types of Players – some like to help others (as **Philanthropists**) as long as it gives them a reward and others connect with others (as **Socialisers**) that are useful to gain something.

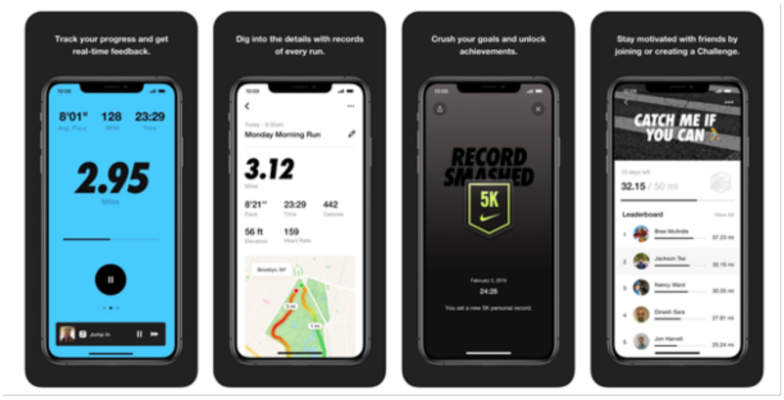
2.1.2 Gamification Examples

In this section, some examples of gamification in different areas, such as health care/fitness, are presented.

Nike Run Club

Nike Run Club[1] is a gamified application that intends to make people better runners by giving them the guidance, the inspiration and the innovation they need, through a user-friendly interface (Figure 2.2).

In this app, you can count the number of kilometres they ran as well as the time they took doing it. Then, they can see your progress throughout time. If you want personalized coaching, you can have a plan executed by a Nike Coach taking into account your goals which, in the future, is adapted as you progress. One of the things Nike uses to keep runners motivated is having playlists available in the app to push them to run a little harder and a little farther. Another interesting and interactive feature is the capability to easily compare and compete with your friends and fellow runners.



Track your progress and get real-time feedback.

Dig into the details with records of every run.

Crush your goals and unlock achievements.

Stay motivated with friends by joining or creating a Challenge.

Figure 2.2: Nike Run Club Mobile App Interface

[1] https://www.nike.com/nrc-app

Starbucks

Starbucks is an American multinational chain of coffeehouses, known all over the world. Its mobile application (Figure A.2) has nailed loyalty by giving mobile rewards to customers. Its loyalty program is gamified using clear goals, progress bars and attractive rewards like free drinks or free food. With this reward structured, customers started to buy more products and got more loyal too buying exclusively on Starbucks.

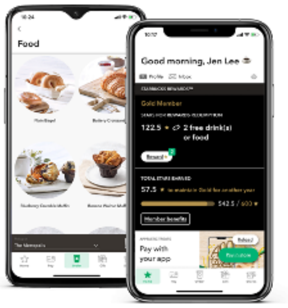
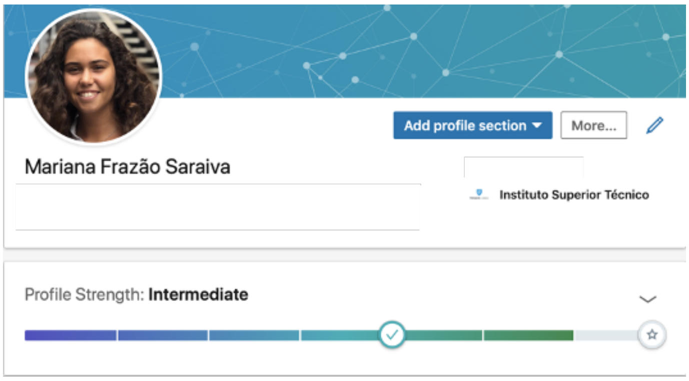


Figure 2.3: Starbucks Mobile App Interface

LinkedIn: Profile strength

LinkedIn is a social network for business professionals, with millions of users from all over the world. They applied gamification through a progress bar to the strength of the user's profile, as we can see in Figure 2.4. When editing your profile, this progress bar is visible, which arouses users' psychological desire to complete their profile by 100%. Users will naturally want their profiles 100% filled, as this gives them more visibility on the network, which leads to an increasing number of opportunities.



Mariana Frazão Saraiva

Instituto Superior Técnico

Profile Strength: **Intermediate**

Figure 2.4: LinkedIn Profile with the Strength Bar

Besides these fields, gamification has been successfully applied in the education field, as we can see in the next section.

2.2 Gamification in Education

One of the main fields in which Gamification is applied is education. The purpose of applying gamification to the learning process is to make it more engaging for students and, consequently, maximize their success.

Some websites use game elements to maintain their users engaged. For example, Duolingo[2], which is a free language learning application that uses gamification, provides a great number of exercises to learn a language applying several game elements in a meaningful way that keeps the will of learning in the users. Some of the game elements are achievements, leaderboard, levels, badges and Experience Points (XP), among others. All questions, when correctly answered, are rewarded with XP. As long users complete lessons, they gain badges (equivalent to skills) and each badge has a progress bar representing how good are they at that skill. Throughout the experience, there is locked features to stimulate users to unlocked them by progressing their knowledge. There is also a leaderboard where their performance relative to their friends can be checked. By comparing them to their friends, they motivate themselves to learn more. As a bonus, it has a very appealing interface (Figure 2.5).

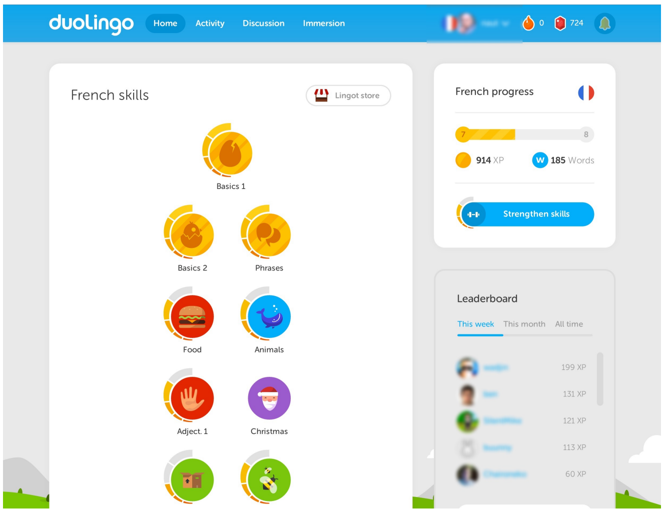


Figure 2.5: Duolingo User Interface

Another example of an educational website is Khan Academy[3], a non-profit educational organization created in 2008. They provide free, world-class education for anyone by making available a large number of lessons in a wide range of subjects like math, physics, economics, computer science, among others. Remarkable gamification has been combined with the learning process, including badges to award when goals are achieved. There are six types of badges (Figure 2.6) with different levels of difficulty and they are given on different occasions, for example, the Meteorite badge is easy to earn when the user is

[2] https://www.duolingo.com

[3] https://www.khanacademy.org

getting started. On the other hand, Earth badges are rare, because they require a significant amount of learning. This platform has also the capability to set goals along with progress indicators related to those goals to keep users motivated.

BADGE TYPES

Meteorite badges are common and easy to earn when just getting started.

Moon badges are uncommon and represent an investment in learning.

Earth badges are rare. They require a significant amount of learning.

Sun badges are epic. Earning them is a true challenge, and they require impressive dedication.

Black Hole badges are legendary and unknown. They are the rarest Khan Academy awards.

Challenge Patches are special awards for completing topic challenges.

Figure 2.6: Different types of Khan Academy badges

In a more formal learning context, there are several studies regarding student engagement and also many studies proving the benefits of gamification.

To study how gamification affects students' engagement, Garden and Rivera [Garden and Rivera, 2018] presented a framework based on three theories - Landers' Theory of Gamified Learning [Landers, 2015], Kahu's Framework for Student Engagement [Kahu, 2011], and Bedwell's Taxonomy of Game Attributes [Bedwell et al., 2012]. The elements from the different theories that are linked to each other are mapped into a single state. The states are connected according to their cause-effect relation. For example, the antecedents of Engagement (e.g. assessment) state precedes the Engagement state (e.g. behaviours/attitudes). In turn, the latter precedes the Consequences of Engagement state (e.g. learning outcome). Kahu proposes that this whole process (Engagement) is circular. Thus, he believes that increased engagement leads to increased learning outcomes (e.g. academic performance), which can directly affect the antecedents of engagement, for instance, the motivation.

The importance of Narrative for gamification is discussed by Toledo Palomino et al. [Toledo Palomino et al., 2019], who argues it is misunderstood and, therefore underutilized, in this context. To be used as a design element for gamified projects, narrative, in the field of education, has to cover specific features like having an actor, the element of choice, a sequence of events (must be in a logical order so the progress makes sense to the user) and, essentially, the feature of user experience, since it is an element able to enhance motivation on students. Therefore, narrative "can be understood as the process in which the user builds his own experience through a given content(...)". The narrative is one of the most important game dynamics, which are stated as "the big-picture aspects of the gamified system(...)" by Werbach and Hunter [Werbach and Hunter, 2012].

In the education field, it is important to make sure that the majority of the students in the class are engaged. Otherwise, their lack of motivation and interest would lead them to fail the course. Recently, Kocadere et al. presented a research about which game elements affect each player type positively and negatively [Arkün Kocadere and Çağlar Özhan, 2018]. In this 7-week study with undergraduate students, the game elements used were: leaderboard, achievements, point, badge, content unlocking, level, gifting, team, and story. The Bartle's four types of players were considered as the "Player Type Scale".

At the end of those seven weeks, they concluded that:

– **Killers** like to see the leaderboard, the points since it stimulates competition and their level because they can see their status which can increase their reputation. Besides, team and gifting affect them negatively because working as a team can ruin their status and gifting ruins the competitive environment.

– **Achievers** were positively affected by the leaderboard, points and levels, and achievements because of the competition, the sense of progression and facilitated movement, respectively. Teamwork has a negative effect on Achievers since each one's cooperation could not be equivalent. They also believe achievements are to make it on their owns and, therefore, gifting affects them negatively.

– **Explorers** see the element level as a good indicator for their progression which allows them to explore while progressing. They consider the story a fundamental element to be immersed in the environment. Furthermore, badges and achievements given as rewards are seen as part of the narrative. Explorers like to play on their own thus team is an element that has a negative effect.

– **Socializers** are a person-oriented type of player, they prefer to enjoy the game and talk to people rather than compete over achievements and, therefore no elements are affecting them negatively. On the other hand, some elements keep them motivated, such as story, badges and achievements that make them feel part of the game. Moreover, these players like to help each other, in other words, do teamwork along with gifting, providing interaction with other players and the sense of cooperation.

In order to apply gamification in a classroom, Lee Sheldon created The Multiplayer Classroom, a model to design a course [Sheldon, 2011]. This model includes quests and missions which would equal to traditional evaluations such as quizzes and assignments. Moreover, students were not evaluated with the normal grading system with a letter, their evaluations were graded with XP. At the end of the course, their XP had equivalence to a letter grade. This grading system helped the teacher tracking students' progress throughout the course. With this method, Sheldon showed great results concerning the application of game design elements in a traditional educational course, since students were the players and the teacher the Game Master in a game environment - the classroom.

Falip et al. conducted a study about a university course of the Multimedia degree at Ramon Llull University, Barcelona, Spain where they applied gamification, along with new methodologies as learning by doing and collaborative work among students, and used new technology such as visualization 3D [Falip et al., 2014]. Since this course teaches 3D arts, their goal was, besides increase student motivation, discover different methodologies for teaching 3D modelling. The gamification was employed by creating a profile page with the following elements: Avatar that shows user information, Score that shows the user grades, Mission contains all exercises and exams in the course and Adventure Map that gives global information of the course. In order to analyse the results, they did both quantitative and qualitative evaluation. For quantitative evaluation, the students answered some questions and it gave them good results as more motivation, efficiency and satisfaction perceived by the students. Qualitative evaluation allowed them to understand what went well, what was negative for the students' experience and how they could improve it. Analysing the academic results, even with all the negative aspects, the motivation increased, showing that the employment of gamification with some other teaching methodologies such as collaborative work and with new technologies like virtual reality, created an environment able to provide a more engaging experience and, consequently, a better performance. In short, it increased the effectiveness of the learning process.

Student engagement is one of the main goals of gamification. Therefore, attempting to convert a boring university course – Research Methods – into a more enjoyable course, Martin Sillaots design this course as a game, using game elements [Sillaots, 2014]. Clear goals, feedback and collaboration in groups were successfully implemented. In turn, quizzes and scoreboard divided the students, some enjoyed the feeling of competition provided by the scoreboard and some did not. Elements that were not successfully integrated were avatar and luck. After all, the employment of some gamified activities made it possible to create a more active involvement in this course, adding the fun factor to its serious environment.

2.3 User Interface

Another important factor to drive motivation and engagement in a learning app is its visual appeal and ease of use [Pengnate and Sarathy, 2017]. In general, perceived visual appeal significantly influences the perceived ease of use. And perceived visual appeal can affect trust stronger than ease of use. Website design features have a different effect on website trust evaluation for different genders. Men consider both visual appeal and ease of use to evaluate a website trust while women give more attention to the website visual design and less to its ease of use.

As we've seen, the narrative is an important game dynamic. Another important game dynamic is Emotion [Werbach and Hunter, 2012]. Gasah et al. [Gasah et al., 2019] presented a guideline for the Emotional Interface Design. Based on basic emotions classified by their positivity/negativity, User Interface was linked to the emotions it could trigger. The final guideline was built based on Interface Design elements as Layout, Colour and Typeface that trigger positive emotions on children. For example, regarding Colour, red transmitted energy while orange excitement.

Knaving and Björk proposed two sets of design suggestions for a well-designed gamified context [Knaving and Björk, 2013]. The first one concerns the main activity and how gamification should be discrete and imperceptible to not hide the main activity. Thus, the users' focus should not leave the main activity to save their intrinsic motivation. Furthermore, mandatory activities should always be related and meaningful to the main activity to avoid giving the users the sense of being forced to take some actions. The second set of guidelines suggests that to keep users' engagement, not only gamification should be motivating itself, but it also should make users feel competent and autonomous. It also argues that it could support possibilities for playful behaviour since the increase of interaction makes more fun which can develop intrinsic motivations.

As we have seen so far, the use of gamification is increasing fast in this later decade. However, some features should be improved, for instance, the customization of the gamification. Monterrat et al. presented a model to adapt the gamification features taking into account the player profile of the learners, providing them with game features suitable to their personality [Monterrat et al., 2015]. The main goal was to present different interfaces, with different game elements, depending on their profile. For this survey, they considered the BrainHex gamer typology, explained in section 2.1.2, as a player model since it is not related to a specific game genre.

In order to have an interface for each profile, they had to evaluate which game features fit better with each profile. The proposed model estimates how much a game feature fits a player profile, by combining the affinity of the players to the different player types with the weight of the player type with each of the game features. That is, they reach the preferences of the m users for n game features as a product of two matrices: R = B A. The matrix B is a m x k matrix in which its values represent the connection of each one of the m users with the k player profiles, which in this case will be 7. The matrix A is a k x n matrix that contains the weights of the k player profiles for each of the n features.

The values of the matrix B can be obtained through the BrainHex questionnaire . However, the matrix A must be reached by some means. They considered two means:

Human expert derived (1): In this case, experts estimate the values for the matrix A intuitively as they know what motivational factors are involved in each feature.

Empirically from observed data (2): Matrix A can be reached if we know matrix R. And R can be obtained through observation, questionnaires or measuring the preferences of students for the game features. Thus, A can be reached with the following equation:

$$A = (BB^T)^{-1}B^TRB^T \tag{2.1}$$

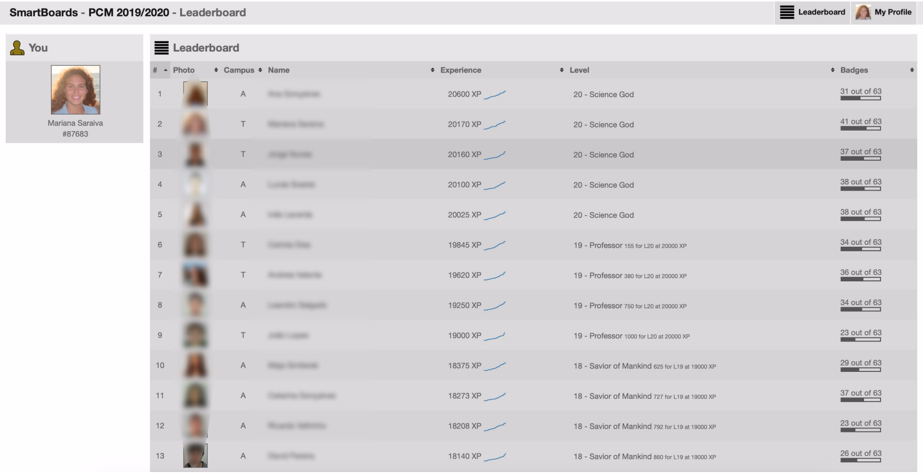
Initially, they analysed the effectiveness of the expert derived vs. the empirically derived. Then, they evaluated the correlation between the real values of the matrix R provided by the users, and the estimated values in both matrices, using the correlation coefficient. Matrix A from experts had a better coefficient (0.2207) than the empirically derived one (0.1822), which lead them to use the matrix A from the experts in the second experiment. This experiment intended to adapt the game features based on the players' profile according to their BrainHex questionnaire. In this second experiment, the students were divided into two groups where students from Group 1 were provided with the two features that fit better with their personality, and students from Group 2 with the two feature that match worse. Therefore, at the end of the experiment, students from Group 1 had spent 38% more time on the learning environment than students from the other group. Nevertheless, the mean values for the item "This feature is motivating for me" (on a scale from 1 to 7) of both groups were surprisingly close (Group 1 - 4.58; Group 2 - 4.55). Thus, it suggests that users seem not to care about the impact of the adapted game features on their experience and engagement.

2.4 Multimedia Content Production

MCP is an MSc course in Information Systems and Computer Engineering at IST. MCP has been a gamified course for almost 10 years, being a target for studying student engagement and behaviour in a gamified system [Barata et al., 2013a] [Barata et al., 2013b].

Currently, MCP assessment is divided into in-class assessment, such as weekly questionnaires and lab assignments, and online, through the Moodle platform, with online assignments, of which not all are mandatory, that is, the students have the freedom to choose what they want to do to achieve their goals. Online assignments are graded along with helpful and useful feedback. All the assessment objects are graded with XP and throughout the semester students are gathering XP that, in the end, will be translated to their final grade (corresponding to their level in the leaderboard).

Gamification has been employed by adding game elements such as a **leaderboard**, where students can see their performance relative to the others, their total **points** and corresponding **level** as well as other students' points and level (Figure 2.7); **badges**, for which students need to complete accomplishments, for instance, attending to theoretical classes, participate in Moodle by providing extra material about each class and show creativity and quality in their online assignments; and a **skill tree**, which is a set of assignments, distributed in four levels, and possible to complete at any time of the semester (Figure 2.8). To achieve level 2, you have to complete the required assignments from level 1 and so on. This allows students to have numerous possible paths to gather the maximum XP of the skill tree. Both badges and skill tree are elements than can be checked in the profile page of each student as well as their performance and progress through the semester (Figure 2.9).



SmartBoards - PCM 2019/2020 - Leaderboard

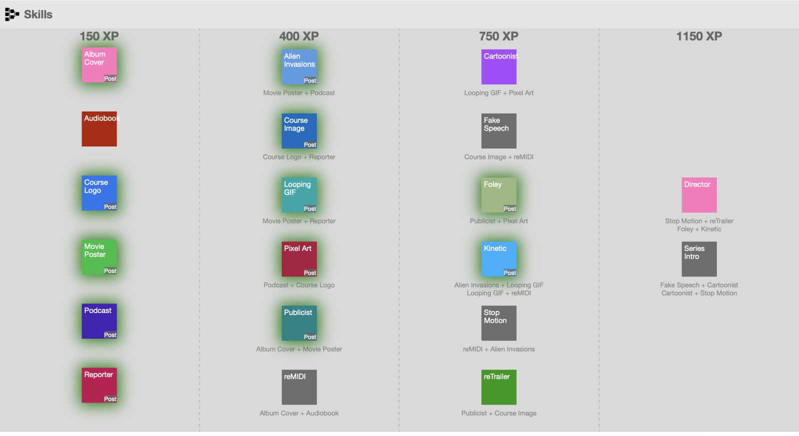
You

Mariana Saraiva
#87683

Leaderboard

#	Photo	Campus	Name	Experience	Level	Badges
1		A	[redacted]	20600 XP	20 - Science God	31 out of 63
2		T	[redacted]	20170 XP	20 - Science God	41 out of 63
3		T	[redacted]	20160 XP	20 - Science God	37 out of 63
4		A	[redacted]	20100 XP	20 - Science God	38 out of 63
5		A	[redacted]	20025 XP	20 - Science God	38 out of 63
6		T	[redacted]	19845 XP	19 - Professor 155 for L20 at 20000 XP	34 out of 63
7		T	[redacted]	19620 XP	19 - Professor 380 for L20 at 20000 XP	36 out of 63
8		A	[redacted]	19250 XP	19 - Professor 750 for L20 at 20000 XP	34 out of 63
9		T	[redacted]	19000 XP	19 - Professor 1000 for L20 at 20000 XP	23 out of 63
10		A	[redacted]	18375 XP	18 - Savior of Mankind 625 for L19 at 19000 XP	29 out of 63
11		A	[redacted]	18273 XP	18 - Savior of Mankind 727 for L19 at 19000 XP	37 out of 63
12		A	[redacted]	18208 XP	18 - Savior of Mankind 792 for L19 at 19000 XP	23 out of 63
13		A	[redacted]	18140 XP	18 - Savior of Mankind 860 for L19 at 19000 XP	26 out of 63

Figure 2.7: Leaderboard of the MCP course



Skills

150 XP

Album Cover Post
Audiobook
Course Logo Post
Movie Poster Post
Podcast Post
Reporter Post

400 XP

Alien Invasions Post
Movie Poster + Podcast
Course Image Post
Course Logo + Reporter
Looping GIF Post
Movie Poster + Reporter
Pixel Art Post
Podcast + Course Logo
Publicist Post
Album Cover + Movie Poster
reMIDI
Album Cover + Audiobook

750 XP

Cartoonist
Looping GIF + Pixel Art
Fake Speech
Course Image + reMIDI
Foley Post
Publicist + Pixel Art
Kinetic Post
Alien Invasions + Looping GIF
Looping GIF + reMIDI
Stop Motion
reMIDI + Alien Invasions
reTrailer
Publicist + Course Image

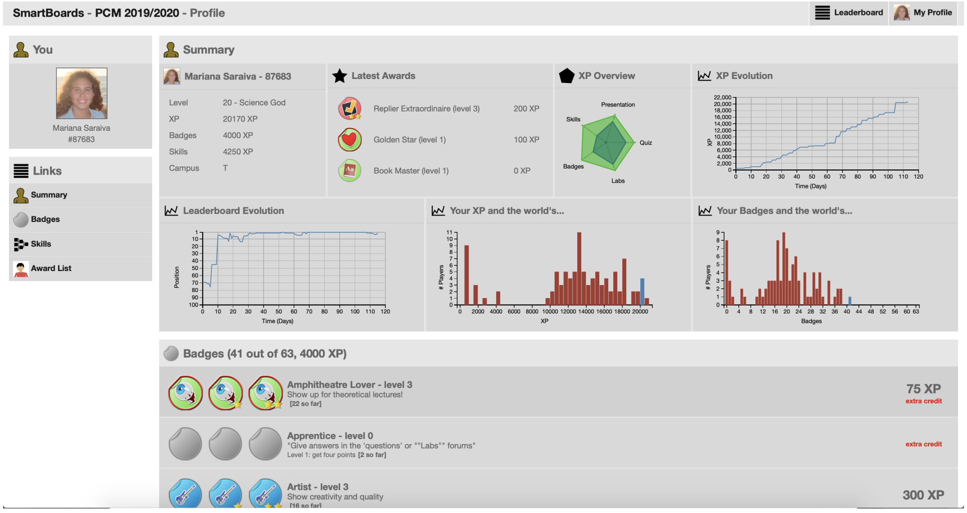
1150 XP

Director
Stop Motion + reTrailer
Foley + Kinetic
Series Intro
Fake Speech + Cartoonist
Cartoonist + Stop Motion

Figure 2.8: Skill Tree

In the early years of MCP as a gamified course, the scoring process was done manually. Data from classes and Moodle were downloaded every day. Besides, the game elements were shown in static web pages created by scripts that had to be run frequently. Many improvements have been done throughout the years to create a better and more automatic system that does not need to run scripts manually.

In 2016, André Baltazar developed SmartBoards, a web application with an improved look for leaderboard and profile pages [Baltazar, 2016]. However, in this version, the faculty still had to run scripts that collect data from Moodle, excel spreadsheets and text files to load it into the SmartBoards system. This web application was used as a basis for Alice Dourado's thesis - GameCourseNext [Dourado, 2019]. Dourado made some improvements to SmartBoards, for instance, restructured the whole system, and added new functionalities of managing the list of students and customizing levels, for example. One of the improvements was to create a relational database to make it more efficient. Regarding the interface design of the GameCourse system, Silva is now working on it, making it more visually appealing [Silva,



Mariana Saraiva
#87683

Mariana Saraiva - 87683

Level	20 - Science God
XP	20170 XP
Badges	4000 XP
Skills	4250 XP
Campus	T

Replier Extraordinaire (level 3)	200 XP
Golden Star (level 1)	100 XP
Book Master (level 1)	0 XP

Badges (41 out of 63, 4000 XP)

Amphitheatre Lover - level 3
Show up for theoretical lectures!
[22 so far]
75 XP
extra credit

Apprentice - level 0
"Give answers in the 'questions' or ""Labs"" forums"
Level 1: get four points [2 so far]
extra credit

Artist - level 3
Show creativity and quality
[16 so far]
300 XP

Figure 2.9: Profile page of an MCP student

2020].

Barata et al. compared the learning experience of students in three years of non-gamified course with the next two years of the gamified course [Barata et al., 2013b]. The impact of gamification turned out to be very positive. In the gamified years, the students were more participative, interested and engaged, since they downloaded considerably more lecture material and there was a significant increase in both average posts per student and the number of reply posts per student. Regarding the grades, the first gamified year was not as successful since the average grade was lower than in the non-gamified years. In turn, in the second gamified year there were a few improvements based on students' feedback from the previous year, such as a more uniform distribution of the workload throughout the semester. This year had the highest average grade, the higher percentage of maximum grades and also the smallest variance. This suggests that feedback is one of the key points to make gamification successful. At the end of the semester, the students filled a questionnaire where they demonstrated more motivation and interest in MCP than in non-gamified courses.

During these first gamified years of MCP, it is known that students' engagement and participation increased. However, the individual experience of the students was never deeply explored [Barata et al., 2016].

In 2016, Barata et al. explored the behavioural and performance patterns of the students, during three gamified years [Barata et al., 2016]. They found out six clusters of students:

Achievers, who want to be the best and collect every single badge. Their XP accumulation was the highest, revealing to be the more proactive students. **Underachievers**, who had the worst performances, being in the bottom of the leaderboard. They did the minimum to pass the course, ignoring several of its components. Achievers and Underachievers were observed in all three years. **Disheartened** students began along with Achievers, but in the middle of the semester, their performance decreased, and the majority were in the middle of the leaderboard. This cluster only existed for the first two years. In the second year, a new cluster emerged - **Late Awakeners**. These students are the inverse of the Disheartened students, that is, they started in lower positions but then their performance increased reaching a better status in the leaderboard. **Regular Students** appeared only in the third year due to improvements to the course. They achieve the second-best performance, yet above the average grade and finally the **Halfhearted students**, that also only emerged in the last year, who had a performance better than the Underachievers but bellow the Regular Students, which leaves them with a below-average grade. These students seemed to be the least engaged with the course.

Therefore, this work intends to create different profile pages to different types of students. For this, it is important to understand what stimulates them to provide an enjoyable experience, increase their participation and learning outcomes.

2.5 Discussion

Throughout the section, we discussed how gamification has been showing its benefits. It is deeply important to notice that gamification has to be seen as an instrument to increase engagement and motivation rather than the solution for the lack of it, that is, the gamified context itself may arouse intrinsic motivation in the users [Deterding, 2012]. To reach a successful gamified context, continuous improvements are required, based on user needs, in order to keep their motivation. It is possible to guarantee engagement and motivation if users see the results of their behaviour and receive feedback, which will influence their behaviour and their intention to continue participating [Hassan et al., 2019]. Besides feedback, it is deeply important to balance user skills with the proposed challenges. Maintaining every student in their flow zone is one of the biggest challenges when designing a gamified context.

As a pioneer in gamification, The Multiplayer Classroom model created by Sheldon revealed unquestionable successful [Sheldon, 2011]. The implementation of this new teaching model including experience points and a different grading system allowed better tracking of students' progress by the teacher. In the end, this experiment showed great results both in student engagement and students' final grades.

We have seen that game elements do not have the same impact on every player, that is, game elements can trigger either a positive or a negative emotion, which is an important dynamic [Werbach and Hunter, 2012]. In fact, Gasah et al. presented a survey in which they use emotion to find out which elements trigger positive emotions in children, demonstrating that different colours and different layouts trigger different emotions [Gasah et al., 2019]. Regarding the game elements, on one hand, the leaderboard has a significant impact on players whose extrinsic motivation is to see their position related to the others. On the other hand, badges and rewards are extremely motivating elements for players whose goal is to gather badges and achievements (also known as Achievers). Thus, to create a gameful experience for the majority of the students, and avoiding losing some students in the middle of the semester, it is essential to know the students beforehand by evaluating their player types resorting to questionnaires, for example, the BrainHex questionnaire. This would help to create that engaging and pleasant environment that students need to have a great experience, and in which they feel it takes into account their motivations and personality.

Gamification has proven its ability to help to transform a serious environment in something fun and attractive, and MCP course is a great example of that, with some relevant studies in student engagement [Barata et al., 2013b]. Even so, the customization of the interface is the next big step. This adaptation of some game features to different types of students is fundamental to generate more motivation in students. On the one hand, we can attract underperforming students with the features that motivated them to increase their performance. On the other hand, we can keep high-performance students by giving them their points and their position on the leaderboard, for example, which makes them want to keep up. One of the means to achieve this customization relies on gather information about each student and evaluate the affinity of their personality with each type of player [Monterrat et al., 2015]. Then, is necessary to combine this information with the weight of the player type with each of the game features. Hence, we can create a personalized interface for each student to make them feel engaged within the environment, with features suitable to their motivations. However, the students' perception of the adaptation of the game features presented in their interface seems not to impact their motivations.

By implementing the mechanisms for customization and adaptation of the interface, we aim to help teachers in managing the courses and their views in order to try to achieve better results in the participation and engagement of the students.

3

The GameCourse system

Contents

The GameCourse system, resorting to some gamification features, allows us to create and manage courses. This system has been in use for the last five years in the MCP course, partially replacing a previous, script-based approach. Since its first version, many adjustments and improvements have been done. Nevertheless, these features refer mainly to the back end, which left the front-end with a poor and unattractive appeal. For this reason, the UI has been improved during the last year by Patricia Silva [Silva, 2020]. This new design, better user experience, and responsiveness are now visible. Our goal is to continue this improvement and create a better mechanism for the management of the courses. In this chapter we will approach some of the recent improvements to the system as well as its state when we started our work, to contextualize the work on chapter 4.

3.1 The GameCourse UI

The main technology used to develop the UI of the system is AngularJS[1]. It takes care of the interaction between pages through states, controllers, and scopes. On states, it is defined the routing of the GameCourse system. We define which pages are available to navigate and through which URLs. On the controllers, the pages' HTML structure is generated and the needed requests to the API to obtain the desired data are made. This data is stored in the scope, managed using scope functions, and associated with GUI elements, which enabled automatic re-renders when the associated information changes.

The front-end part of the project includes, essentially:

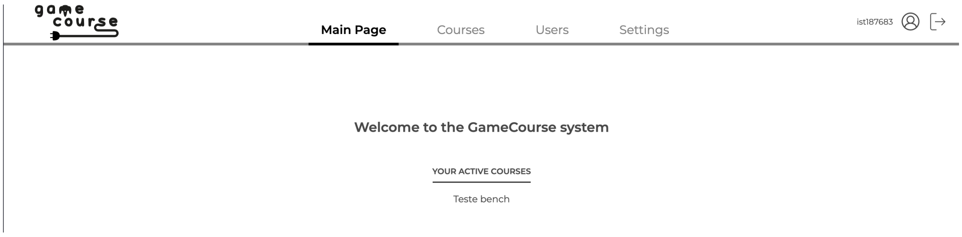
- Controllers, which are JavaScript files that build the pages with the information gathered from the Application Program Interface (API). The API functions are defined in PHP files.
- Other JavaScript files that create dynamic pages.
- Images, Cascading Style Sheets (CSS) files, and other user handlers.

The back-end stores and manages all the information, such as the users data, through a MariaDB database (Appendix A), PHP classes (version 5.6), and other external sources.

The website itself has a top navigation bar with the available pages, and on the right a button to our profile. When entering the website as a professor, we find the system navigation bar. This bar includes the `Main Page` tab as the welcome page, the `Courses` tab, in which we can see the courses we have access to, the `Users` tab, where we can see all the users of the system, and the tab of the global `Settings` of the system (Figure 3.1).

On the `Courses` and (system) `Users` pages we can add new ones, import more data such as new courses or new users, export a file with that information, filter and change the order in which it

[1] https://angularjs.org/

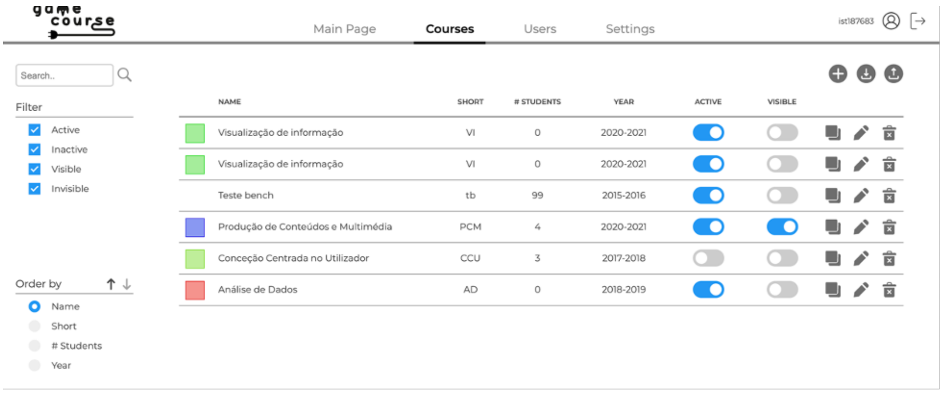


Welcome to the GameCourse system

YOUR ACTIVE COURSES

Teste bench

Figure 3.1: View after entering the system as a professor



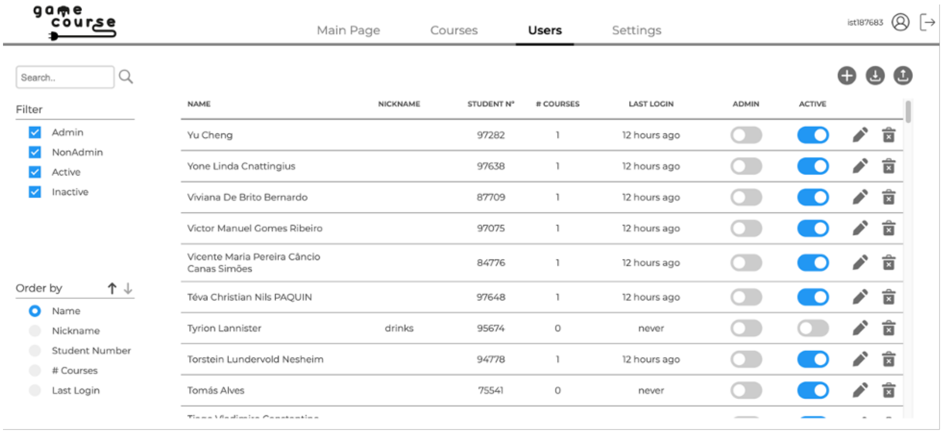
☑ Active
☑ Inactive
☑ Visible
☑ Invisible

● Name
○ Short
○ # Students
○ Year

	NAME	SHORT	# STUDENTS	YEAR	ACTIVE	VISIBLE
🟩	Visualização de informação	VI	0	2020-2021	●	○
🟩	Visualização de informação	VI	0	2020-2021	●	○
	Teste bench	tb	99	2015-2016	●	○
🟦	Produção de Conteúdos e Multimédia	PCM	4	2020-2021	●	●
🟩	Conceção Centrada no Utilizador	CCU	3	2017-2018	○	○
🟥	Análise de Dados	AD	0	2018-2019	●	○

Figure 3.2: `Courses` page

is presented (Figures 3.2 and 3.3). Besides, we can edit each instance data, for example, the short name of a course or, in the case of the users, their nickname or their email. We can also change some individual features, such as whether it is active or not, and delete a course (or user). Regarding the import feature, right now it has the replace duplicates option set by default as true, on course users and users, and false on courses.



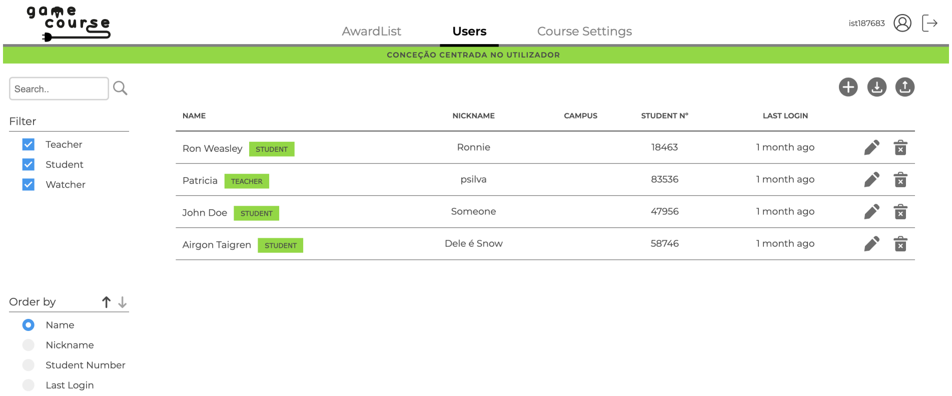
☑ Admin
☑ NonAdmin
☑ Active
☑ Inactive

● Name
○ Nickname
○ Student Number
○ # Courses
○ Last Login

NAME	NICKNAME	STUDENT Nº	# COURSES	LAST LOGIN	ADMIN	ACTIVE
Yu Cheng		97282	1	12 hours ago	○	●
Yone Linda Cnattingius		97638	1	12 hours ago	○	●
Viviana De Brito Bernardo		87709	1	12 hours ago	○	●
Victor Manuel Gomes Ribeiro		97075	1	12 hours ago	○	●
Vicente Maria Pereira Câncio Canas Simões		84776	1	12 hours ago	○	●
Téva Christian Nils PAQUIN		97648	1	12 hours ago	○	●
Tyrion Lannister	drinks	95674	0	never	○	○
Torstein Lundervold Nesheim		94778	1	12 hours ago	○	●
Tomás Alves		75541	0	never	○	●

Figure 3.3: System `Users` page

Once inside a course, the navigation bar presents the pages of the course (in this case, the `AwardList`), the `Users` tab, in which we can see all the users (students and professors) of that course, and the tab of the `Settings` of the course, depicted in Figure 3.4. This `Users` page has the same features as the system users, described above. However, in this page, we can see the role each user has in that course, for example, whether it is a teacher or a student, for example.



NAME	NICKNAME	CAMPUS	STUDENT Nº	LAST LOGIN
Ron Weasley STUDENT	Ronnie		18463	1 month ago
Patricia TEACHER	psilva		83536	1 month ago
John Doe STUDENT	Someone		47956	1 month ago
Airgon Taigren STUDENT	Dele é Snow		58746	1 month ago

Figure 3.4: Course `Users` page

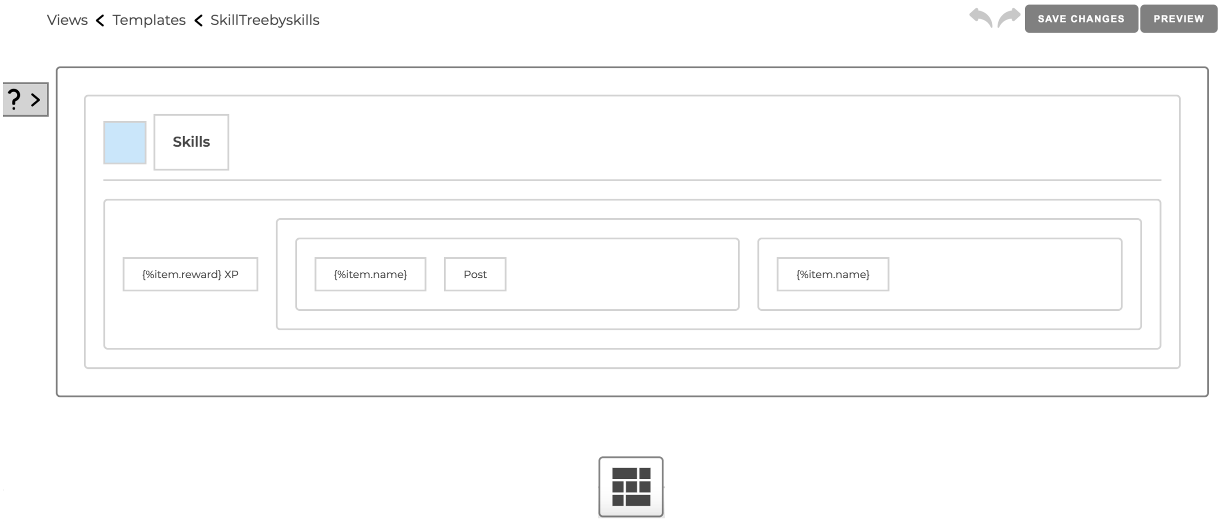


Figure 3.5: View editor

To better understand how the GameCourse system works, regarding the UI, we first need to be aware of the concepts of **views** and **pages**. Views are built with the help of a view editor (Figure 3.5), that has different layout elements available (Figure 3.6), and can be customized using the system's Expression Language (EL), which allows us to present the views with dynamic content. A **view** can be a single element, such as an image or text. By compounding these small views, we can create more complex views. For example, we can create a list of images by repeating one single view previously built. When we have a view, we can choose to show it as a **page** with a certain name. Then, this page can be available on the navigation bar whose content is that view. As we can see in Figure 3.4, the `AwardList` page is an example of a page whose content was created in the view editor.

Regarding the EL, this is structured with multiple libraries, each with its own namespace, which provides its functions that return value, objects (a data structure containing values), and collections (a data structure containing any number of objects). By using an expression, it will be evaluated and return different values for different students. For example, the expression **Hello** $\{$**users.getUser(%viewer).name**$\}$**!** would be evaluated as **'Hello John!'**, for example, if the name of the user was John. Some of the vocabulary in the EL comes from the different modules that can be activated for each individual course. Those modules typically provide a game element and make available to the EL the functions required to properly manage that game element.

The view editor also supports building different views for different types of students resorting to **Aspects**, different renderings of a view meant for students of different profiles. The system automatically selects the more suitable aspect for each user when he/she enters. There are two types of aspects for views:

- **Role - Single**: a view that can show a different aspect according to the viewer's role.
- **Role - Interaction**: a view that can show a different aspect according to the roles of the viewer and the user associated with the page. For example, a profile page will have as user the person whose information is displayed and the viewer is the person who is seeing the page.

Regarding the view editor, it could be reached through two different pages. First, on the views page (Figure 3.7), we choose the page whose layout we want to edit. Then, on the next page (Figure 3.8), we choose which role(s) that layout is for. After clicking the intended role, we go to the view editor, where we can build the layout seen by the chosen role (Figure 3.5).

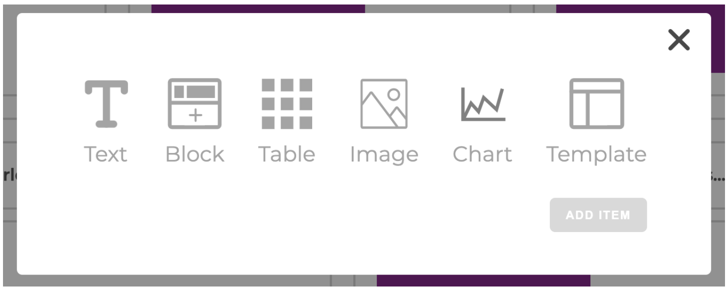


Figure 3.6: Different layout elements

Inside the view editor, when we select a part of the layout, some buttons appear at the bottom of the page. By clicking on these buttons, we can edit fields, change the layout, delete items, switch a part of it for another possible, duplicate a part and save the layout as a view template, respectively (Figure 3.9). View templates can be later reused as parts of more complex views.

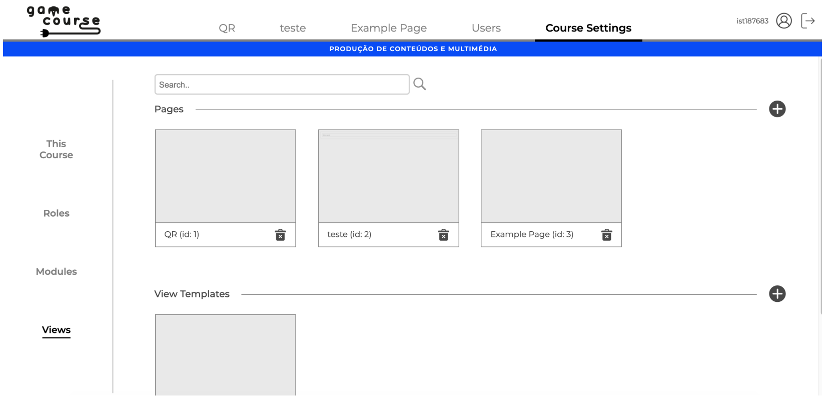
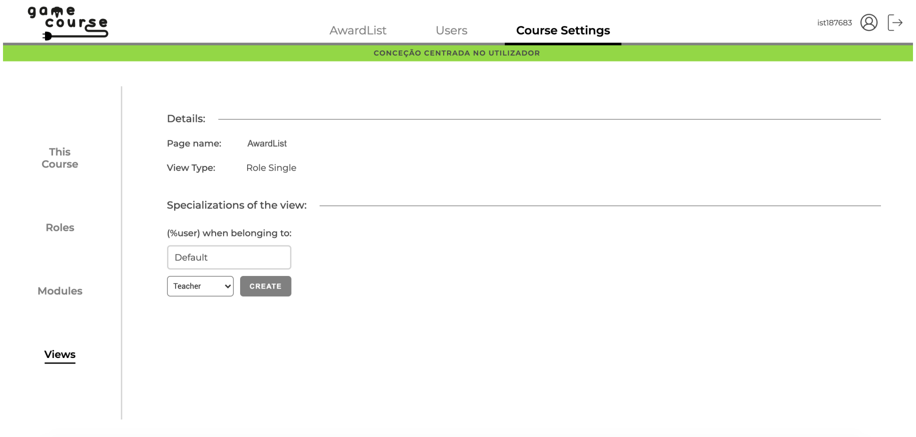


Figure 3.7: Views page



Details:

Page name: AwardList

View Type: Role Single

(%user) when belonging to:

Figure 3.8: Intermediate page to choose the role

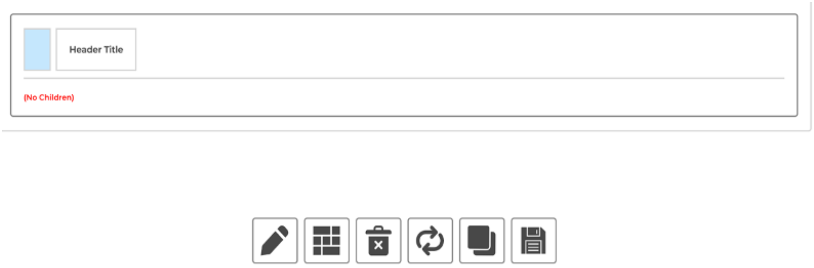


Figure 3.9: View editor with a block selected

3.2 Requirements

Before starting the implementation of the personalized views, some errors and bugs were identified as needing to be fixed, as well as completing the remainder interface changes. So, we decided to organize our tasks by priority, as described below.

High priority tasks

- **Fix existing bugs**: A few bugs were unsolved, such as the routing of the landing page, and the enabling of the modules. These bugs and many other are now solved.
- **Import improvement**: The import feature (in the system users, courses, and course users pages) has to be improved to allow users to choose if they want to replace the duplicates or ignore them. Besides, the configuration of the modules would leverage with this feature as well, to avoid configuring them one element at a time.
- **Views page**: We will need to add a new field for pages (enabled), to show or remove from the navigation bar the ones which are enabled or not.
- **View editor**: The view editor has some issues regarding its usability and how easy it is to work with. In order to give better usability and better support, we proposed:
  - Remove the intermediate page: the choice of the role will be on the view editor page.
  - Incorporate a tutorial for the view editor.
  - Provide auto complete when writing the EL.
  - Improve the preview section.

Medium priority tasks

- **Theme feature**: It will be possible to import a CSS file to override the styles on the course.
- **Mobile version**: We will make the design to be supported by a mobile screen, especially regarding the navigation bar, sidebars, and font-size.

Low priority tasks

- **My Information page**: Add the option to edit some fields.
- **Global settings**: It will be added a section to change the order of the pages on the navigation bar.
- **Navigation bar**: The highlights on the navigation bar of the pages created through the view editor and the users' page need to be fixed.

We could fulfill all the requirements except the mobile version of the system. Although it had medium priority, the low priority tasks were quite simple to execute, unlike that one. So, we were able to finish the low priority tasks. Besides, many other tasks were asked to be done throughout the semester while the professors used the system. The MCP teachers found it important to have a more useful configuration page for some modules, for example.

4

Development

Contents

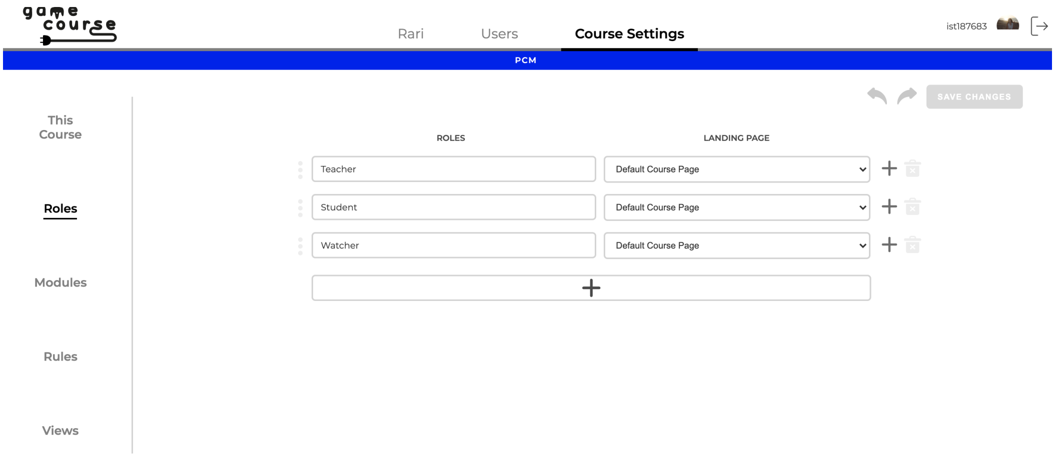
In this chapter, we will explain the whole developing process regarding the improvement of GameCourse.

4.1 Preliminary Work

First, we needed to have a stable version of GameCourse from which we could start our implementation of the new features or the necessary improvements. Thus, we started by fixing the existing small bugs. Besides, as this was done while I was writing my thesis project and my requirements were not clearly defined, I also did some improvements (required/suggested by my previous colleagues), for example, the improvement of the import feature.

4.1.1 Bug fixes

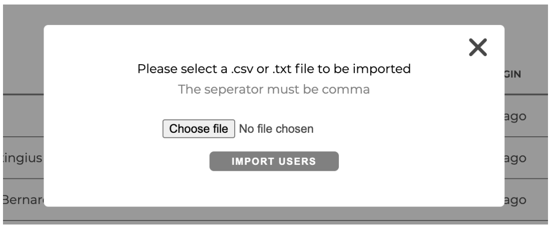
As mentioned before, we started our development earlier, by fixing some bugs and doing small tasks. The first bug fix was regarding the routing of the landing page. In the courses settings, we can define which page is the landing page for each role (Fig. 4.1). When the landing page was set as the Default Landing Page, this routing was working. Otherwise, we could not enter a course since the error redirected us to the system home page. We fixed this bug by including, not only the name of the page (as it was), but also its ID in the URL.



ROLES	LANDING PAGE
Teacher	Default Course Page
Student	Default Course Page
Watcher	Default Course Page

Figure 4.1: Roles page.

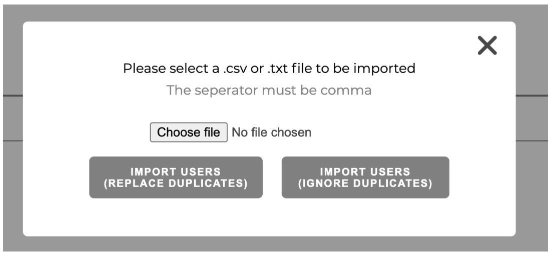
Another required fix was concerned about the view templates. We could turn them into globals, but we could not reverse this action due to an error when introducing the flag *isGlobal*. Also due to a similar error, another fix was needed. In this case, when we enabled the badges module, i.e. when we activated it, this information was not being inserted into the database if we had more than one course. In other words, it was not being created a new table entry when the module is enabled, unless for the first course.



Please select a .csv or .txt file to be imported

The seperator must be comma

Figure 4.2: Import modal before improvements.



Please select a .csv or .txt file to be imported

The seperator must be comma

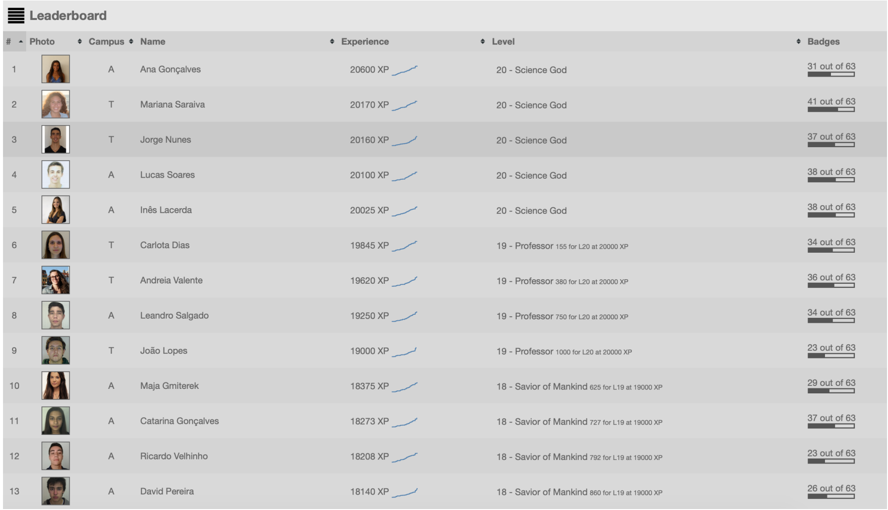
Choose file
No file chosen
IMPORT USERS (REPLACE DUPLICATES)
IMPORT USERS (IGNORE DUPLICATES)

Figure 4.3: Import modal after improvements.

Besides, we also worked on other small tasks. The first one was improving the user import functionality. There was only one button in the import modal (Figure 4.2), and we wanted to give the user the choice of replacing data. For example, if the imported csv file has new users and/or updated information about users already in the system, we would want to add only the new users and update the data regarding the other users. On the other hand, we may want to ignore the new data in the imported csv and not replace the information in our database. Therefore, we added another button to the modal to allow users to choose whether they want to replace it or not, as we can see in Figure 4.3. This improvement was made to the course page, users' page, and course users' page.

The second task was to move the field *campus* from the `course_user` table to the `game_course_user` table, in the database. The `course_user` table saves in which course each user is enrolled, while the `game_course_user` table saves every user that have access to the system. As since many professors from IST, including other departments, and other universities, have been showing their interest in using GameCourse, we changed it to *major*, so we can specify the major of each user. This change forced us to ensure that every function that used this field kept working correctly, such as import and export users. In addition, we added this field to the New User and Edit User modals, both inside the course and in the system users page. Regarding the view templates generated by the modules, some had their content inaccurate, particularly the following modules: AwardList, Leaderboard, Profile and Badges. Since the MCP course was about to start, we had to make sure that those generated were accurately generated and that the resulting view was the one we wanted. Withal, we made a few improvements to their style/look:

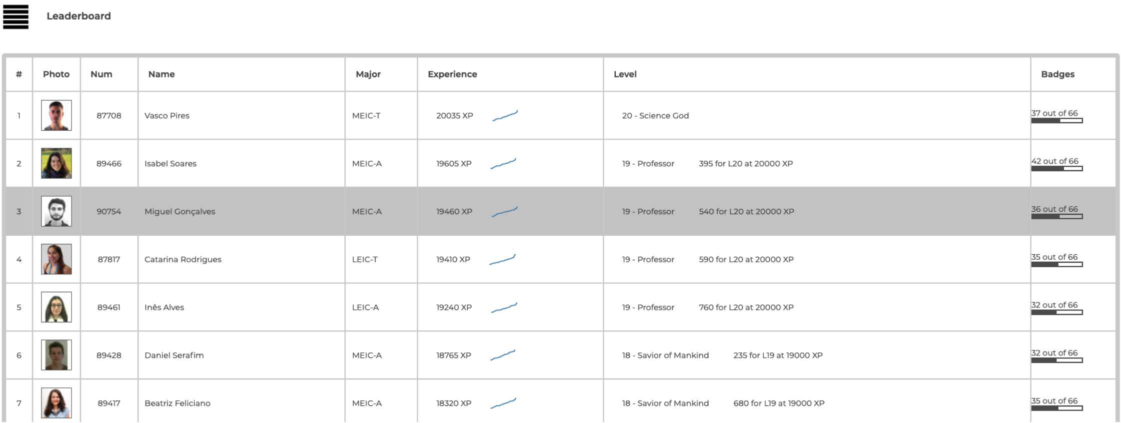
- Changing style and color of tables, present in the leaderboard (Figures 4.4 and 4.5) and award list view templates;
- Changing the background from the skills view template;
- In the profile view template, we included boxes to better understand the page layout/sectioning;
- Adding some missing elements, for example, the awards image in the award list and latest awards view templates.



Leaderboard

#	Photo	Campus	Name	Experience	Level	Badges
1		A	Ana Gonçalves	20600 XP	20 - Science God	31 out of 63
2		T	Mariana Saraiva	20170 XP	20 - Science God	41 out of 63
3		T	Jorge Nunes	20160 XP	20 - Science God	37 out of 63
4		A	Lucas Soares	20100 XP	20 - Science God	38 out of 63
5		A	Inês Lacerda	20025 XP	20 - Science God	38 out of 63
6		T	Carlota Dias	19845 XP	19 - Professor 155 for L20 at 20000 XP	34 out of 63
7		T	Andreia Valente	19620 XP	19 - Professor 380 for L20 at 20000 XP	36 out of 63
8		A	Leandro Salgado	19250 XP	19 - Professor 750 for L20 at 20000 XP	34 out of 63
9		T	João Lopes	19000 XP	19 - Professor 1000 for L20 at 20000 XP	23 out of 63
10		A	Maja Gmiterek	18375 XP	18 - Savior of Mankind 625 for L19 at 19000 XP	29 out of 63
11		A	Catarina Gonçalves	18273 XP	18 - Savior of Mankind 727 for L19 at 19000 XP	37 out of 63
12		A	Ricardo Velhinho	18208 XP	18 - Savior of Mankind 792 for L19 at 19000 XP	23 out of 63
13		A	David Pereira	18140 XP	18 - Savior of Mankind 860 for L19 at 19000 XP	26 out of 63

Figure 4.4: Leaderboard page with the previous styling.



Leaderboard

#	Photo	Num	Name	Major	Experience	Level	Badges
1		87708	Vasco Pires	MEIC-T	20035 XP	20 - Science God	37 out of 66
2		89466	Isabel Soares	MEIC-A	19605 XP	19 - Professor 395 for L20 at 20000 XP	42 out of 66
3		90754	Miguel Gonçalves	MEIC-A	19460 XP	19 - Professor 540 for L20 at 20000 XP	36 out of 66
4		87817	Catarina Rodrigues	LEIC-T	19410 XP	19 - Professor 590 for L20 at 20000 XP	35 out of 66
5		89461	Inês Alves	LEIC-A	19240 XP	19 - Professor 760 for L20 at 20000 XP	32 out of 66
6		89428	Daniel Serafim	MEIC-A	18765 XP	18 - Savior of Mankind 235 for L19 at 19000 XP	32 out of 66
7		89417	Beatriz Feliciano	MEIC-A	18320 XP	18 - Savior of Mankind 680 for L19 at 19000 XP	35 out of 66

Figure 4.5: Leaderboard page after styling adjustments.

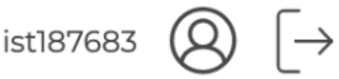


Figure 4.6: User icon with generic icon.

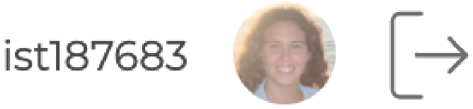


Figure 4.7: User icon with profile photo.

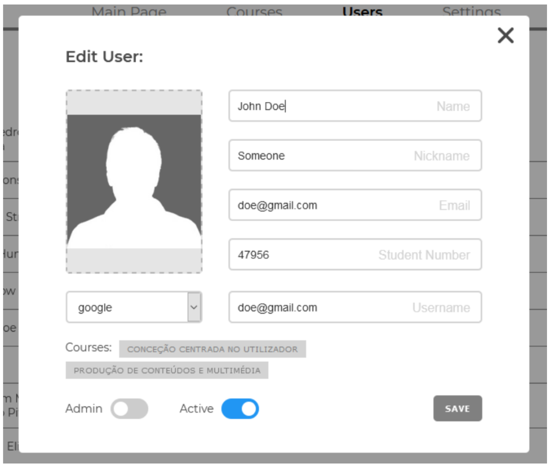
Finally, we decided to replace the generic user icon (on the top right of the page) with their photo. These changes can be seen in Figures 4.6 and 4.7. Throughout the project, and as we were working on different parts of the project, many other bugs were found and fixed, such as the text files used to generate the view templates from the modules. Some of them were not well built. The system was being used during the semester which gave us the opportunity to test with real users. This provided us with very useful feedback and bugs we were not aware of, which helped us to improve the system even more.

4.1.2 Moodle and Test Environment

The previous system had some issues two weeks before the second semester (2020/2021) started, which had to be fixed for it to be ready to use in that semester. Therefore, in February (classes started on March 1st, 2021), we had to fix all the bugs that would compromise the functioning of the system. Moreover, since the new system would be used, we had to upgrade Moodle in the Virtual Machine. After we achieved this, we realized that we needed an environment in which we could deploy our new work, but this could be the same that was in use for the MCP course. Thus, we created a test environment in our Virtual Machine.

4.2 UI General improvements

In the course user and user modals – New User and Edit User – there were some issues regarding the UI. The first one was that the course and roles tags presented in the system domain and course domain, were not readable. Therefore, we increased the contrast by changing the background to a darker grey and the text color to white. Another issue was regarding the selection of the auth service. The text color was always the same regardless whether an option is selected or not. We tackled this by adding a condition to verify if it was selected an option or not. If not, the text color is light grey, otherwise is black. Thus, it is noticeable the difference. Finally, the student number box was bigger for the possible values, since it is a number with 5 or 6 digits. Besides, as mentioned before, we added the major field, which has maximum 6 characters. Therefore, we used one row for both, one on the left and the other on the right.



Edit User:

John Doe — Name

Someone — Nickname

doe@gmail.com — Email

47956 — Student Number

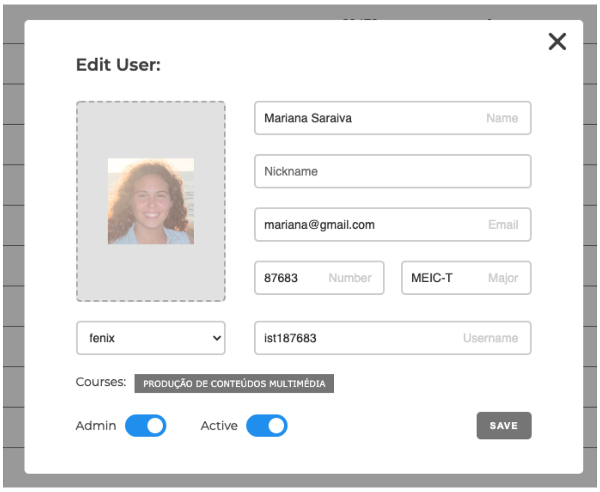
google

doe@gmail.com — Username

Courses: CONCEÇÃO CENTRADA NO UTILIZADOR PRODUÇÃO DE CONTEÚDOS E MULTIMÉDIA

Admin ○ Active ●

Figure 4.8: Edit user modal before.



Edit User:

Mariana Saraiva — Name

Nickname

mariana@gmail.com — Email

87683 — Number

MEIC-T — Major

fenix

ist187683 — Username

Courses: PRODUÇÃO DE CONTEÚDOS MULTIMÉDIA

Admin ● Active ●

Figure 4.9: Edit user modal after styling adjustments.

4.2.1 File Explorer

To allow the user to access the files in the server, or to upload new ones, we created a modal from scratch in which the user can either upload a file from their computer or browse files saved on the server side (Figures 4.10 and 4.11).

On the upload tab, when the user uploads a file, it will be saved on server side, and it will appear below. By clicking it, it will be selected (Figure 4.12) and then the user can delete it (clicks on the delete button on bottom left) or select it to include it on the page (clicks the select button on bottom right). On the other hand, if the user wants to choose another one, he/she must upload another file of their choice.

On the browser tab, the user can only browse files from the current course. In this folder, the user can navigate through every folder in the respective course. When a course is created, it is created a folder for it. Then, as the modules are activated and configured, their folder are created to save all the information. For example, the badges folder, in which we keep every badge with the respective images; the skills folder, where are saved every skill resources and the corresponding HTML file.

This file explorer can be used in every page of the GameCourse. However, in some cases, not every file extension can be chosen. Thus, the developer is accountable for allowing which type of files are acceptable in each case by showing only the permitted ones, by provide a list of allowed extensions to the function that will instantiate the file explorer.

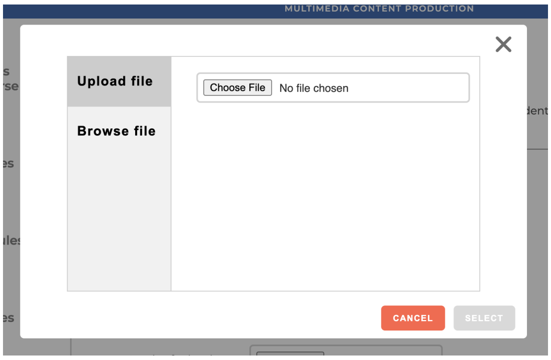


Figure 4.10: Upload file tab.

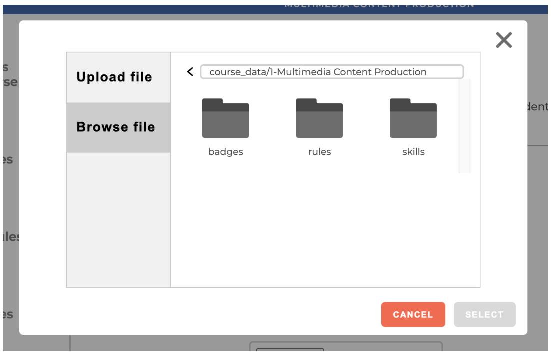


Figure 4.11: Browse file tab.

4.3 Module Configuration

The modules' configuration had a poor UI, and some configurations were simple text boxes where the user must copy and paste a text file to them, typically in csv format. This part of GameCourse truly needed improvements in its UI, and also UX. Besides, each module had its own controller, and sometimes there was duplicated code. Thus, we decided to use one single Controller (Configuration Controller) for most of the configurable modules which had a similar configuration. For example, XP and

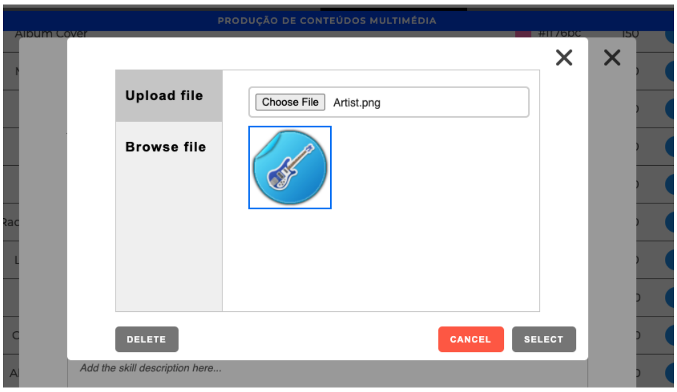

PRODUÇÃO DE CONTEÚDOS MULTIMÉDIA
Upload file
Browse file
Choose File
Artist.png
DELETE
CANCEL
SELECT
Add the skill description here...

Figure 4.12: File pick modal after uploading a file and selecting it.

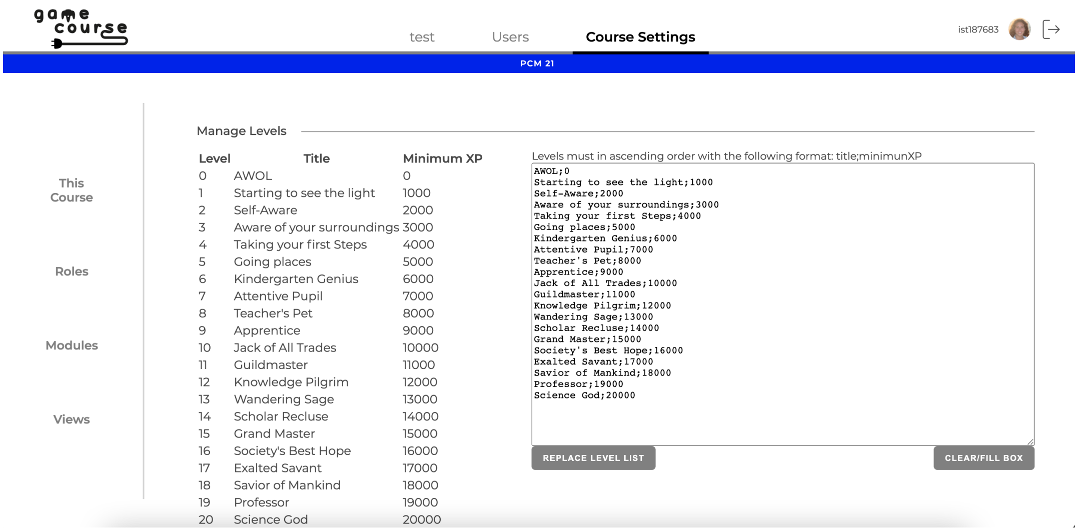
Levels, Badges and Skills all have a list of general items - levels, badges, and skills, respectively.

4.3.1 XP and Levels

This module configuration page was composed of a text box on the right where we pasted the content of a text file with a button below to confirm the data and another one to clear the box. On the left, we could check the values introduced, as we can see in Figure 4.13. In order to keep the system consistent, and give a better look, as well as UX and usability, we changed this page (Figure 4.14). We used the generic Configuration Controller, which has three possible sections: General Inputs, Listing Items and Personalized section. For this module, we only needed one of the them: listing items. In the Levels module, we defined in the function *get_listing_items* the name of the list, in this case, Levels, the name of the columns (Level, Title, and Minimum XP), the items to show, and the corresponding attributes of each item to each column. The items were given through a database query. Then, we defined five new features, two of which are general features that affect all the items, and the other three are individual features.

- Import Levels - by clicking on the import button, the user can choose a text file from his/her own computer, and import it. Then, the database is updated, as well as the list on the page.
- Export Levels - this button calls a function that will generate a text file with all the information of every level (level number, title, and minimum XP) of the course and then, download it.
- Add Level - by including an add button, we allow the user to add one level to the list. By clicking it, a modal is shown and the user can fill in all the information of the new element. This information will then be stored in the database.
- Edit Level - we included an edit button in each row. By clicking it, a modal is shown and the user can edit information regarding only that element, which will be updated in the database.

- Delete Level - finally, we have included a delete button in each row. If the user desires to delete a level, he/she clicks on it. Then, a verification modal is shown asking if the user is sure about deleting it.



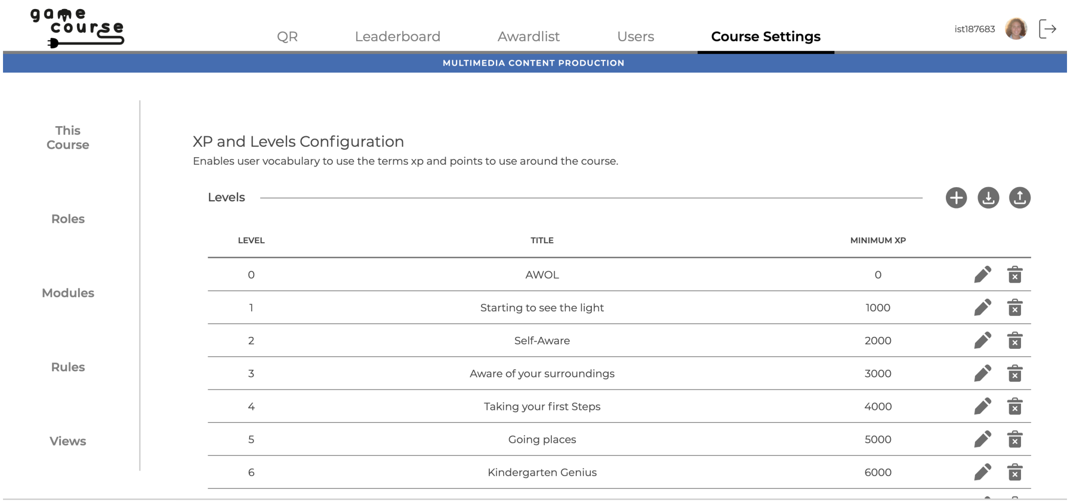
Manage Levels

Level	Title	Minimum XP
0	AWOL	0
1	Starting to see the light	1000
2	Self-Aware	2000
3	Aware of your surroundings	3000
4	Taking your first Steps	4000
5	Going places	5000
6	Kindergarten Genius	6000
7	Attentive Pupil	7000
8	Teacher's Pet	8000
9	Apprentice	9000
10	Jack of All Trades	10000
11	Guildmaster	11000
12	Knowledge Pilgrim	12000
13	Wandering Sage	13000
14	Scholar Recluse	14000
15	Grand Master	15000
16	Society's Best Hope	16000
17	Exalted Savant	17000
18	Savior of Mankind	18000
19	Professor	19000
20	Science God	20000

Levels must in ascending order with the following format: title;minimunXP

```
AWOL;0
Starting to see the light;1000
Self-Aware;2000
Aware of your surroundings;3000
Taking your first Steps;4000
Going places;5000
Kindergarten Genius;6000
Attentive Pupil;7000
Teacher's Pet;8000
Apprentice;9000
Jack of All Trades;10000
Guildmaster;11000
Knowledge Pilgrim;12000
Wandering Sage;13000
Scholar Recluse;14000
Grand Master;15000
Society's Best Hope;16000
Exalted Savant;17000
Savior of Mankind;18000
Professor;19000
Science God;20000
```

Figure 4.13: Previous levels configuration page.



XP and Levels Configuration

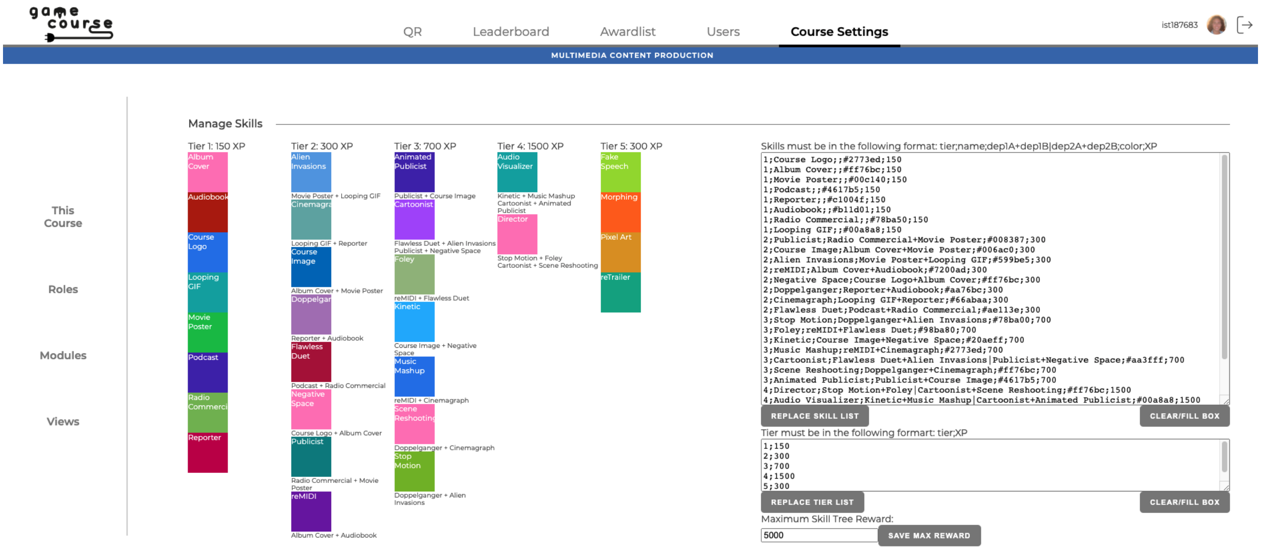
Enables user vocabulary to use the terms xp and points to use around the course.

LEVEL	TITLE	MINIMUM XP
0	AWOL	0
1	Starting to see the light	1000
2	Self-Aware	2000
3	Aware of your surroundings	3000
4	Taking your first Steps	4000
5	Going places	5000
6	Kindergarten Genius	6000

Figure 4.14: New levels configuration page.

4.3.2 Skills

This module configuration page was similar to the levels page. It was composed of three text boxes on the right - one for the skills, one for the tiers, and one for setting up the maximum skill tree reward. In the first two, we could paste the content of a text file, and for each, we had a button below to confirm the data and another one to clear the box. In the bottom box, we wrote the number and then saved it by clicking the save button. On the left, after saving all the configurations, we saw the graphic visualization of the configured skill tree (Figure 4.15).



Skills must be in the following format: tier;name;dep1A+dep1B|dep2A+dep2B;color;XP

```
1;Course Logo;;#2773ed;150
1;Album Cover;;#ff76bc;150
1;Movie Poster;;#00c140;150
1;Podcast;;#4617b5;150
1;Reporter;;#c1004f;150
1;Audiobook;;#b11d01;150
1;Radio Commercial;;#78ba50;150
1;Looping GIF;;#00a8a8;150
2;Publicist;Radio Commercial+Movie Poster;#008387;300
2;Course Image;Album Cover+Movie Poster;#006ac0;300
2;Alien Invasions;Movie Poster+Looping GIF;#599be5;300
2;reMIDI;Album Cover+Audiobook;#7200ad;300
2;Negative Space;Course Logo+Album Cover;#ff76bc;300
2;Doppelganger;Reporter+Audiobook;#aa76bc;300
2;Cinemagraph;Looping GIF+Reporter;#66abaa;300
2;Flawless Duet;Podcast+Radio Commercial;#ae113e;300
3;Stop Motion;Doppelganger+Alien Invasions;#78ba00;700
3;Foley;reMIDI+Flawless Duet;#98ba80;700
3;Kinetic;Course Image+Negative Space;#20aeff;700
3;Music Mashup;reMIDI+Cinemagraph;#2773ed;700
3;Cartoonist;Flawless Duet+Alien Invasions|Publicist+Negative Space;#aa3fff;700
3;Scene Reshooting;Doppelganger+Cinemagraph;#ff76bc;700
3;Animated Publicist;Publicist+Course Image;#4617b5;700
4;Director;Stop Motion+Foley|Cartoonist+Scene Reshooting;#ff76bc;1500
4;Audio Visualizer;Kinetic+Music Mashup|Cartoonist+Animated Publicist;#00a8a8;1500
```

Tier must be in the following formart: tier;XP

```
1;150
2;300
3;700
4;1500
5;300
```

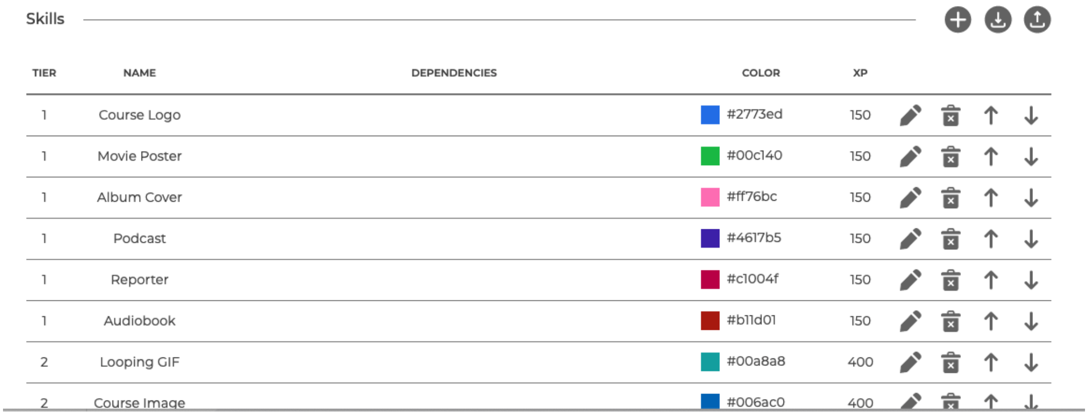
Maximum Skill Tree Reward: 5000

Figure 4.15: Previous skills configuration page.

Once again, in order to keep the system consistent, and give a better look, as well as User Experience and usability, we changed this page (Figure 4.16). By using the generic Configuration Controller, we used the three possible sections: generic items - for the maximum reward -, listing items - for the skills, and the personalized section - for the tiers. Besides, we added the graphic visualization of the skill tree, as we had before, in order to check our configuration. In the Skills module, for the general item, we defined the function *get_general_items*, which, in this case, will return the information only regarding the maximum XP. For the skills, we defined in the function *get_listing_items* the name of the list, in this case, Skills, the name of the columns (Tier, Name, Dependencies, and Color), the items to show, and the corresponding attributes of each item to each column. The items were given through a database query. For the tiers, we defined a personalized function - *get_tiers_items* - that returns the information for the Tiers table, similar to the skills' but the name of the columns are Tier and XP. For these two lists we added a new feature – reordering. By adding two columns to the lists, the user can manage and reorder the skills and the tiers as he/she wants (moving up and moving down). These changes are propagated to both tables, in other words, if the user reorders the tiers table, the skills table will reorder automatically to reflect those changes. Regarding preventing unwanted moves, our mechanism prevents moving up the first tier or moving up the first skills of a tier, as well for moving down the last tier or the last skill of a tier. The maximum reward box is now on the general items section (on top of the page).

Then, we defined new features to deal with the management of these items in this new configuration:

- Add Tier - by including an add button in the tiers section, we allow the user to add one tier to the



Skills

TIER	NAME	DEPENDENCIES	COLOR	XP				
1	Course Logo		■ #2773ed	150	✎	🗑	↑	↓
1	Movie Poster		■ #00c140	150	✎	🗑	↑	↓
1	Album Cover		■ #ff76bc	150	✎	🗑	↑	↓
1	Podcast		■ #4617b5	150	✎	🗑	↑	↓
1	Reporter		■ #c1004f	150	✎	🗑	↑	↓
1	Audiobook		■ #b11d01	150	✎	🗑	↑	↓
2	Looping GIF		■ #00a8a8	400	✎	🗑	↑	↓
2	Course Image		■ #006ac0	400	✎	🗑	↑	↓

Figure 4.16: New skills table in configuration page.

skill tree. By clicking it, a modal is shown, and the user can fill in all the information of the new element. This information will then be stored in the database.

- Edit Tier - we included an edit button in each row. By clicking it, a modal is shown, and the user can edit information regarding that tier, which will be updated in the database.

- Delete Tier - we have included a delete button in each row. If the user desires to delete a tier, he/she clicks on it. Then, a verification modal is shown asking if the user is sure about deleting it.

- Import Skills - by clicking on the import button, the user can choose a text file from his/her computer and import it. Then, the database is updated, as well as the list on the page.

- Export Skills - this button calls a function that will generate a text file with all the information of every skill (tier, name, dependencies, color, and XP) of the course and then, download it.

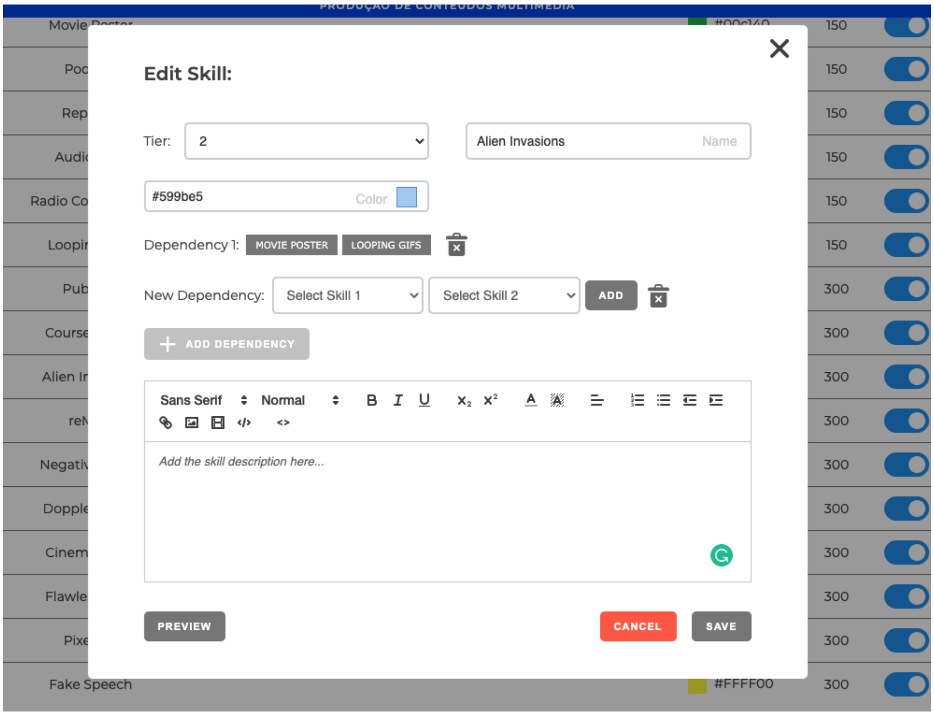
- Add Skill - the add button in the skills section allows the user to add one skill to the skill tree. By clicking it, a modal is shown, and the user can fill in all the information of the new element, including its dependencies and the content that should appear on its page. We will get in more detail below. This information will then be stored in the database.

- Edit Skill - we included an edit button in each row. By clicking it, a modal is shown, and the user can edit information regarding only that element, which will be updated in the database.

- Delete Skill - finally, we have included a delete button in each row. If the user desires to delete a skill, he/she clicks on it. Then, a verification modal is shown asking if the user is sure about deleting it.

We decided to not implement import/export features for the tiers since their configuration is very simple, with the user only having to fill in their name and the corresponding XP.

Moreover, each skill has its description which is an HTML page, and to write the description, we used a framework - Quilljs[1] - to help us building the text editor. We have look for alternatives, such as the RichTextEditor[2], but we ended up choosing Quilljs because it is cleaner, with a customizable toolbar, where we can choose which settings we want to include in it. The alternative had a default toolbar with too many elements that would end up not being useful to our system and even confuse the users. With Quilljs, the user can build a page with text and images, links, with many features such as using different fonts, bullet points and format the text. Then, this page will be saved as an HTML file in the server side, in the course folder. This editor is present on Add and Edit Skill modals (Figure 4.17). As we can see in this figure, the skills can have dependencies. Previously, skills had dependencies on skills from lower tiers. We now have a new way of dependency - wildcards. Wildcards are skills from a "special" tier, and if a skill has a dependency on a wildcard, it means that depends on that tier instead of on a specific skill. For the image picking, we used the picking file modal mentioned before.



Edit Skill:

Tier: 2

Alien Invasions

#599be5

Dependency 1: MOVIE POSTER LOOPING GIFS

New Dependency: Select Skill 1 Select Skill 2

Add the skill description here...

Figure 4.17: New modal to add/edit skills.

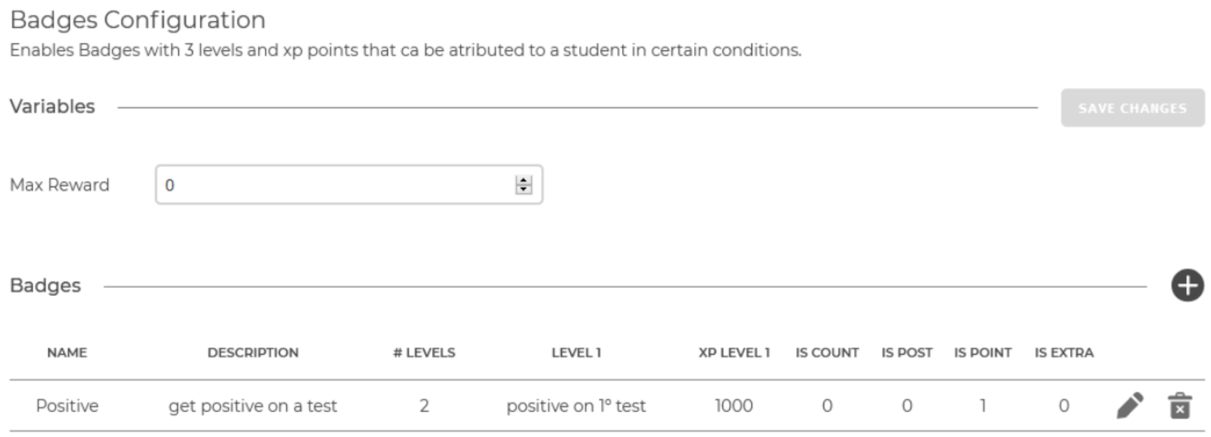
4.3.3 Badges

This module already had its configuration page using the Configuration Controller. However, this page only allowed the user to add one badge at the time and set the maximum reward value (Figure 4.18).

[1] https://quilljs.com/

[2] https://richtexteditor.com/

Consequently, the first thing we added was the import and export features, to allow users to import a file with all the badges and their information, such as name, XP, description, among others. However, the image for each base cannot be imported or exported. We have kept an add button if the user wants to add only one badge (Figure 4.19). By importing a text file, it is not possible to define the image(s) for each badge. Previously, these images were in the system, and they were global. To give more flexibility, we moved the badges folder to the courses folders, since courses may have different badges. Badges can have up to 3 levels, worth extra points and be bragging and worth nothing. All these features can be represented with a layer in the badge image. These layers were the same for all badges. Once again, to give GameCourse more flexibility, these layers can now be chosen in the badge configuration page. Alongside max reward input, we added 4 new fields to pick up an image (using our file explorer), one for each – extra overlay, bragging overlay, level 2 overlay, and level 3 overlay. These overlays will then be used for every badge of the course.



Badges Configuration

Enables Badges with 3 levels and xp points that ca be atributed to a student in certain conditions.

Variables

Max Reward 0

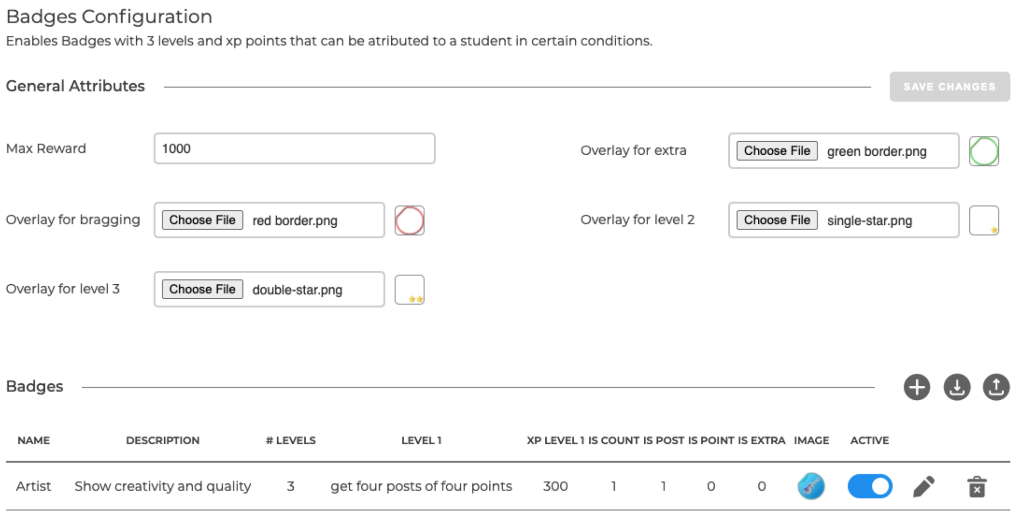
Badges

NAME	DESCRIPTION	# LEVELS	LEVEL 1	XP LEVEL 1	IS COUNT	IS POST	IS POINT	IS EXTRA
Positive	get positive on a test	2	positive on 1º test	1000	0	0	1	0

Figure 4.18: Previous badges configuration page.

To permit a user to pick up an image for each badge, we added a field in the new and edit badge modals to choose an image, which use our file explorer to upload or browse an image. After the user picks an image, we generate the other images of the badges according to its specifications (number of levels, is extra, is bragging). To do so, we used a library[3] that merge images. For each badge, if it is extra, then we merge the extra overlay for every level, and the same if it is bragging. Besides, for levels 2 and 3, if exist, we also merge the respective overlay. The resulting images appear next to the choosing field, as we can see in Figure 4.20.

[3] https://unpkg.com/merge-images



Badges Configuration

Enables Badges with 3 levels and xp points that can be atributed to a student in certain conditions.

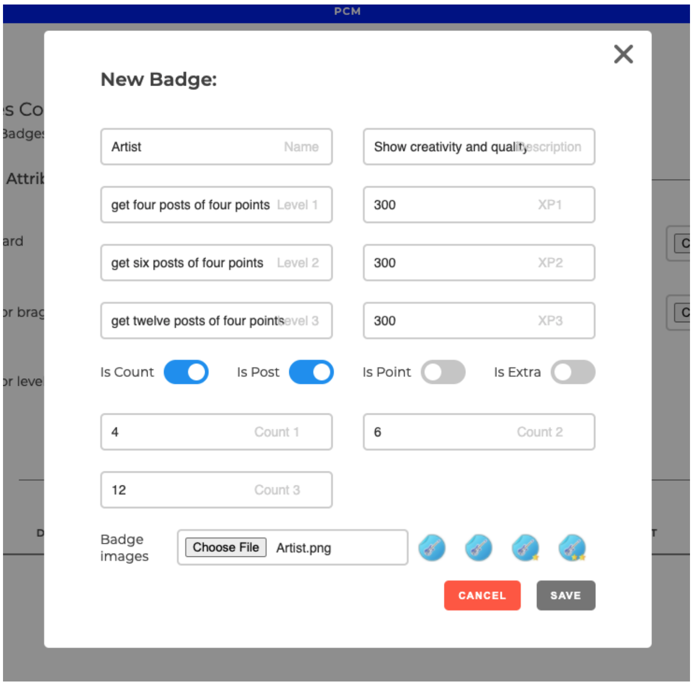
General Attributes

Max Reward	1000	Overlay for extra	Choose File green border.png
Overlay for bragging	Choose File red border.png	Overlay for level 2	Choose File single-star.png
Overlay for level 3	Choose File double-star.png		

Badges

NAME	DESCRIPTION	# LEVELS	LEVEL 1	XP LEVEL 1	IS COUNT	IS POST	IS POINT	IS EXTRA	IMAGE	ACTIVE
Artist	Show creativity and quality	3	get four posts of four points	300	1	1	0	0		●

Figure 4.19: New badges configuration page.



New Badge:

Name: Artist	Description: Show creativity and quality
Level 1: get four posts of four points	XP1: 300
Level 2: get six posts of four points	XP2: 300
Level 3: get twelve posts of four points	XP3: 300

Is Count ● Is Post ● Is Point ○ Is Extra ○

Count 1: 4	Count 2: 6
Count 3: 12	

Badge images: Choose File Artist.png

Figure 4.20: New modal for add/edit badges.

4.4 View Templates and Pages

In the first place, there was a huge misunderstanding regarding the difference between a page and a view template and the code reflected it. So, we started by focusing on understanding it, since our work would essentially lean over this part of the project, and, therefore, we must have a clear understanding of these concepts. We established the difference between a view template and a page: **View Templates** are a composition of views, which are defined in the chapter 3.1. As we explained there, when we have a view, we can choose to show it as a **page** with a certain name. Due to these adjustments, other ones were necessary, especially, on the modules. Most of them provide a default view template that is created when the respective module is enabled, and provides a default way to visualize their respective game element. In addition, previously, a page with the same content was also created and included directly in the navigation bar. The pages should not be connected to the modules, therefore, we no longer create any page as a module is activated. If a user wants a page with the content of the module's view template, he/she should create a page and choose that view template to show.

4.5 Views page

On the views page, many modifications were needed, mainly due to the modifications in the view templates / pages logic. As we have seen in chapter 3.1, the Figure 3.7 shows how this page looked before. Taking that into consideration, it did not make sense to access the view editor by clicking on a page card to edit it. So, we removed that, and we now only allow the user to access the view editor when he/she wants to edit a view template. Another adjustment we made was to have the role type related only to the view templates. Previously, both pages and view templates had their role type, but it only makes sense to keep it for view templates, since they are the ones editable in the view editor, where we add and edit views, as well as aspects. Furthermore, when we created a page, it would always appear on the navigation bar, and to remove it from there, we had to delete it. This was not a good approach, because we might want to deactivate it for a period. Hence, we added a new property to the pages - *isEnabled*, whose value is stored in the database. This new property allows us to manage the pages that should appear (or not) on the navigation bar. Besides, it was not possible to change any property of a page or a view template, such as their name. Consequently, we add the configuration icon to the cards. By clicking it, a modal is shown and here we can change the name, the view template that the page should show, and whether it is enabled or not, for the pages, and for the view templates, its name, and its role type. The new views page is presented in Figure 4.21. To access the view editor, we clicked on a page/view template card, and we moved to an intermediate page for aspect selection. This extra step was removed, and all the aspect management is now done in the view editor.

4.6 Navigation Bar

Even though we improved the functionality of the navigation bar by adding the *isEnabled* property to the pages, there was one feature that was not there. It did not consider the role of the user. For example, if we create a page whose view template only has content for role = Teacher, it should not appear on the

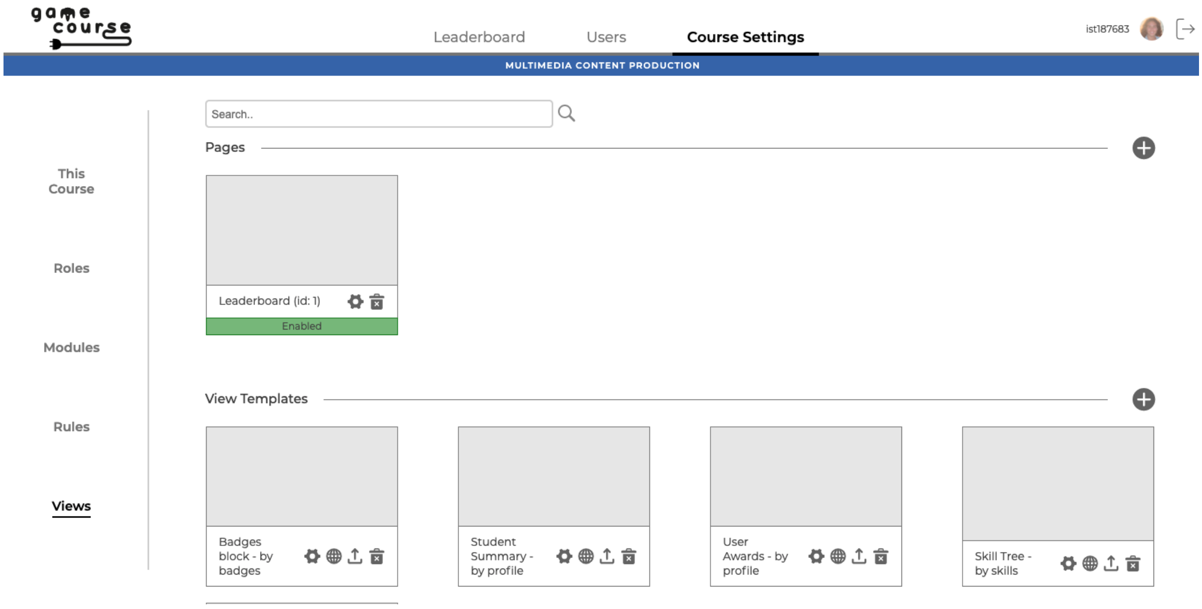


Figure 4.21: New views page.

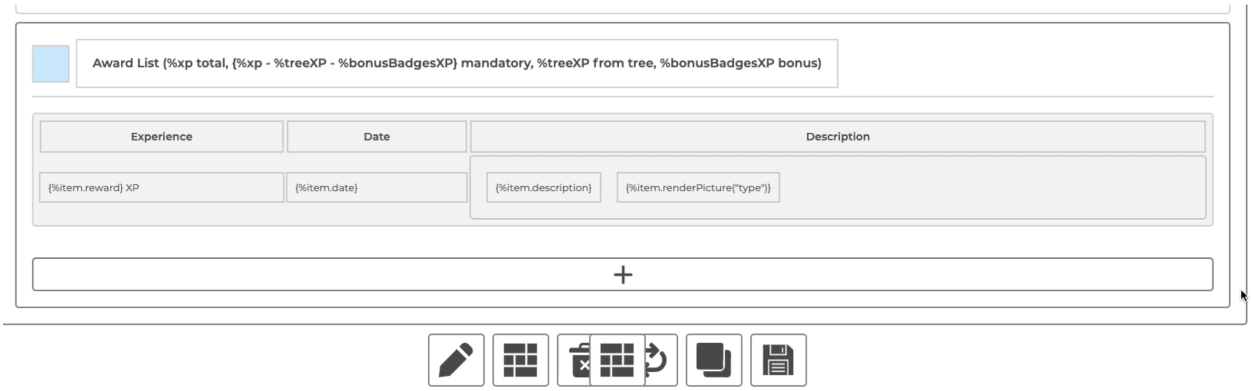
navigation bar when the user has role = Student. The navigation bar also needed one fix respected to the highlights. In other words, when we were on the users' page, for instance, and then we reloaded the page, the Users tab on the navigation bar did not appear highlighted as it should, since this was the page we were at. This also occurred for the custom pages. These two features are now fully implemented, preventing those situations.

4.7 View Editor

The view editor was the focus and the main goal of this project. This part was not fully implemented and the logic behind the code was not appropriate for the required functioning of the system. Therefore, many changes were needed, including a redefinition of the database schema.

4.7.1 Logical improvements

The view editor allows the user to build custom views by providing some built-in parts (text, image, blocks, tables, charts or even another view template that is already created). Before diving deep into its functionality, we fixed a bug that came from beforehand and brainstormed about the restructuring of the database and how to interact with the editor. The bug was regarding the toolbar of the tables – if we clicked on the table and then on an outside view, both respective toolbars would appear and therefore, they would end up overlapping, as shown in Figure 4.22.



Award List (%xp total, {%xp - %treeXP - %bonusBadgesXP} mandatory, %treeXP from tree, %bonusBadgesXP bonus)

Experience	Date	Description
{%item.reward} XP	{%item.date}	{%item.description} {%item.renderPicture("type")}

Figure 4.22: Overlapping of toolbars in the view editor.

Furthermore, our brainstorming led us to change the logic behind the aspects and the way we deal with them. First, aspects are variations of a view for different roles. For example, a view can be showed differently according to the person who is seeing it.

Before, we could only choose the aspect at a top level, and this aspect would apply to the whole view template. Now, we can choose the aspect for every view and at every level.

So, previously, the views only had an incremental *id* and an *aspectClass* (Foreign Key for the table `aspect_class`). The value of the *aspectClass* field was common to all the views of every aspect of the view template, or, if the view template only had one aspect, it was set to null. Then, we have the table `view_template` that saves the view template id and the correspondent view id. The aspects of a view template were found by searching the views whose *aspectClass* value was equal to the *aspectClass* of the view whose id was on the `view_template` table. Also, the first view of each aspect was found with the condition **part == block && parent == null**, that is, each view template had to have, at its root, a Block view element. This condition was changed and now view templates can be any possible view element, from the simple Text element to more complicated ones such as Block or Table. This allows us to create view templates that are meant to be used as parts of a larger view, and not as top-level view templates (eventually used as pages) themselves. Thus, we had to manage this logic, to find all the aspects of a view regardless their type.

As we have seen, each view can have several aspects. Inside the view editor, and while building the view templates, we use only views and their aspects. So, we had to find a way to save these correspondences between the views and the respective aspects, which was important to further build the view to present to each role.

Therefore, the main changes regarding the logic to deal with the aspects were:

- Removing the `aspect_class` table since it was unnecessary in the new model as well as the *aspectClass* column of the view table.

- Adding the column *viewId* to the view table in the database, to identify all the aspects of a view. All the aspects of a view have the same *viewId* so we can easily identify them. This implied a change in the page table. The column *viewId* used to refers an id of a view. Now it refers to a *viewId*.
- Saving each view / aspect according to its role. As mentioned above, before we had one entire view per aspect, it was not possible to create an aspect for a view inside a view template. Now, every view can have aspects.
- Adjusting the flexibility for the aspects: now each view template can have any part as main view (block, text, image, etc.).

We could have maintained the `aspect_class` table and not adding the *viewId* column, but, in this case, we could not find the aspects of a specific view, which is now possible with the *viewId*. Regarding the `aspect_class` table, we found it useless in our new logic, and its usability was not clear even previously.

As stated before, we can add a view template to another view, and this can be done as a copy or by reference, and, in this case, it means that the views will have different parents in different contexts. This enforced us to change the logic of the parent/child relation. Beforehand, a view only had one parent and it was set on the view table. We removed that column and created a new table – `view_parent`, which has as columns:

- *parentId*: id of the parent view;
- *childId*: *viewId* of the child view;
- *viewIndex*: indicates the order in that relation.

The *parentId* is the id of the view since a view only has one parent. However, the *childId* is the *viewId* since a view can have many aspects of a view as children.

4.7.2 UI improvements

Since we removed the page where the user would choose which aspect he/she wanted to create or modify, we needed to add this feature to the view editor. We decided to use dropdowns to choose the different aspects. The options are the aspects for which there is any content. By changing these dropdowns, the view contents change accordingly. When the role of the view template is `ROLE_SINGLE`, only the viewer dropdown appears, whereas when the role is `ROLE_INTERACTION`, appears the user and viewer roles, as we can see in the Figure 4.23. By having it in the view editor, it allows the user to manage the aspects of a view template without leaving the view editor. Besides, it makes possible to the user see which aspect he/she is seeing.

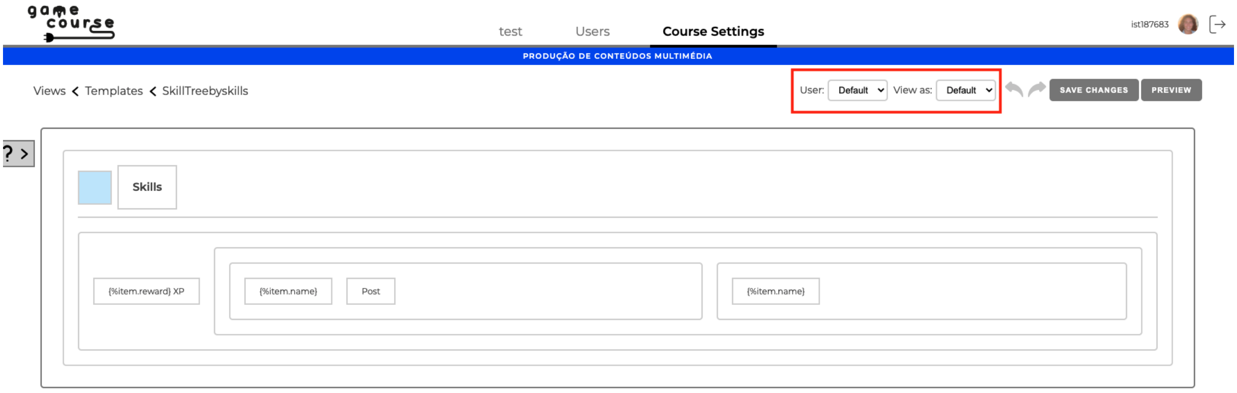


Figure 4.23: Aspects dropdowns for user and viewer roles.

To create new aspects for a view, we decided to add a new tool to the toolbar of each part – the Manage Aspects tool. At the end of the toolbar, we also added a new label that shows which role will see the selected view (Figure 4.24).

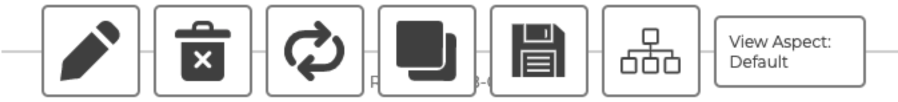
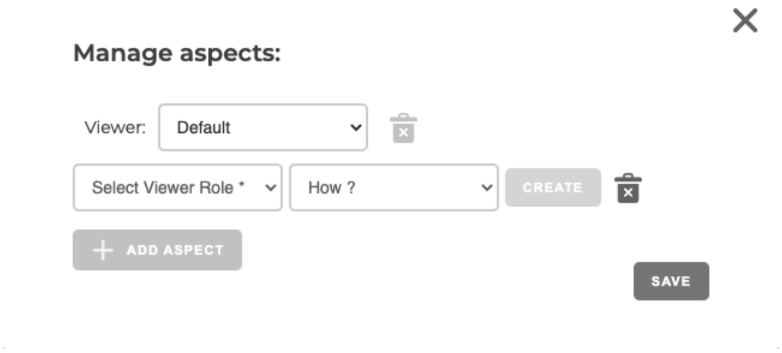


Figure 4.24: New edit toolbar.

We created a new modal that is shown after clicking the Manage Aspects tool (Figure 4.25). And here we can:

- Change the aspect we want to see for that specific view (without changing the global role, which is the one selected on the dropdown on top). If the role is the same as the global, the view will appear with a solid border. However, if the role we are seeing does not correspond to the global one, that view will appear with a dashed border.
- Add a new aspect for this view. By clicking the "Add Aspect" button, a form appears above to select for which role we want to create an aspect and how. The latter has 2 options:
  - Create from scratch
  - Create with the selected aspect as a basis

The preview section also needed some improvements since we could not see exactly how the page will look like. The preview section showed the views inside the boxes as it happens in the view editor while editing, and it did not position the elements as they will be in the page. Therefore, we improved this section to give us the view as it would appear when see in a page.



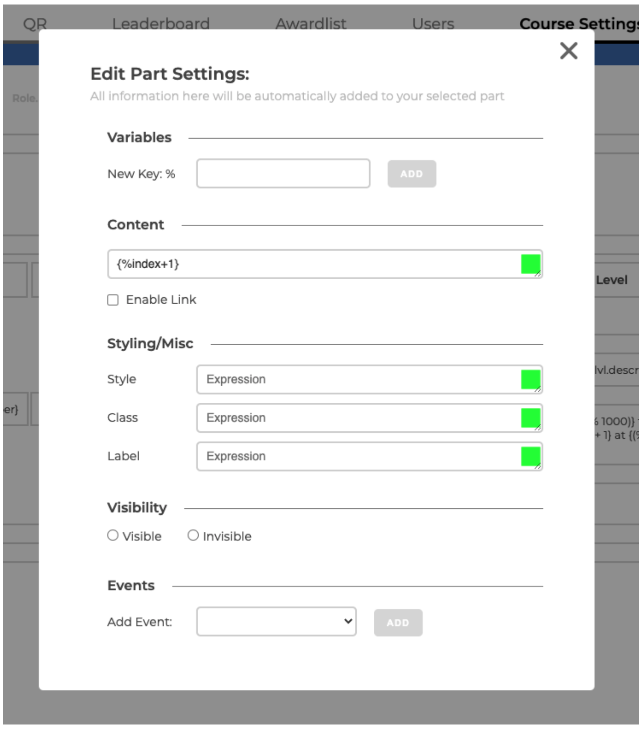
Manage aspects:

Viewer: Default

Figure 4.25: Manage Aspects Modal.

4.7.3 Edit Part Modal

While building a view resorting to the view editor, there are properties of the parts that we want to change or add. For this, in the toolbar, we have the edit part button. If the user clicks it, the edit part modal opens (Figure 4.26). In this modal, the user can define the function for the loop data, set new variables and their value, set events, change the visibility, add styling, among others specific for each part. Some of these properties must be defined using the EL. In those, there was an algorithm that gives suggestions, provides auto complete and a green light sign regarding what the user write and whether it was rightly written or not. However, only if the users start to write, this auto complete appeared.



Edit Part Settings:

All information here will be automatically added to your selected part

Variables

New Key: %

Content

{%index+1} 🟩

☐ Enable Link

Styling/Misc

Style: Expression 🟩

Class: Expression 🟩

Label: Expression 🟩

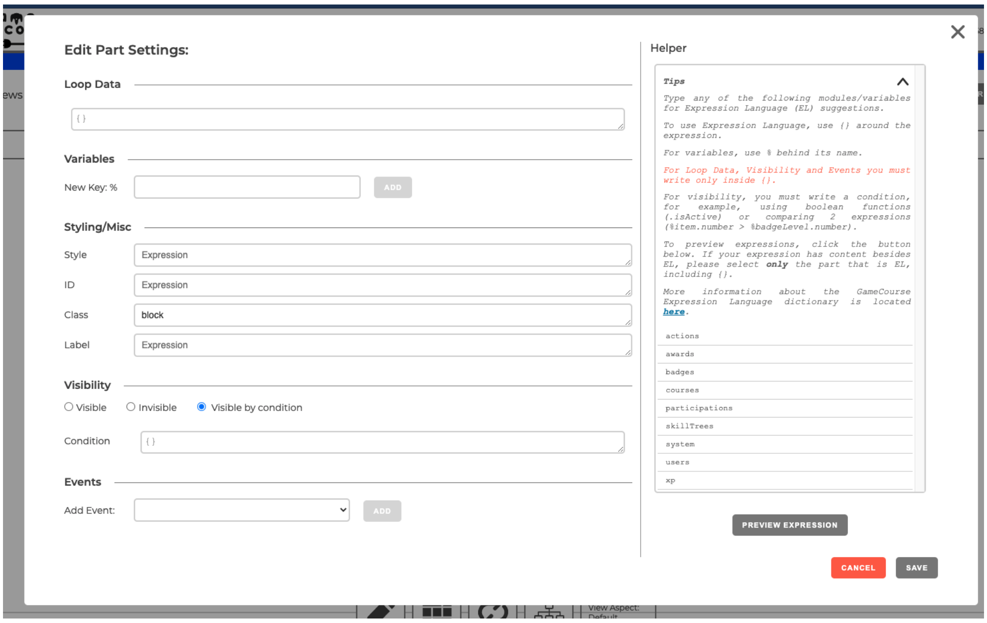
Visibility

○ Visible ○ Invisible

Events

Add Event:

Figure 4.26: Previous Edit Part Modal.



Edit Part Settings:

Loop Data

{ }

Variables

New Key: %

Styling/Misc

Style: Expression

ID: Expression

Class: block

Label: Expression

Visibility

○ Visible ○ Invisible ● Visible by condition

Condition: { }

Events

Add Event:

Helper

Tips

Type any of the following modules/variables for Expression Language (EL) suggestions.

To use Expression Language, use {} around the expression.

For variables, use % behind its name.

For Loop Data, Visibility and Events you must write only inside {}.

For visibility, you must write a condition, for example, using boolean functions (.isActive) or comparing 2 expressions (%item.number > %badgeLevel.number).

*To preview expressions, click the button below. If your expression has content besides EL, please select **only** the part that is EL, including {}.*

*More information about the GameCourse Expression Language dictionary is located **here**.*

Figure 4.27: New Edit Part Modal.

Although there is a button that takes the user to the documentation page, in which there are some guidelines on how to write using the EL, the live support was not helpful. To tackle this problem, we expanded the modal (Figure 4.27), and used the CodeMirror[4] framework. This framework was already used in the rules editor, so to keep it consistent, we decided to use it too in this section. For every input box that is filled resorting to the Expression Language, we replaced them with a CodeMirror box. This framework provides a suggestion box and an auto complete functionality, according to what is written (Figure 4.28). These suggestions are given by the system. We check what is written and what is accepted and manage them accordingly. For example, in the loop data, the expression should return a collection, so the user should write (inside the curly braces) is whether a variable or a module followed by function(s). The function that deals with the suggestions, detects them using Regular Expressions. For variables, it detects with the % symbol. For functions, it detects if it has a module name followed by a dot or module name followed by a dot, a function, and a dot again. Variables can have properties, which are detected with the % symbol followed by the variable name followed by a dot.

Besides, on the right side, we placed a section for help. On the top of this section, we have a box, in which we have:

- **Tips section**: here are some tips to guide the user before/when he/she is writing. This list can be collapsed using the arrow on the top right.
- **Suggestions section**: this section is dynamic, and it changes according to what is written. This section will only have content is the user clicks one of the input boxes. If a function of a module, or

[4] https://codemirror.net/index.html

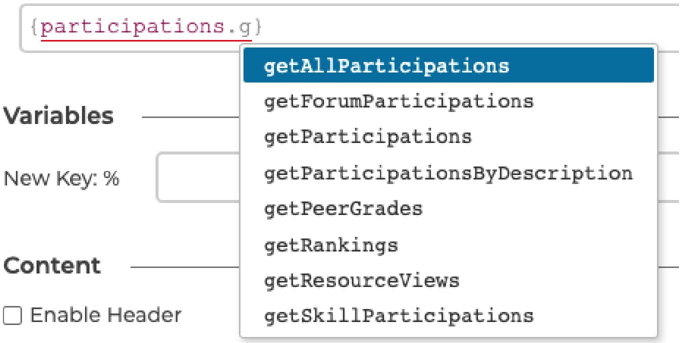


Figure 4.28: Suggestions box provided while the user types.

a variable, is selected, its properties, such as arguments (for functions), return type and description appear in this section (Figure 4.30). Otherwise, it will appear the possible functions (Figure 4.29).

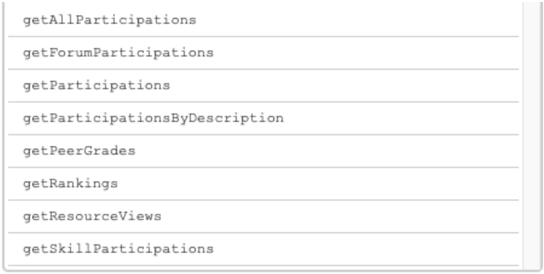
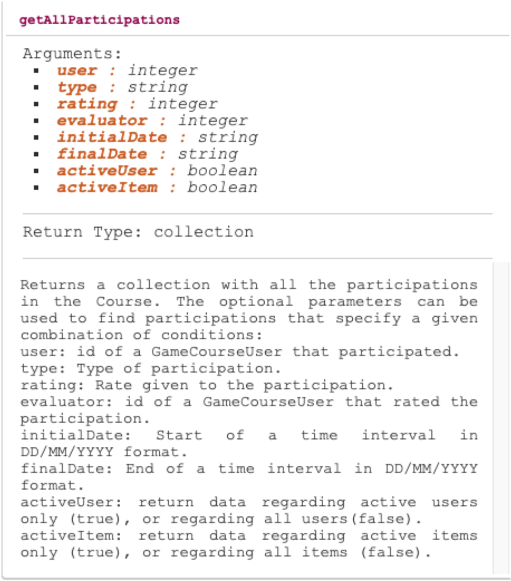


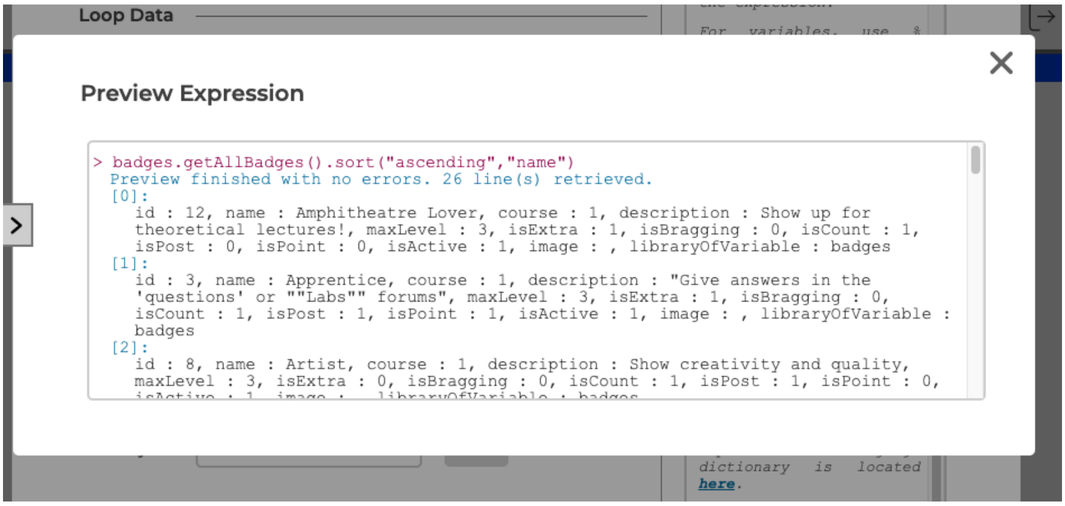
Figure 4.29: Suggestion section while the user types.



```
getAllParticipations

Arguments:
 ▪ user : integer
 ▪ type : string
 ▪ rating : integer
 ▪ evaluator : integer
 ▪ initialDate : string
 ▪ finalDate : string
 ▪ activeUser : boolean
 ▪ activeItem : boolean

Return Type: collection

Returns a collection with all the participations
in the Course. The optional parameters can be
used to find participations that specify a given
combination of conditions:
user: id of a GameCourseUser that participated.
type: Type of participation.
rating: Rate given to the participation.
evaluator: id of a GameCourseUser that rated the
participation.
initialDate: Start of a time interval in
DD/MM/YYYY format.
finalDate: End of a time interval in DD/MM/YYYY
format.
activeUser: return data regarding active users
only (true), or regarding all users(false).
activeItem: return data regarding active items
only (true), or regarding all items (false).
```

Figure 4.30: Suggestion section when a function is selected.

On the bottom, we have a preview button. By clicking it, it will open a modal (Figure 4.31). If no box is selected, it will show the message *"Select one of the to see its preview!"*. While the expression can be previewed – expressions with variables cannot be previewed since their value has to be calculated and that need a complete evaluation of the whole view – the modal shows whether the result of that expression if correctly written or shows the message *"Something is wrong with your expression... :(Try again!"*. If it cannot be previewed, it shows the message *"Expressions with variables cannot be tested, you have to preview the whole view"*.

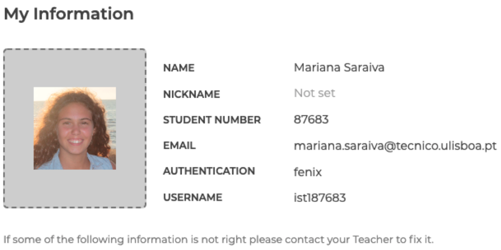


```
> badges.getAllBadges().sort("ascending","name")
 Preview finished with no errors. 26 line(s) retrieved.
 [0]:
   id : 12, name : Amphitheatre Lover, course : 1, description : Show up for
   theoretical lectures!, maxLevel : 3, isExtra : 1, isBragging : 0, isCount : 1,
   isPost : 0, isPoint : 0, isActive : 1, image : , libraryOfVariable : badges
 [1]:
   id : 3, name : Apprentice, course : 1, description : "Give answers in the
   'questions' or ""Labs"" forums", maxLevel : 3, isExtra : 1, isBragging : 0,
   isCount : 1, isPost : 1, isPoint : 1, isActive : 1, image : , libraryOfVariable :
   badges
 [2]:
   id : 8, name : Artist, course : 1, description : Show creativity and quality,
   maxLevel : 3, isExtra : 0, isBragging : 0, isCount : 1, isPost : 1, isPoint : 0,
   isActive : 1, image : , libraryOfVariable : badges
```

Figure 4.31: Preview Expression Modal.

4.8 My Information Page

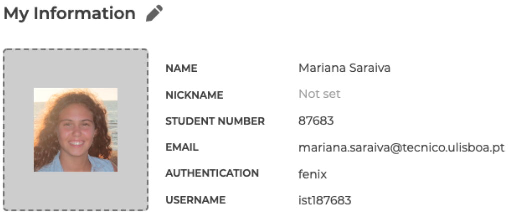
My Information page was static (Figure 4.32) - it was only possible to check their information, and users that would like to change some information, and did not have admin permissions to do it on users' page, had to ask an admin user to do it. To prevent students (non-admin) from having to request such changes from professors (admins), we added the possibility of changing some of their information in this page (Figure 4.33). We added an edit icon next to the title "My Information" and by clicking it, a modal will show up (Figure 4.34). This modal has some editable fields – the ones that are possible to be changed. Some are not editable since are not supposed to be changed by non-admin users, for example, the student number and the authentication service.



NAME	Mariana Saraiva
NICKNAME	Not set
STUDENT NUMBER	87683
EMAIL	mariana.saraiva@tecnico.ulisboa.pt
AUTHENTICATION	fenix
USERNAME	ist187683

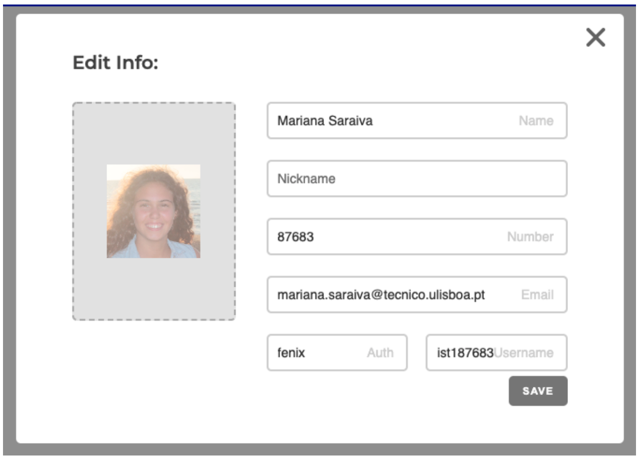
If some of the following information is not right please contact your Teacher to fix it.

Figure 4.32: Previous My Information Page.



NAME	Mariana Saraiva
NICKNAME	Not set
STUDENT NUMBER	87683
EMAIL	mariana.saraiva@tecnico.ulisboa.pt
AUTHENTICATION	fenix
USERNAME	ist187683

Figure 4.33: New My Information Page.

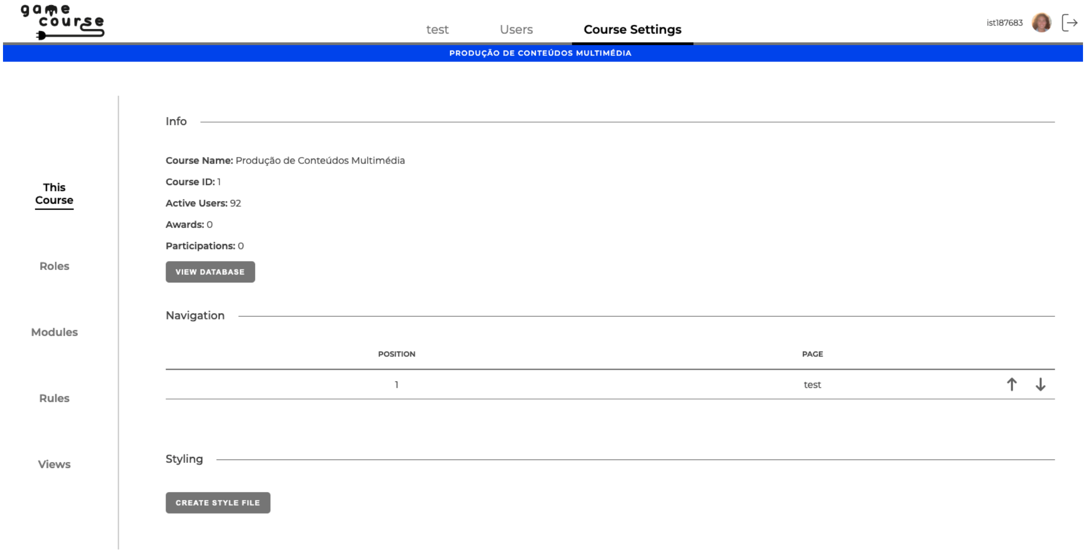

Edit Info:

Name	Mariana Saraiva
Nickname	
Number	87683
Email	mariana.saraiva@tecnico.ulisboa.pt
Auth	fenix
Username	ist187683

Figure 4.34: Edit My Info Modal.

4.9 Course Settings page

There were two tasks that was related to the course settings – the change of order of pages on navigation bar and the override of the default style with a personalized CSS file. We decided to use the global course settings to place these two new features (Figure 4.35). First, we added a section to change the order of the pages on the navigation bar - Navigation. This section contains a table with all the pages but users and course settings. We decided to not permit changing their order since they are only visible for admin, and for that reason, they are always the last two tabs on the navigation bar. To manage the order of each page, we added a column to the page table, *seqId*, which indicates the order in which each page appear in the navigation bar.



Info

Course Name: Produção de Conteúdos Multimédia
Course ID: 1
Active Users: 92
Awards: 0
Participations: 0

Navigation

POSITION	PAGE
1	test

Styling

Figure 4.35: New Course Settings Page.

Regarding the theme feature, we created another section – Styling (Figure 4.36). If the course has not a file yet, a button “Create style file” appears. Once the user clicks it, a CSS file is created and saved in the course folder. The styling in the file will only apply to the respective course and it will override the styles of every page of that course. When the course already has a style file, this can be modified through the User Interface, in an editor generated with CodeMirror. With this feature, GameCourse leverages the power of the styling, since it allows the customization of every view, even every aspect, which will impact the way users see each element, the system in general, and mainly how they will accept gamification. By setting different styles for different views, we are facilitating the views customization. The system has a clear separation between the classes in view rendering, in which this file has impact, and the backend classes, which are defined internally as the style of the whole system and it is not overridden by the classes defined in the file, it can only be changed by the developers. Regarding the use cases, we have two main examples:

- Defining new properties, such as a new border color, for a specific view. For this, the user must go to the view editor and edit the view, adding a class or ID as its property. Then, in this section, write the CSS for that new class or ID.
- Redefining the style of views of the same type, for example, tables. For example, if the user wants every table of the course to have bold text in the header, they may override the styling of the table element.

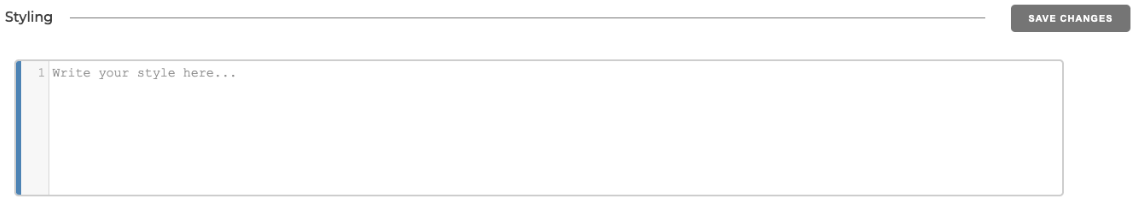


Figure 4.36: Styling section.

5

Evaluation

Contents

In this section, we will present the evaluation performed over the new features of the GameCourse to analyse whether their implementation was helpful to the system. We first describe the procedure of the user tests, explain the tasks, and characterize the users that participated. Then, we provide a deep analysis of our results and discuss what we observed.

5.1 User Tests

To evaluate if we have been successful in improving the GameCourse, we performed user tests on the developed work. We gathered a group of 20 potential users, who were asked to perform **13 tasks** of different contexts on the new system. The users performed the tasks as an admin user. To evaluate the performance of a user in each task, we collected the following data:

- Whether the user completed the task
- The time spent on each task
- The number of errors

Because we are in a pandemic situation, some user tests were performed remotely on the Zoom[1] platform. We chose Zoom not only because it allows voice and video calls, screen-sharing, recording and remote-control features, but also because the users who needed to perform the tests remotely already had it installed.

5.1.1 Participants

We conducted the user tests with a total of 20 users, the minimum number for a meaningful analysis. Our users' ages range from 17 to 54 years old, and the median age is 24 years old. Three of them did not have any programming experience, while one had as a hobby and the remaining had programming experience, either academically or professionally. Regarding their contact with the GameCourse system, three users had previously contacted with it as students, but with the previous UI version, and two as seen the admin version although neither have worked with the views in this new version.

5.1.2 Procedure

We started each test by sharing a questionnaire with the users in order to obtain some information, such as their age, whether they have programming experience or not, and if they have any experience with the GameCourse system. For the users who had to perform the tests remotely, we added a question

[1] https://zoom.us/

to request permission to record the Zoom meeting for further analysis of how the participant interacted with the system.

The next step was presenting some context about the project, specially to the users who had no previous contact with the MCP course. This included a brief explanation of what gamification is and how it is being used in the MCP course, as well as some details about the GameCourse system, such as the EL and the views. To the users who had contacted with course, we only introduced the new features of the system.

Then, we allowed every user to explore the system, the modules page and the view editor for at most five minutes, so they could better understand how the interface is organized before trying to complete the tasks. We then gave them the list of tasks to be performed:

1. On the PCM course create a new Page.
2. Configure the module ”XP and levels” by adding a new level.
3. Configure the module skills by adding a new skill.
4. Configure the module badges by setting the overlay for extra as the `red_border`.png image by uploading it from the computer.
5. Go to your info and change your nickname to “Mariana”.
6. Now we want that the PCM course looks differently. We want to change the look of some elements. For that, define a new style for this course.
7. On the PCM course create a new view template.
8. Add a text part to the outside block.
9. Create a new aspect, from scratch, for role Teacher for the outside block.
10. Create a new aspect, from scratch, for the text box for which the content is “PCM” for role Student.
11. I want the textbox for which the content is “IPM” to have another content: Student number of the user with the id=1.
12. Add the class `blue_border` to the outside block. (use “; `blue_border`;”).
13. I want this view to go through all the active courses. How would you perform this?

To elaborate this list of tasks, we intended to test specific elements and features:

– **Tasks 1 and 7**: these tasks test the basic interactions regarding the pages and view templates and, asking the user to create one of each, respectively.

– **Tasks 2, 3 and 4**: we came up with these tasks to test the new modules configuration for the “XP and Levels”, Skills and Badges modules, by testing some new features: adding a new level, adding a new skill and setting the overlay for extra for the Badges.

– **Tasks 5, 6 and 12**: These tasks were testing the general improvements regarding new functionality in the My Information page, the ability to add personalized styling to a course and apply it to some elements (in the view editor).

– **Tasks 8, 9 and 10**: with these tasks we could assess the view editor, with a simple task of adding a new part (task 8) and two tasks to test the aspects functionality.

– **Tasks 11 and 13**: we chose these tasks since they would test the EL and if the system has enough help for a user without any programming experience would understand how it works.

Before doing the tests, we did pilot tests to better formulate the tasks that could be ambiguous, and modified them to make them more clear. However, the task 13 remained hard to understand for the users with no experience in programming. Besides, the tasks 6 and 12 which mention the term “class” aroused doubts and confusion among users with no previous knowledge of CSS, for example. These tasks were given randomly so there is no direct influence of the learning curve of the system. We decided to set a time limit for each task as a safeguard - 3min 30sec for the hard tasks (tasks 11 and 13) and 1min 30sec for the easier ones (the remaining tasks). The Appendix B presents the detailed version of the tasks.

In addition, to gather suggestions of improvements and things that are not clear to improve later, we asked 3 users to explore the system and simultaneously think-aloud.

5.2 Result Analysis

We will now analyze the results of the tests. We first show some of the general data for each task and then organize the tasks according to the features they were testing as well as explain our observations.

Table 5.1 shows the results for task completion. In general, users did not have difficulty performing tasks 1, 2, 3, 4, 7, 8, 9 and 10 which all present a success rate above 95%. On the other end, tasks 11 and 13 proved to be the most difficult to our users, with success rates below 30%. These tasks have the lowest success rate since they required writing an expression using the EL after little interaction with the system. Besides, to write an expression, most users needed to research about the language in the documentation in order to learn about it and, even so, we noticed that users with no programming experience had much more difficulty performing these tasks. The remaining tasks were all within the 30% − 90% range and the obstacles that we observed while users were attempting to solve them gave us good insights on current problems with the UI.

Task	1	2	3	4	5	6	7	8	9	10	11	12	13
Success Rate	100%	100%	100%	100%	85%	90%	100%	100%	100%	95%	30%	90%	30%

Table 5.1: Success Rate of each task.

Afterwards, we calculated the mean, median and standard deviation of the measures we took: time of completion and numbers of errors, which can be seen in Table 5.2. Regarding the tasks 11 and 13, we did not measure the number of errors since the tasks ask to write an expression, which may not be performed at the first try and the user can try as long as they are researching and learning the EL. As is shown in table 5.2, tasks 5, 8, 10 and 12 had more errors while users perform them. Regarding task 5, which asks to change the nickname of the user's info, we realized that the amount of errors was mainly because this info can be changed in two places when the user is admin, which led to a success rate below 100% on this easy task. Regarding tasks 8 and 10, these tasks were in the view editor, and the errors were, on average, because users were discovering its functioning and what each icon on the toolbar does. Finally, the task 12 asked to add a class to a view, which revealed to be confusing to three users, two of them did not have CSS knowledge. In section 5.2.1, we look into all the tasks in more detail resorting to other observations we made while testing.

	Task	1	2	3	4	5	6	7	8	9	10	11	12	13
Time	Mean	00:33	00:29	00:59	00:53	00:38	00:54	00:23	00:38	00:38	00:51	03:02	01:09	03:10
	Median	00:29	00:29	00:59	00:48	00:21	00:53	00:20	00:30	00:37	00:50	03:30	01:00	03:30
	Std. Dev	00:13	00:10	00:18	00:15	00:30	00:22	00:11	00:23	00:25	00:26	00:49	00:48	00:47
Errors	Mean	0.3	0	0.3	0.15	0.6	0.6	0	0.85	0.6	0.6	-	0.7	-
	Median	0	0	0	0	1	0	0	1	0	1	-	1	-
	Std. Dev	0.48	0	0.48	0.38	0.65	0.77	0	0.55	0.77	0.65	-	0.48	-

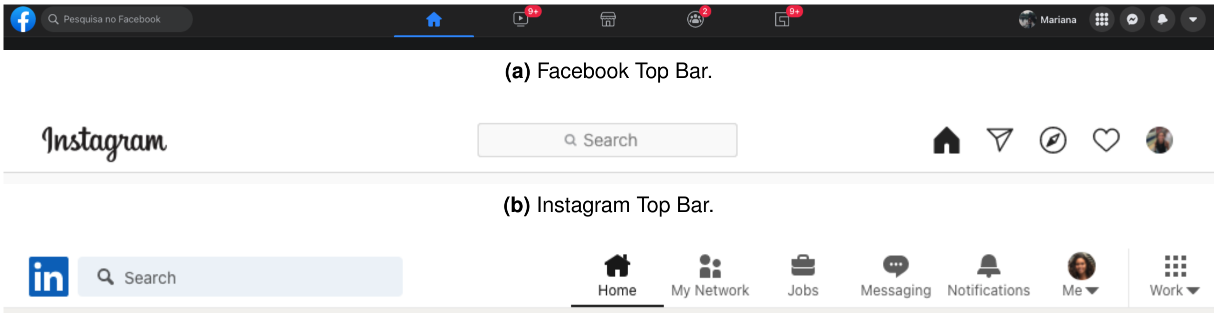
Table 5.2: Mean, Median and Standard Deviation of the measurements taken from each task.

5.2.1 Task Specific Analysis

The 13 tasks we asked to be performed had different levels of difficulty and different character. We asked users to rate each task regarding their perception of difficulty in a scale between **1 (Very Easy)** and **7 (Very Difficult)** after they had completed it. We will now look into these tasks and analyzing the most frequent mistakes they made, and taking this into account along with their suggestions to draw some conclusions on what can be improved.

We start by analysing the tasks related to the basic interactions with the views page, **tasks 1 and 7**, which revealed to be the easiest tasks with an average rating below 2.0 in the difficulty scale. These tasks had a 100% of success rate and only the **tasks 1** had some insignificant errors, mainly, because some users did not remember where pages are and ended up looking for the pages in other sections.

Then, **tasks 2, 3, and 4**, which are related to the modules configuration, also had a average rating in the difficulty scale between 1.0 and 2.5. We noticed that some users were confused with the names



(a) Facebook Top Bar.

(b) Instagram Top Bar.

(c) LinkedIn Top Bar.

Figure 5.1: Example of where user icons are located in systems with a profile.

of the modules and templates. **Task 2** was considered the easiest task and every user performed it without any trouble. **Task 3** required to add a new skill in the module Skills. The system also had the template Skills in the views page. Therefore, one user was confused and thought the task should be performed in the template, but as they re-read the task, they realized they were in the wrong place, and went to the modules page and completed the task successfully. **Task 4** asked to configure the module Badges, and that confusion also happened to another user, but they ended up completing the task. We noticed a great increase in the interaction with modules configuration. The users that made that mistake performed the tasks in the very beginning. Then, when performing other configurations in the modules, they were already aware on where they should go.

Task 5 asks users to change the nickname of the logged in user and had an average perceived difficulty below 2.0. This task was specially interesting since three users did not see the icon of the user on the top right corner of the page. This issue led users to either not complete the task or use the Users page to do it, which was not the goal, since this page will not be able for non-admin users. On the other hand, most users that performed successfully this task, by going to the My Information Page, did it in a short time, and immediately saw the icon. This can be because of its location, which matches the location in many websites with profiles such as social networks, for example, Facebook, Instagram and LinkedIn (Figure 5.1).

We analyse **tasks 6 and 12** simultaneously, since both may require extra knowledge of CSS and, specially, the term “class” in this context in order to understand what is asked. Nevertheless, one of the users without this knowledge completed both. They were both rated with an average difficulty between 2.5 and 3.5. **Task 12**, which asked to add a class to a view, was affected essentially by the misunderstanding of the real meaning of an element having a class and how they could change it. Users without any programming experience found it extra hard to perform however 2 of the 3 of this group could complete it since they could find the class label in the edit part modal. Besides, a user with programming experience, could not understand where they should do this task. Regarding **task 6**, which asked to add new styling to a course, was not completed only by 2 users with no programming experience.

We then analyse the **tasks 8, 9 and 10** that were concerned about the tasks within the view editor. **Task 8** asked the users to add a new part to the outside block. This task has a perceived difficulty of around 2.5, mainly because users struggled with the icons in the edit toolbar and what they could do in which one. Regarding **tasks 9 and 10**, where users must create new aspects, the main struggle was, once again, find the right icon - the aspects icon. These last two tasks had an average rating between 2.0 and 3.0. In all these 3 tasks, users tended to first click on the edit icon to open the edit part modal. Then, they realized that the tasks could not be performed in that modal and tried to explore more the view editor until reach the correct icon. The main struggle of the users was regarding the aspects icon and the icon for edit layout that were not understandable or intuitive. However, after the first task of creating an aspect, users performed the second one easily. Also, a user suggested having a label for each icon that is visible on hover. Although we already have labels (Figure 5.2), they do not appear immediately when hovering and we need to wait a few seconds until they appear.

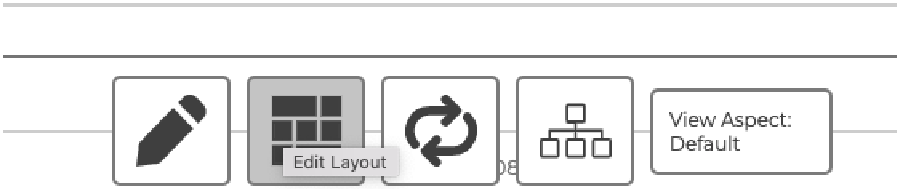
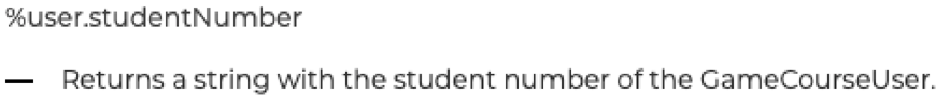


Figure 5.2: Edit toolbar when hovering one of the icons.

Finally, we analyse **tasks 11 and 13**, the more difficult tasks, with an average difficulty rated above 5, which asked the user to write expressions using EL, which takes time to understand how it works. All users that completed at least one of these tasks already had programming experience and it may be more logical after reading part of the documentation. Even users with programming experience, some only could complete the task they performed in the second place, after failing the first one. Most users consulted the documentation page to try to write the desired expression. But some did not understand it. In **task 11**, we expected the user to write {**users.getUser(1).studentNumber**}. In the documentation, we can see that the properties of the object user are described as *%user.studentNumber* for the student number, but for users with no programming experience, it was not clear that *%user* was an user object (Figure 5.3).



%user.studentNumber

— Returns a string with the student number of the GameCourseUser.

Figure 5.3: Documentation for property studentNumber of a GameCourseUser object.

Task 13 asked the user to get all the active courses. To write this expression, almost every user started by using the variable *%course* and the property *isActive*, that, by chance, we give as an example for a visibility condition in the tips, and users thought that they were supposed to use this property because of the task asking for active course.

Overall, when it comes to write an expression using EL, we noticed some struggle, mainly regarding:

- **Variables**: users did not understand what they represent or when they should use them.
- **Loop Data**: for three users, it was hard to get what it means.
- **Libraries and Functions**: most of the times, users were not aware that they must first select the library and only then choose the function name. The performance of the users without any programming knowledge was even more affected by the fact that they did not know how a function works, namely that its invocation includes an argument list enclosed by parenthesis.
- **Suggestions section**: tips are occupying most part of the modal and users end up not seeing the suggestions.

5.2.2 System Usability Scale (SUS)

The data collected from the user questionnaire allows us to calculate the System Usability Score (SUS) score [Brooke, 1995], since it can help understanding the users perceptive of the usability of a system. We processed the data by following these steps:

- For each of the odd numbered questions, subtracted 1 from the score.
- For each of the even numbered questions, subtracted their value from 5.
- Added the previous results for the total score. Then multiplied it by 2.5.

From the table 5.3, we can see that the mean of the SUS scores is 71.7 which is above 68, the average score, which means that our system has good usability.

	Mean	Median	Std. Dev.
SUS Scores	71,7	77,5	12,89

Table 5.3: Analysis of the obtained SUS scores.

5.2.3 NASA-TLX

After each task, we asked the users to complete an overall NASA Task Load Index (NASA-TLX) analysis, which measure and conduct a subjective mental workload assessment of the work performed in Human-Machine Interface systems (Table 5.4). For this analysis, we used a variant of this tool - Raw NASA-TLX - which sums the results of each dimension of the test, rather than using weighted scales.

The results show a low average value which means that the system has a small impact on the workload of the user.

	Mean	Median	Std. Dev.
NASA-TLX	33,5	32,5	7,925

Table 5.4: Analysis of the obtained NASA-TLX scores.

5.3 Discussion

The users tests helped us understanding the most likely sources of confusion when using the system. The new pages for configuring the modules were, in general, well understood by all the users. Similarly, the creation of new pages and view templates were also comprehensible.

In addition to the difficulties we could observe during testing, we also asked users for direct suggestions.

Regarding the Edit Part modal, users proposed many changes:

- Summarizing tips, to visualize better the suggestions part, because when we only have one suggestion, this does not stand out.
- Writing suggestions in other color, or bigger font size.
- Making suggestions clickable.
- Having an info icon next to each text box (where EL may be used) to open the tips.
- Auto complete: it should appear before the user starts to write, when they click on the box.

Regarding the overall system, users proposed:

- Having a tutorial on the view editor and for the most common tasks.
- Having more concrete examples of expressions written in EL.
- Using more color in buttons to intuitively show the next interaction.

We consider these results acceptable, since some tasks had limited time to perform. Writing expressions using a new language with high expressiveness as the EL needs some adaption and time to learn. However, we believe the view editor would leverage from a tutorial for new users. In addition, to tackle the icons understanding issue, there are two possible solutions: change for more understandable icons or add labels that appear immediately when hovering. In the edit part modal, the tips and suggestions sections should be reviewed to stand out the suggestions, because most people did not notice them. Although we have obtained good results, we still believe the system can be improved, especially in the view editor and in helping users to write expressions using EL, which will improve these results.

6

Conclusion

Contents

As a successful technique, gamification has been used to increase the motivation of users as well as their performance while performing certain tasks. Its success relies on an appropriate design to ensure its meaningfulness, mostly in the education context. The employment of gamification in education still has some flaws regarding its adaptability since the different learners could have a significant discrepancy in their needs.

For this thesis, we improved the GameCourse system by adding new features and improving the UX of the system: new configuration pages for some modules, with import and export functionalities. In the skills configuration, we provided an editor for each skill page. In the badges configuration page, we added new inputs to choose general layers to be used in all badges, as well as merging those with the base image of each badge. For this, we developed a file picker for images and files in which we can access the saved files of the course.

Regarding the view editor, we started by restructuring the views logic, and then improve the view editor to make it easier to use. Finally, we developed a suggestion system to give more support to use the EL in the view creation.

We tested the new functionalities resorting to user testing with 20 users which was helpful to understand the strengths and weaknesses of our work. We observed that our users, in general, could perform tasks regarding module configuration and views and pages management.

All the work we done has contributed to increase the system's flexibility and helped it to achieve the goal of being used in other higher education courses as it has aroused interest in other teacher of IST to use it in their courses this school year.

6.1 System Limitations and Future Work

Despite we have achieve almost every goal to improve the system regarding flexibility and usability, the system would leverage if some improvements were made. First, the mobile version for it should be taken into consideration since the UX will be improved if this is done.

Regarding the view editor, although it is working, the aspects deal with roles but not with specific users. It would be interesting if it was added the possibility to create an aspect for a specific user, for which we are already working. Furthermore, in role interaction views, when the user is the same as the viewer, for example, if I am seeing my Profile page with my information, we may want to add extra information for that user. Furthermore, the possibility of choosing a specific user to preview a view would be useful as well. For a more friendly interaction, the icons in the toolbar should be reviewed to be more comprehensible.

In the file picker, it would be very useful if it allows the creation and deletion of folders, instead of just uploading and deleting files.

The GameCourse system already has a documentation page in which we can see how views work, how to configure a part in the view editor, all the documentation of the EL, and extra information about the modules. However, this needs to be updated and reviewed to reflect all the changes that we made.

Bibliography

[Amaral, 2013] Amaral, J. (2013). Gamecourse. Master's thesis, Instituto Superior Técnico.

[Arkün Kocadere and Çağlar Özhan, 2018] Arkün Kocadere, S. and Çağlar Özhan, [U+FFFD] (2018). Gamification from player type perspective: A case study. *Educational Technology & Society*, 21:1436–4522.

[Baltazar, 2016] Baltazar, A. (2016). Smartboards. Master's thesis, Instituto Superior Técnico.

[Barata et al., 2013a] Barata, G., Gama, S., Jorge, J., and Gonçalves, D. (2013a). Engaging engeneering students with gamification. In *VS-GAMES 2013*, pages 1–8. IEEE.

[Barata et al., 2013b] Barata, G., Gama, S., Jorge, J., and Gonçalves, D. (2013b). Improving participation and learning with gamification. In *Gamification 2013*, pages 10–17.

[Barata et al., 2016] Barata, G., Gama, S., Jorge, J., and Gonçalves, D. (2016). Studying student differentiation in gamified education: A long-term study. *Computers in Human Behavior*, 71.

[Bartle, 1996] Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit muds.

[Bedwell et al., 2012] Bedwell, W. L., Pavlas, D., Heyne, K., Lazzara, E. H., and Salas, E. (2012). Toward a taxonomy linking game attributes to learning: An empirical study. *Simulation & Gaming*, 43(6):729–760.

[Brooke, 1995] Brooke, J. (1995). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.

[Brühlmann, 2013] Brühlmann, F. (2013). *Gamification From the Perspective of Self-Determination Theory and Flow*. PhD thesis.

[Deterding, 2012] Deterding, S. (2012). Gamification: Designing for motivation. *interactions*, 19:14–17.

[Deterding et al., 2011] Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. volume 11, pages 9–15.

[Dourado, 2019] Dourado, A. (2019). Gamecoursenext. Master's thesis, Instituto Superior Técnico.

[Garden and Rivera, 2018] Garden, C. and Rivera, E. (2018). Putting theory into practice: Gamification for student engagement. In *EDULEARN18 Proceedings*, 10th International Conference on Education and New Learning Technologies, pages 4563–4570. IATED.

[Gasah et al., 2019] Gasah, M., Mat Zain, N. H., and Baharum, A. (2019). An approach in creating positive emotion for children's e-learning based on user interface design. *Indonesian Journal of Electrical Engineering and Computer Science*, 13:1267–1273.

[Hassan et al., 2019] Hassan, L., Dias, A., and Hamari, J. (2019). How motivational feedback increases user's benefits and continued use: A study on gamification, quantified-self and social networking. *International Journal of Information Management*, 46:151–162.

[Johnson et al., 2016] Johnson, D., Deterding, S., Kuhn, K.-A., Staneva, A., Stoyanov, S., and Hides, L. (2016). Gamification for health and wellbeing:a systematic review of the literature. *Internet Interventions*, 6:89–106.

[Kahu, 2011] Kahu, E. (2011). Framing student engagement in higher education. *Studies in Higher Education*, 2011.

[Knaving and Björk, 2013] Knaving, K. and Björk, S. (2013). Designing for fun and play: exploring possibilities in design for gamification.

[Landers, 2015] Landers, R. (2015). Developing a theory of gamified learning. *Simulation & Gaming*.

[Leclercq et al., 2020] Leclercq, T., Poncin, I., and Hammedi, W. (2020). Opening the black box of gameful experience: Implications for gamification process design. *Journal of Retailing and Consumer Services*, 52:101882.

[Marczewski, 2015] Marczewski, A. (2015). *User Types HEXAD*, pages 65–80.

[Mekler et al., 2013] Mekler, E., Brühlmann, F., Opwis, K., and Tuch, A. (2013). Disassembling gamification: The effects of points and meaning on user motivation and performance. pages 1137–1142.

[Monterrat et al., 2015] Monterrat, B., Desmarais, M., Lavoué, E., and George, S. (2015). A player model for adaptive gamification in learning environments.

[Nacke et al., 2013] Nacke, L., Bateman, C., and Mandryk, R. (2013). Brainhex: A neurobiological gamer typology survey. *Entertainment Computing*, 5.

[Ning, 2018] Ning, B. (2018). *A UX-Driven Design Method for Building Gamification System*, pages 112–124.

[Pengnate and Sarathy, 2017] Pengnate, S. F. and Sarathy, R. (2017). An experimental investigation of the influence of website emotional design features on trust in unfamiliar online vendors. *Computers in Human Behavior*, 67:49–60.

[Ryan and Deci, 2000] Ryan, R. and Deci, E. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *The American psychologist*, 55:68–78.

[Sheldon, 2011] Sheldon, L. (2011). *The Multiplayer Classroom: Designing Coursework as a Game*. Course Technology Press, Boston, MA, USA, 1st edition.

[Sillaots, 2014] Sillaots, M. (2014). Gamification of higher education by the example of course of research methods.

[Silva, 2020] Silva, P. (2020). Implementing a new graphical user interface for gamecourse. Master's thesis, Instituto Superior Técnico.

[Strode, 2019] Strode, A. (2019). A board game as a teaching tool in a pedagogically psychological perspective. pages 5437–5444.

[Toledo Palomino et al., 2019] Toledo Palomino, P., Toda, A., Oliveira, W., Cristea, A., and Isotani, S. (2019). Narrative for gamification in education: Why should you care?

[Werbach and Hunter, 2012] Werbach, K. and Hunter, D. (2012). *For the Win: How Game Thinking can Revolutionize your Business*.

A

Entity Relationship Model of GameCourse

A.1 Previous Version

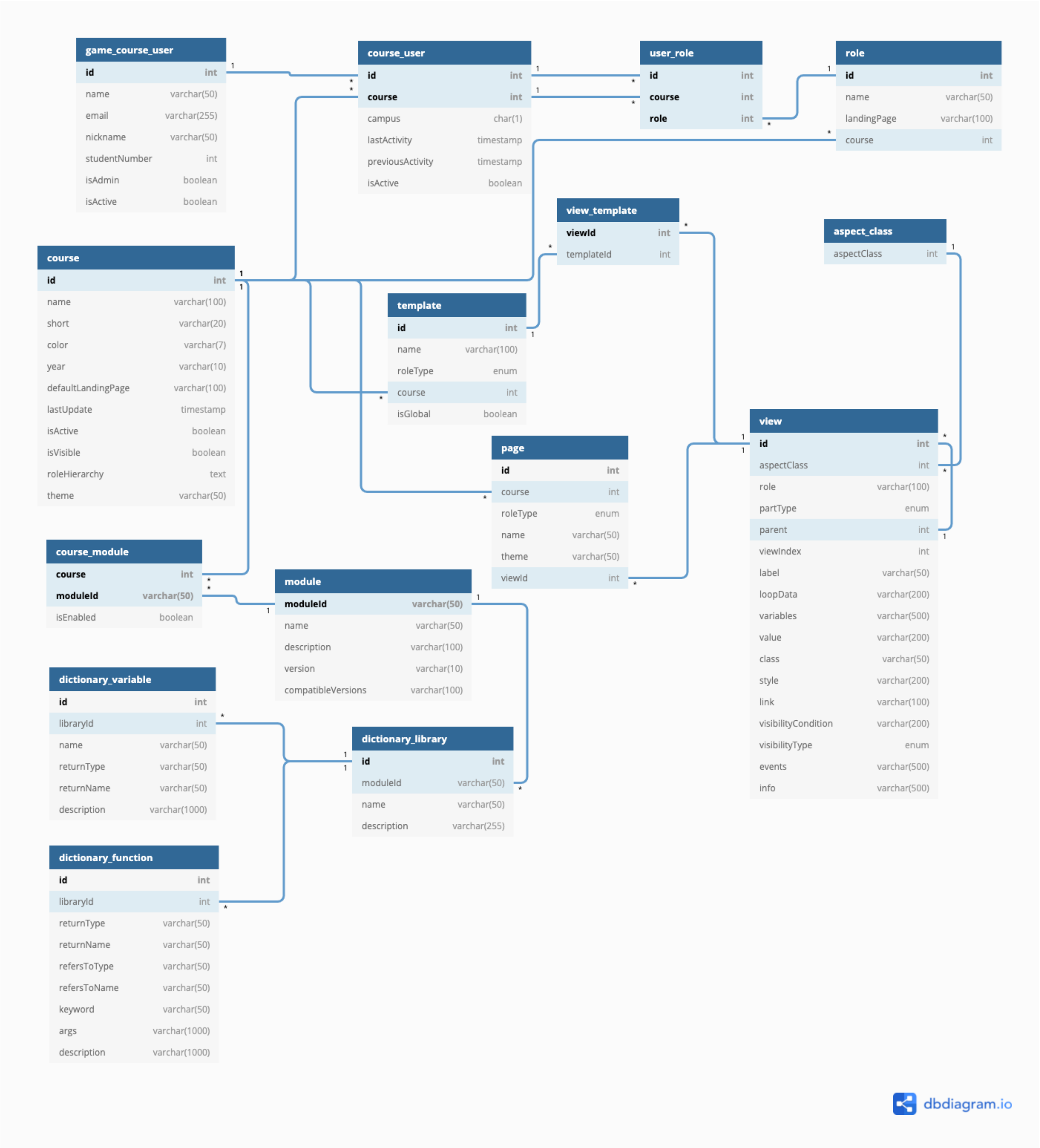


Figure A.1: Previous Entity-Relationship model of GameCourse.

A.2 New Version

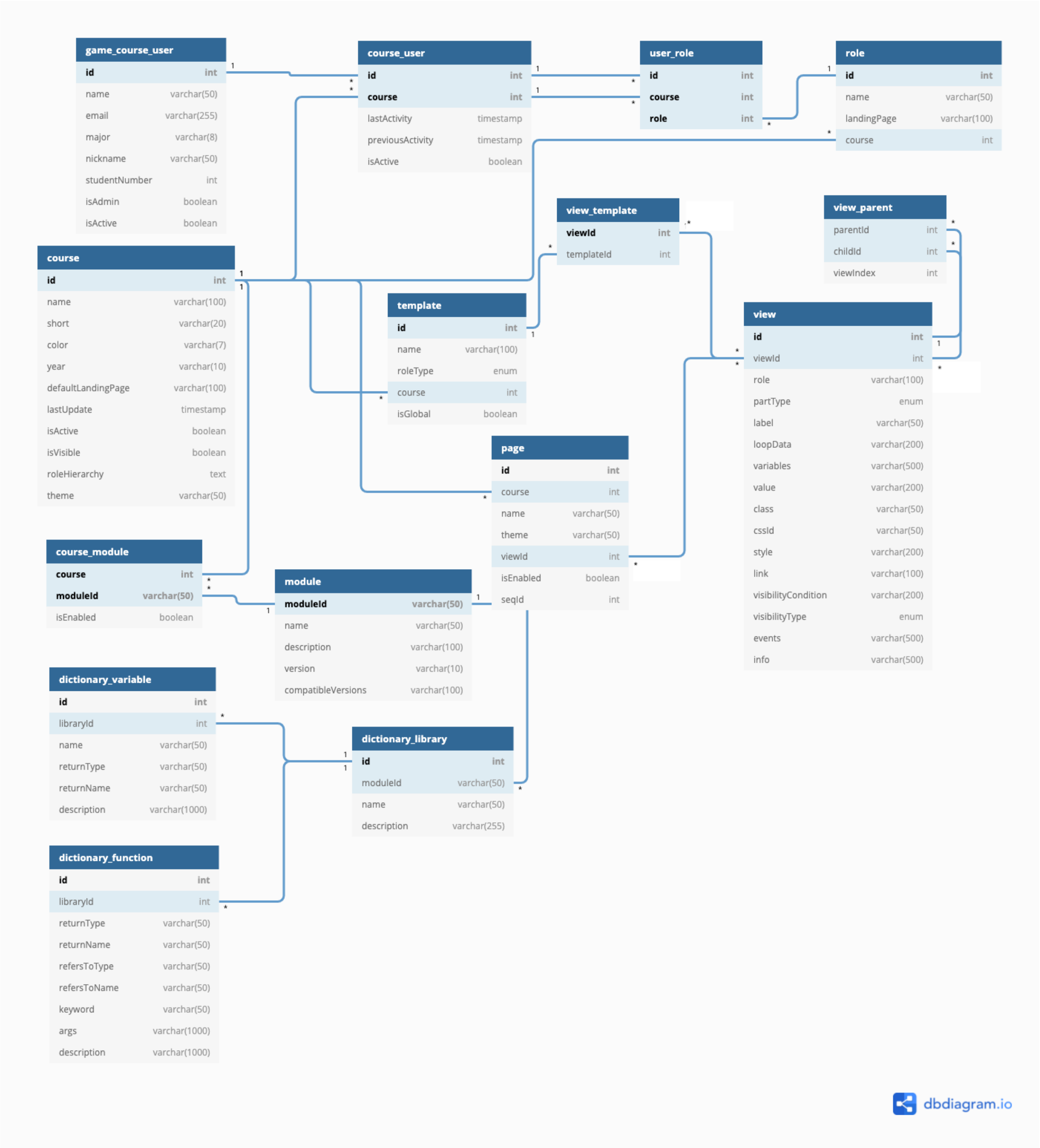


Figure A.2: New Entity-Relationship model of GameCourse.

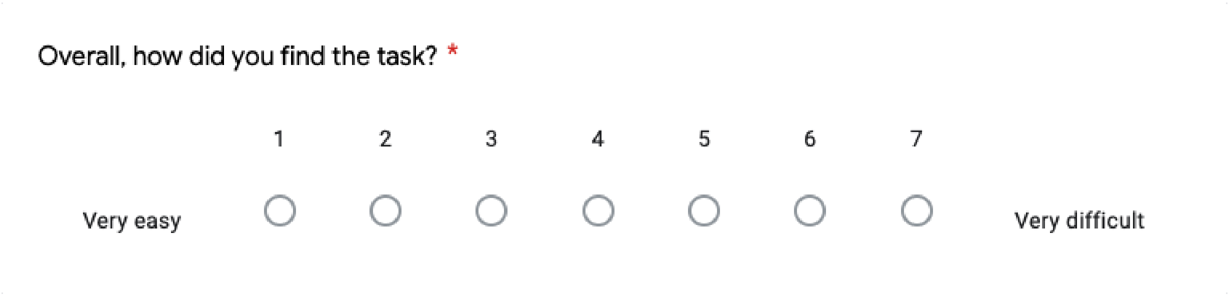
B

User Tests Procedure

B.1 Initial Questionnaire

1. Have you ever had any contact with the Multimedia Content Production (Produção de Conteúdos Multimédia) course?
   - (a) No
   - (b) Yes, as a student
   - (c) Yes, as an admin/teacher of the MCP course
2. Do you have any programming experience?
   - (a) Yes, professionally/academically
   - (b) Yes, as a hobby
   - (c) No
3. What is your age?
4. Do you allow the recording of your interaction with the system? (Microphone and screen recording only, used exclusively for the analysis of the test results)
   - (a) Yes
   - (b) No

B.2 Individual Task Rating



Overall, how did you find the task? *

	1	2	3	4	5	6	7	
Very easy	○	○	○	○	○	○	○	Very difficult

Figure B.1: Question used for rating the user's perception regarding the difficulty each task.

B.3 SUS Questionnaire

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

B.4 Final Questions

1. Did you have any trouble understanding the view editor? (e.g: Understanding icons, finding menus)
2. On the Edit Part modal, was the information provided enough to write an Expression?
3. When help was needed, did you have trouble finding it?
4. Any other suggestions? (Things that, in your opinion, could be improved)

B.5 Tasks

1. On the PCM course create a new Page
   - Name: My page
   - Template: Badges block - by badges
   - Enabled
2. Configure the module ”XP and levels” by adding a new level:
   - Title: 1

   - Minimum XP: 1000

3. Configure the module skills by adding a new skill:

   - Tier: 1
   - Name: Course Logo
   - Color: any
   - Content
     - This is my skill

4. Configure the module badges by setting the overlay for extra as the `red_border`.png image by uploading it from the computer.

5. Go to your info and change your nickname to “Mariana”.

6. Now we want that the PCM course looks differently. We want to change the look of some elements. For that, define a new style for this course by:

   - Defining a red border for the elements of the “`red_border`” class
   - `.red_border` { border: 1px solid red; }

7. On the PCM course create a new view template

   - Name: My Template
   - Role: single

8. *Inside the view editor*

   - Add a text part to the outside block

9. *Inside the view editor*

   - Create a new aspect, from scratch, for role Teacher for the outside block.

10. *Inside the view editor*

    - Create a new aspect, from scratch, for the text box for which the content is “PCM” for role Student.

11. *Inside the view editor*

    - I want the textbox for which the content is “IPM” to have the following content:
      - Student number of the user with the id=1.

12. *Inside the view editor*

    - Add the class `blue_border` to the outside block. (use “; `blue_border`;”)

13. *Inside the edit window*

    - I want this view to go through all the active courses. How would you perform this?