SAPIENZA
UNIVERSITÀ DI ROMA

QuBE - The Quiz Battle Editor:
An environment for educational game development

Facoltà di Ingegneria dell'informazione, informatica e statistica
Corso di laurea in Informatica

Candidato
Sadman Sakib Rahman
n° matricola
1632174

Relatore
Marco Temperini

A/A 2016/2017

Index

Introduction

"Computer and video games are of the utmost importance as cultural assets, as a driving force for innovation and as an economic factor."

–Angela Merkel, Gamescom 2017, Cologne
(source: http://www.develop-online.net/news/chancellor-merkel-opens-gamescom-2017/0234500)

As human beings, trivially guided by the pursuit of happiness during the course of our existences, we often and gladly seek for pleasant activities, as we associate our physical and mental well-being to our happiness.

It doesn't come as a surprise then that we're naturally inclined to combining business with pleasure, to make our activities more fulfilling. Sometimes, almost extraordinarily, we even favour these kind of activities to those exclusively pleasant that involve no business.

This tendency, less abnormal than one might imagine, is not an accident, but the result of a particular process that happens subconsciously: according to the studies made by **Mihály Csíkszentmihályi** (1990), the activities that we find more pleasant are associated with a particular state of our mind, known as the "**flow**". During the flow state we happen to be concentrated, we lose almost self-consciousness, we are fully absorbed by the activities we're doing, we have a feeling of total control, associated with an intrinsic, autotelic pleasure (Deci et al., 1985).

This state of flow can be experienced in many different environments: it may be felt by a surgeon absorbed in a delicate operation, by an athlete running in an extenuating race, or even by a **player involved in a hard video game** (Chen, 2007) (Sweetser et al., 2005). In the aforementioned activities there's a

fascinating combination of distinct factors that make the experience challenging, yet engaging and pleasant, in a singular manner.

The objective of this dissertation is to analyse video games, with the purpose of using their potentiality beyond playful scopes, focusing ultimately on building a proper environment meant for creating educational games.

An insight on the learning process

Before speaking of **educational video games**, or, alternatively, of **game based learning and/or gamification**, it is appropriate to discuss the reasons that may bring someone to think about mixing up two distinct worlds that may seem to have so little in common, if something at all. Does it make sense to merge what is a source of entertainment to what is pointed towards learning?

The answer to this question lies in a branch of psychology, known as **cognitive psychology**, its objective the study of those mental processes that we use to acquire information in our cognitive system, elaborate it, memorize it, and recover it.

Our objective, in our approach to this discipline, revolves ultimately around improving our tools for the acquisition of information regarding education: to create a digital instrument meant to teach, it is appropriate to understand the mechanics the human brain uses to learn.

Our aim here is to outline a paradigm that will allow us to create an effective application fit for our purposes, exploiting some fundamental notions of cognitive psychology, hereby listed and described.

- ## Attention

  With the word "attention" we identify a mental state in which we're actively conscious and concentrated on some elements perceived by our senses. The number of elements on which we can concentrate however doesn't match the actual number of external inputs we receive from the environment: our brain in fact has limited resources

for what concerns the ability of concentrating on multiple things at once, and to effectively use this ability, it requires to exclude other inputs when our attention is required by some element in particular (Kahneman, 1973).

Fundamentally, humans are poorly inclined towards multitasking, unless the secondary activity we're involved with is simple, or automatic, like chewing gum. If the activity is particularly engaging, on the other hand, like doing complex calculations, even a simple activity such as contemporarily pressing a button when a red light turns on can become difficult.

As we can guess, it is important to exploit the potentiality and the limits of the human being acknowledging his ability to concentrate at best on one task at a time. Forcing the user to concentrate on multiple objectives at once at the same time, on the contrary may be deleterious, and may cause a condition known as inattention blindness (Mack et al., 1998) (an example of this issue is shown in the famous "gorilla experiment", where the participants are asked to count the number of times a team of basketball players pass the ball to one another; during the passages, a man in a flamboyant gorilla costume makes his way through the players; the people involved in the counting often can't see the gorilla) (Simons, 1999).

- ## Memory

Memory fundamentally is what allows us to elaborate, store and recover information, and it's divided in three kinds: sensorial memory, working memory, (also known as short-term memory) and long term memory.

Sensorial memory is involved in perception, and gathers sensorial information for a relatively short time (like a fraction of second) without being consciously processed. The persistence of vision (like the one created by a fast moving image), for example, is caused by sensorial memory, which allows us to perceive reality in 24 frame per second, similarly as if it were an animation.

Working memory, also known as short-term memory, allows the brain to store information for a few minutes and the manipulation of a limited quantity of information, as much as necessary to do an activity, like a calculation that requires to remember momentarily some numbers. This kind of memory requires a lot of attention: exceeding the cognitive capacity of an individual, as told previously, can jeopardize the learning process (Atkinson et al., 1971) (Baddeley et al. 1986).

Long term memory, last but not least, keeps track of all the data concerning knowledge and the abilities we possess. The limits of this kind of memory are unknown, and it is believed that it can store information for an indefinite amount of time, although it is possible to forget; it is indeed demonstrated that the capacity to store information decreases with the passage of time, and the maintenance of information, without a proper emotional involvement or meaning, can be fragile. Some variables have a great positive impact on the capacity of memorization, like repetition. The human brain, on the other hand, has a natural inclination to distort memories, and to cause memory lapses (Ebbinghaus, 1885).

For such reason, during the designing of a learning system, it is fundamental to keep in mind these limits: **it is useful to decrease the**

total amount of information to remember at the same time, and redistribute it in a modular manner. Furthermore, using stored notions favours the consolidation of information, (Sweller, 1994).

As can be guessed by the latter sentence, one of the most useful methods to learn is through activities that require the use of the information.

- ## Perception

Perception involves all the mental processes that allow us to perceive the environment that surrounds us, and build a mental elaboration of it. Naturally, all the processes involved in this scope represent the lowest level of the infrastructure that connects the sensations to the mental cognition. Our own representation of reality works in the opposite direction as well, cognition can in fact influence sensation: the "Save" icon, represented by a floppy disk in text editors such as Microsoft Word probably doesn't make sense to young people, until they don't find out its meaning through a computer. This example shows also that perception is subjective. It varies according to the context in which the input is presented, and the knowledge or the expectations of the one to which the input is directed.

Consequently, **a software user can have a vision that may differ from the one belonging to whoever designed the software.** To avoid this kind of dissonance, it is important to develop following the paradigms of User Centred Design (Hodent, 2014), pointing towards designing a system that is suitable to the needs of the users (Isbister et al., 2008) (Johnson, 2010).

Some useful guidelines to reach this goal can be found in the principles of the Gestalt psychology (Wertheimer, 1923).

- **Good form**: the perceived structure is always the simplest.
- **Proximity**: the elements are grouped in function of their distance.
- **Similarity**: tendency to group together similar elements.
- **Good continuation**: all the elements are perceived as belonging to a continuous and coherent set.
- **Common fate**: if the elements are in motion, they're grouped by the ones moving coherently.
- **Background-shape**: all the elements in a particular area can be either interpreted as objects or part of the background.
- **Induced movement**: a reference scheme formed by some structures that allows the perception of objects in motion.
- **Meaning**: in case the stimuli are ambiguous, a good perception relies on the information caught by the retina.

• Emotion

It is common belief that the emotional component in the design phase of a product is a substantial aspect in determining the success of the product itself, more than the practical aspects. This emotional side, in video games, is implemented through **good aesthetics, a catchy soundtrack, or an engaging narrative** (Przybylski et al., 2010). Nonetheless, another aspect needs to be considered as well: the "**game feel**". Game designer Steve Swink (2009) describes this concept as "the feelings of mastery and clumsiness, and the tactile sensation of

interacting with virtual objects". Keeping in mind the game feel means putting attention on the design of elements such as the camera, the controls, and the characters; a camera with a restricted field of view, for example, may cause a feeling of claustrophobia, inappropriate if the game is meant to give an experience of serene exploration; on the other hand, it may come useful if the game belongs to the horror genre.

Another important factor to consider, during the development, is the motivation of the players towards which the game is addressed: **the appeal and positive effects of video games are, as a matter of fact, based upon their potential to be able to satisfy some elementary psychological needs.** A game that satisfies the need for **competitiveness, autonomy and relatability**, as a consequence, has the requirements to be really engaging.

Competitiveness allows the players to comprehend their own sense of mastery of the game, and gives them a feel of progression (e.g. in FromSoftware's *Dark Souls* players are required to have an increasing mastery to proceed in the game).

Autonomy offers the players the possibility to make significant choices, and gives the opportunity to express themselves (e.g. in Obsidian Entertainment's *Fallout: New Vegas* it is possible to change the course of the plot in a non-linear manner, according to the choices made in the game).

Relatability, at last, is basically aimed towards satisfying the need to feel connected to other individuals. This component often gets implemented through the insertion of multiplayer functions, that

allow the players to interact with one another (e.g. the team chat function in *Overwatch*, by Blizzard).

A solid motivation and a strong emotional component have an extraordinary impact on the players' liking, and not only: these aspects influence the learning and the quality of the maintenance of the information.

Thanks to these aforementioned notions of cognitive psychology, we may positively state that there are indeed valid reasons to exploit video games for learning purposes; nonetheless, it is important to respect some conditions. The human brain's potentialities have to be used taking account of its limits, and designers should take into account the end-users' needs. To grant success, it is suggested to run usability tests, referring to pre-existent heuristics (Desurvire et al., 2004) (Nielsen, 1994).

From theory to practice

We've shown that theoretically speaking, it is possible to use video games to improve learning; nonetheless, this isn't enough to prove that they're effective in practice. To reinforce our argument, it is convenient to gather empirical evidence and analyse already existing games related to education, to verify their effective potentiality (Fullerton, 2014). Furthermore, we'll try to extrapolate more guidelines from those products that may reveal as effective, while, eventually, we'll discard those ideas that prove ineffective or unrealizable.

First of all, we'll analyse the games by dividing them in three categories, according to the educational gaming models outlined by the studies of Richard N. Van Eck (2006). These paradigms are distinguished as following:

- **Serious educational games**, created specifically by a team of educators and programmers
- **Commercial off-the-shelf games**, not necessarily meant for education, integrated in the courses by the teachers
- **Student-made games**, often created under the supervision of teachers during programming courses

Nothing implies that these categorizations are absolute: it is indeed possible the existence of hybrid variants that integrate one or more aspects that distinguish these paradigms.

• Serious educational games

The first model we'll analyse represents generically commercial titles developed specially for educational purposes.

Although non-commercial serious educational games do exist, they're often exclusively implemented in particular environments, such as schools and colleges; being niche, these kind of games are not really popular, and consequently, it's hard to find documentation regarding them. Other aspects that explain their unpopularity are related to the lack of use of resources to guarantee a good design. It is indeed hard to find qualified programmers and educators willing to spend a great amount of time without retribution for the development of a software not meant for profit. This doesn't imply that there can't be valid serious non-commercial games: later on, in fact, we'll analyse an example of a game belonging to this category.

For what concerns commercial educational games, often made with the contribution of relevant software houses, on the contrary of their non-commercial counterparts they require a discreet budget for the employment of programmers, game designer, and above all, educators, to guarantee a satisfying final product, that fits not only educational purposes, but the laws of the market as well: the product has to be designed to be popular among the end-users, to assure an economical comeback to the developers. **This model, although doubtlessly advantageous for many reasons, has sadly some negative sides inherent to the time and the money required for the development.**

To ensure a major cognition of cause, in the next section we'll study and analyse some particularly popular titles among Serious Educational Games. To evaluate them correctly, empirical analyses of the aforementioned products have been run through individual tests where possible.

❖ **Dr Kawashima's Brain Training**

This title made for Nintendo DS, first of a homonymous series of video games made by Nintendo since 2005, is one of the most popular examples of educational games available on the market. The title is based on the tests made by neuroscientist Ryuta Kawashima, resident professor in the University of Tohoku, who appears in the game as a polygonal avatar.

Fundamentally, the game's main goal is to allow players to calculate their own brain age through math exercises, mnemonic trials and Sudoku. Passing the trials, in function of the correctness of the answers and the time spent answering, allows the player to self-assess their own performance through a value named "brain age". The less this value is, the better is considered the performance of the player. This title, centred on improving one's mental abilities rather than learning, seems to be particularly effective on individuals who are more than twenty years old. Some studies furthermore suggest its employment to prevent and fight against Alzheimer syndrome at a late age (Nouchi et al., 2012). Nintendo, though, has avoided releasing officially any declaration about the potentialities of the game on a scientific level, underlining how the company's sole goal is **entertainment**. The game modes inside Brain Training, except for Sudoku, are defined by the following mini games.

1. **Calculations X 20:** One question will appear on the top screen, and the player must hand-write the answer on the touch screen. There is a total of 20 questions, including addition, subtraction, and multiplication.

2. **Calculations X 100**: same as Calculations X 20, although with 100 questions instead of 20. It features a Hard mode, which has the same mechanics but adds division problems.
3. **Reading Aloud:** the player is given an excerpt from a classic story such as *The Legend of Sleepy Hollow* or *Little Women*, and is tasked with reading the story aloud to see how quickly he or she can do it. The player progresses through the excerpt by pushing Next, until he or she reaches the end of the excerpt. If the player pushes Next too quickly, the puzzle will end prematurely.
4. **Low to High**: it features several boxes on both screens, each in the same pattern as each other. The game will count down at varying speeds, and when it hits zero, numbers will appear in these boxes for a short period of time. Afterwards, the player must touch the boxes on the touch screen from the lowest number to the highest by memorizing the numbers on the top screen. Afterwards, the game will introduce one puzzle after the other in a similar fashion. The quantity of boxes to memorize increases after each correct answer, and decreases after each incorrect answer, with the minimum quantity of boxes being four, and the maximum being 16.
5. **Syllable Count**: it shows several phrases, one after the other, on the top screen, and the player must write the number of syllables in each phrase on the touch screen.
6. **Head Count:** features a group of people on the top screen. After a few seconds to allow the player to count the number of people, a house falls over them. The player must watch the screen carefully, as the people inside will leave the house and more people will enter the house. This will eventually cease,

and the game asks the player to write down how many people are currently in the house. The puzzle gets more difficult as the player progresses in it. There is also a hard mode in which people also come in and out of the chimney.

7. **Triangle Math:** has a series of mathematical equations that the player must solve. It is designed similarly to the Calculation puzzles, in that the equation appears on one screen, and the player writes the answer on the touch screen. The equations involve three numbers and two mathematical operations (e.g., 3 + 4 + 8 or 3 − 4 + 8), and are solved by performing the first operation, and then the second. This also features a hard mode where an extra tier is added to the triangle.
8. **Time Lapse:** two analogue clocks are displayed (e.g. one at 2:45 and one at 7:30), and the player is required to calculate the difference in time between these clocks.
9. **Voice Calculation:** it is similar to the Calculations puzzles. However, this puzzle requires the player to speak the correct answer into the microphone instead of writing it on the touch screen, similar to the Stroop Test (Stroop, 1935).

It is important to notice that Brain Training offers **diversified gaming styles**: this characteristic makes the player's experience variegated, avoiding boredom caused by the repetition of the same pattern of gameplay. From the latter part comes another consideration: **the game must entertain to keep active the player's attention**.

In the main menu it is also possible to interact with Dr Kawashima, using the integrated microphone. According to the words

pronounced, the avatar's expression will change. This detail, although not functional to the purpose of the game, makes the game more appealing, merely speaking of design. **It is interesting the idea of using non-functional elements to make the game more engaging**.

The product seems to be particularly fascinating because of the interface and the gaming method, as it requires a console meant principally for playful use. **An interesting concept behind this experience is to develop ad-hoc software for platforms centred on giving entertainment**. Brain Training, in fact, even being a game with simple mechanics and a low commercial potential compared to other games on the same platform, owes its popularity to the masterly use of the Nintendo DS technology. As a further proof of the game's commercial success, it is worth mentioning that three sequels have been released up 'til now in Europe: *Dr. Kawashima's More Brain Training* (released in 2007), *A Little Bit of... Dr Kawashima's Brain Training* (released in 2009), and *Dr. Kawashima's Devilish Brain Training: Can You Stay Focused?* (released in 2017).

- **Carmen Sandiego**

Although this series of educational products isn't well known in every country worldwide, in most English-speaking countries, particularly in the USA, the situation is quite different. The fictional figure of Carmen Sandiego, in fact, is known on the American soil as one of the most celebrated icons of the **Edutainment** sector.

With this word we refer to a particular kind of approach towards teaching known as **Educational Entertainment**. To this tag belong many different kinds of media, spacing from cartoons to TV shows, and obviously, video games as well.

For what concerns the Carmen Sandiego series in particular, technically it can be defined as belonging to all the aforementioned media quoted before; this apparent paradox is basically related to the fact that Carmen Sandiego is a product that uses **transmedia storytelling**, (Jenkins, 2007).

Fundamentally, products that present this characteristic utilize the potentiality of other entertainment media to expand their own popularity, in a symbiotic manner. This approach, although more functional to marketing than software design, is interesting because, as it can be shown through the Carmen Sandiego series, can be implemented later, following some precautions during the development phase.

In fact, although this series of products originated from a video game (*Where in the World is Carmen Sandiego*, released in 1985 by Brøderbund for Apple II), since its release many TV game shows and cartoons based on the brand have been made ever since. In 2015 the twentieth title in the video game series has been released for iOS and PC, *Carmen Sandiego Returns*, while Netflix has announced a new animated TV series coming in 2019.

So, what makes Carmen Sandiego such a successful product? **The use of an iconic character catches the eye**: to be fair, in the Dr Kawashima's games, analysed in the previous paragraph, we've got something similar as well. An icon, if chosen well or created

wisely, can be adapted in numerous contexts. These notions about the power of cross media do not only work in one direction: as the fame of a video game can bring to the creation of non-video game media (*Halo: Fall of Reach*, by Eric Nylund, inspired by Bungie's *Halo*), similarly the opposite can happen: there are indeed many games inspired by already existing products, like movies and novels; this idea has even been applied for educational purposes as well (*Disney's Learning with Mickey*, 2002).

We won't dwell on the latter possibility though, as it goes excessively beyond the purpose of this dissertation, and the possibilities we have: as a matter of fact, **among the many problems in using an already existing icon, intellectual property comes in the way, and the costs related in acquiring it**. Without further delay then, we shall proceed with the analysis of "Where in the World is Carmen Sandiego", from where the whole series began.

The game's purpose is to track down and capture the goons that work for Carmen Sandiego, an infamous international criminal. The final objective is to capture Carmen herself, all within a virtual time limit per case. The player, who plays as a detective, starts the game by going in the countries where the crimes have been committed, to search for clues to find out where the criminals might have gone. This clues come often in the form of play on words that address the player towards the destination of the fugitive.

These destinations, constituted by famous cities of thirty countries, are represented by famous sightseeing places from the country we're visiting.

Every case starts with the user being warned of an incredible heist: the player must go to the country where the crime has been committed, and will have to investigate by asking questions; once the player is sure that he has enough information, he'll have to go to the next location that fits the correct deduction. Every action, like asking a question to a testimony or making a journey, will decrease the time left to win, requiring a particular attention towards every choice made.

As can be easily guessed, this game is about teaching geography: using the same leitmotiv, that is the hunt of the wanted man, other subjects have been taught through the other titles of the series, like history (*Where in Time Is Carmen Sandiego?*, 1989), astronomy (*Where in Space Is Carmen Sandiego?*, 1993) and many more, according to the possibility of adapting the same narrative canvas.

The game's user interface, pretty simple, takes inspiration from the style common to the point-and-click genre, themselves being

modernized variants of the interactive fiction, inspired again from gamebooks, known otherwise as choose-your-own-adventure: you proceed through the story by making choices, building the story yourself.

A fundamental characteristic of the Carmen Sandiego series that needs to be underlined, therefore, is the presence of storytelling: **although quite elementary in its essence, a solid storytelling contributes immensely to the player's engagement**. It is a source of motivation, and has a relatively influent emotional impact. We'll eventually consider this important notion during the development of an educational game.

- **The Oregon Trail**

This title, among the oldest between the games we're considering, will here be quoted and shortly discussed for its historical importance: it is in fact considered as one of the milestones of educational games, and its heritage is yet vivid in the United States: one of the game's phrases, "You have died of dysentery", is yet printed nowadays on t-shirts.

Published in 1971 by the Minnesota Educational Computing Consortium, The Oregon Trail allows the players to observe in an active way the experience of the pioneers that went through the Oregon trail in the 19th Century. In this case, the narrative component is strictly related to the historical events, and doesn't require too much fiction. Beneath certain aspects, this is a great example of "**learning by doing**", one of the most suggested methods to strengthen memory (Lesgold, 2011).

For what concerns the game style, it blends the dynamics of a graphic adventure with the features of a management game. The player's goal is to grant the survival of the five emigrants on the Conestoga wagon on their way to the West, avoiding their deaths by making bad choices. These choices are often about issues such as the search for supplies or places to stay.

The quality of The Oregon Trail as a video game has given him a discreet reputation: among some of the video game critics, in fact, the title is counted as "one of the games that must be played before dying" (Mott, 2010).

- **Explorables**

The Explorables, so called by the author of the project, Nicky Case, represent an interesting case study regarding serious educational games. Unlike the other titles analysed so far, the Explorables **are not educational titles meant for profit**: in fact, they are non-commercial educational games. They are therefore not particularly well-known, beyond what is derived from **viral sharing through social networks**, since they are not subject to advertising and marketing moves. I, myself, came across these through an external

link shared on Facebook. The games themselves do not stand out for an extraordinary visual component or for a particular longevity; however, while maintaining a sober minimalism, the Explorables effectively succeed in their main goal: to make learning fun.

These **small mini open source games, made by a community of programmers who voluntarily participate in the project**, consist of individual lessons on singular topics that come from different disciplines, taught through the subterfuge of gaming.

As an analysis sample, "The Evolution of Trust" was chosen among the Explorables, a small game focused on explaining how dynamics of trust work in society. The game is short, and it uses the concept of learning by doing excellently, giving the player the opportunity to experience the effect of human behavioural dynamics through an eye-catching interface where the player simply inserts the parameters and observes the outcome of the evolution of society through simulation.

Among the design decisions some simple but fundamental choices undoubtedly stand out: **the game does not, in the first place, make use of some abused gamification mechanics** (Bogost, 2011), which we will discuss later, like goals, scores, or rankings. The author of the project, on the contrary, openly criticizes this concept, pointing out how it is about modifying *behaviour*, while the aim of the Explorables project is to change *knowledge*.

In addition, the user's evaluation is ignored, focusing on learning. This approach to teaching, much more in line with some

pedagogical principles, is important to consider during development.

Within a certain perspective, it is reasonable to say that interactive explanations are an evolution of the presentations made with PowerPoint. In order to offer a greater understanding to whoever is interested in this project, here is the link to it: http://explorabl.es/

• Commercial Off-The-Shelf games

Commercial Off-The-Shelf (COTS) games include video games already available on the market. In fact, even some Serious Educational Games may fall into this definition, such as those belonging to the Carmen Sandiego series; in Van Eck's three paradigms, however, the distinction is not focused on the educational medium, but on how it is used and implemented in the course by the teacher. In the case of Serious Educational Games, the educator must be an active part in the creation of the product; this approach obviously is extremely costly in terms of time and money. **In the Commercial Off-The-Shelf approach, the educator exploits the existing game, and tries to adapt it to his class**. According to Van Eck, **this paradigm is the best in terms of effectiveness, cost and time**.

These games do not necessarily have to be conceived as Serious Educational Games: there are games that have educational potential, even though their purpose is not teaching. Titles such as Assassin's Creed, Europa Universalis, Age of Empires and Civilization, for example, include remarkable notions of history, Age of Mythology presents a particular approach to Greek mythology, The Sims can be exploited to teach sociology, and Portal has an excellent sandbox

structure where to play with physics. In the following part we'll study an interesting use case of this paradigm.

- **Making History**

  Making History: The Calm and The Storm (published by Muzzy Lane for PC, 2007) is the first title of a series of video games that let live the World War II events through a turn-based strategy game.

  In the field of education, this title has been used by hundreds of American schools for years; among the many experiments conducted to verify its effectiveness, we will consider the one applied by Mr Irvine, a high school teacher living in a rural Midwest area (Watson et al., 2010).

  This teacher, using Making History, has succeeded in passing from the classical teaching model focused on the teacher, to a **student-centred teaching model**, deftly implementing game sessions within the school calendar, cutting out a week of time specially for playing and discussing the knowledge gained or consolidated through the game. As it is clear from this information, **to effectively apply the COTS paradigm, it is crucial for the teacher to be willing to adapt his or her teaching to the video game; Similarly, the developers of such games should aim towards meeting the educators' needs, and design their products consequently**.

  During the gaming week, students were divided into teams representing the nations involved in the global conflict, each one facing their own, historical problems, such as economic crises, popular revolts, state strikes. Each group was confronted with the

consequences of these events within the game, and players found themselves discussing and cooperating with other nations to solve them.

All of this, under the careful supervision of the teacher, proved to be very effective from the learning side: students were emotionally involved, either for the intrinsic fun of play and for the interaction with others players, in line with the notions of cognitive psychology previously exposed.

This experiment, while being extremely fascinating, cannot, however, be considered a general example that represents universally the outcomes of such approaches to education. There are undoubtedly circumstantial factors that have favoured the success of the experiment, and other factors that might have undermined the effectiveness that have gone unnoticed (Watson et al., 2010).

To establish a valid functioning paradigm, it is important to establish a good heuristic, conduct further studies on it, and raise awareness among teachers, outlining guidelines to follow for those who just want to try teaching through video games.

Another hindrance to the application of this approach is the common mentality people share about the concept of learning through video games: even today in American schools this idea is in fact linked to an individual experience like The Oregon Trail: the student plays and learns on its own, while the teacher has a mere passive role during the learning phase.

The limits of this approach, as can easily be understood, are at least so far determined by the teachers and, in minor part, by students: **it is plausible that a teacher may not be familiar with video games, as a student may not be used to certain kind of games**.

Given the non-irrelevant effectiveness of the COTS approach shown in the aforementioned example, the best hope we can afford is that more and more important software houses will eventually invest in the education sector, and that similarly the schools will spend more money to boost the use of educational games in classes.

- ## Student-made games

This kind of approach is very particular, and more niche than the approaches listed above. In this model, teachers guide students to the creation of a game as an active part of the learning process (Sheldon, 2011). Although this paradigm is not costly at all in most cases, the problem here is time: for a teacher, in fact, learning and teaching knowledge that does not necessarily relate to his or her course can be tiring.

This approach, much more common in some areas such as programming, generally consists in the use of related development languages and environments such as *Scratch*, *GameMaker*, *Gamestar Mechanic*, *Python*, *Alice* and *Adventure Maker*, or modern counterparts such as *Inkle*, *Pixel Press*, and *Tynker*, in order to design a game.

As use cases of this educational model, in this dissertation, we will consider two examples utilized in the Computer Science Course at Sapienza, University of Rome.

- **Tag Invaders**

  During the course of Programming Methodologies taught by prof. Roberto Navigli, in 2016, students were asked to individually design a level of a game inspired by Space Invaders, nicknamed "Tag Invaders," using LibGDX libraries integrated into Java.

  Each student was given instructions on the contents to be included in the level to be designed, such as a level where affected enemies fall down according to gravitational force. Along with these guidelines, source code packages, Java libraries, and precompiled code parts were also provided to optimize the job. Students, to effectively overcome what was in fact a homework, had to adapt to the existing framework provided to them, by developing the proper approach to programming independently. Students who until then had to work exclusively on code written by them, had to adapt to code written by others, and consequently learned the importance of writing clear and comprehensible code accompanied by comments where necessary.

  Outside the class, moreover, students found themselves working together, discussing and sharing ideas about the right approach to complete the goal assigned to them, as far as it was allowed. Every student was free to offer a trial of their own game to their colleagues, adding a slight touch of implicit competitiveness in the development of a level that was interesting, not counting the intrinsic fun caused by playing at the designed levels. Lastly, it is worth noting that students have been able to use the knowledge gained during the course to complete the level. As can be fairly

concluded, Tag Invaders is an excellent case of "learning by doing": **although the idea of leaving the development in the hands of the students can be risky, it is undoubtedly true that in a controlled environment it can be effective for learning**. We will therefore take into account the benefits of this paradigm.

- **The Type Tournament**

  The Type Tournament doesn't properly represent a video game, not being a digitally implemented game. However, its mechanics and the positive feedback it has received from the participants makes it undoubtedly worthy of mention.

  This game, optional part of the final exam of the Programming Languages course taught by prof. Cenciarelli in 2017, is structured as follows: given a possibly equal number of students, students are divided into pairs; if the number is odd, the professor may personally compensate for the absence of an opponent for the "excluded" player; alternatively, the extra player goes automatically to the next round.

  Tournament participants, following a group structure, must challenge their opponents by proposing a type to be decoded. The type must comply with some rules, and it can't be too long. Each player will then find himself with an abstract code string that determines a type; main goal is to declare the type made by the opponent before the opponent identifies the type that the player has made.

  The first player identifying the type must show the result of his deduction to the opponent; if he believes the player is right, he has

lost; if not, he will have to prove that he knows how to infer the type he himself created. If the opponent fails, the teacher declares the first player as the winner. As a final prize, the winner of the tournament will automatically receive a high mark, without having to go through the oral exam; as a bonus, though, the professor can give an even higher mark if the winner wants to do the oral exam anyway.

The Type Tournament, as can be noticed, allows students to create the game in a simple but intelligent way, without leaving to much space to go astray. Players basically create the game, providing their opponents with the types on which the challenges are based. At the same time however, they cannot go too off track.

The game is fully in line with some principles of cognitive psychology: it provides a valid motivation (e.g. an excellent final vote), and promotes competitiveness.

In contrast, it is worth noting the time required by the Tournament: the deduction of a particularly elaborate abstract type can, in fact, take a long time. Since the game does not set limits on it, a challenge can last for a really long time.

In both observed cases should be noted that **the framework in which the students had to develop the game had been provided by educators**: to be successful, this educational approach requires that teachers spend their time properly implementing a structure in which students can have creative freedom without exceeding the goals of the course.

An insight on gamification

Up to this point of the dissertation, the concept of gamification has been quoted a few times, and in some cases, it has been spoken of in critical tones. However, it is important to talk about it before proceeding with the development phase, as we may extract some useful notions from studying it. First of all, it is necessary to establish what it is: in general, this term means **the use of playful elements in non-playful contexts**.

The purpose of gamification, of course, also varies from context to context: it is used in many areas, ranging from education to sports, including business. There are many valid gamified applications today: a trivial example is Runtastic (Novak, 2015), a mobile application that provides goals to those who want to keep fit, with rankings to compare your performance with friends. The reason why these elements are implemented is an easy deduction: it tries **to give a motivation not necessarily intrinsic to the context**, hoping to get the typical benefits of the video games described in the preceding paragraphs, including **emotional involvement**.

The problem often lies in the fact that **those who abuse of the concept of gamification ignore elementary game design concepts**, misunderstanding what the real attraction of video games is. This is due to various reasons, first of all a **lack of cognition of cause**, such as poor knowledge of the products in question, and a **possible ignorance of brain mechanics that affect cognitive psychology**.

Basically, anyone who exploits gamification, especially in the educational field, acts in the belief that it is a simple solution to a complex problem; As the study of algorithms teaches, however, a rudimentary solution is not necessarily efficient.

This is what happens in cases of poor implementation of gamification: many educators, deceived by an erroneous perception of video games, focus on **superficial elements such as rankings, experience points, and goals** with no regard to storytelling, game context, or the game itself. According to game scholar Ian Bogost, **whoever has this superficial approach confuses the magical magnetism of games with the simple drive offered by external incentives**.

Superficial gamification, as can easily be guessed, causes little motivation, miserably failing to offer one of the major requirements required by a good game. Unless there is already a good reason to accept gamification (such as wanting to be fit, in the case of Runtastic), it will prove to be a useless accessory.

Similarly to the COTS paradigm, however, **gamification, if implemented correctly, can be one of the most powerful methods to educate in terms of effectiveness, time and cost**. It is therefore appropriate to analyse the mechanics and the advantages of gamification in depth.

First of all, we need to determine for good what are the playful elements we can use in an educational context:

- **Objectives**

  The objectives of a course can easily be tailored to the objectives of a game, transforming the acquisition of a concept into a mission.

- **Interactive activities**

  Necessary to achieve a goal and to complete the challenges. The activities of a course can be re-adapted to simulate game activities. A trivial example: using quizzes instead of tests.

- **Characters**

  Characters include Avatars and Non-Player Characters. Although NPCs are more difficult to integrate into a gamification process, avatars are easier to implement. A simple virtual identity case is the use of a nickname. Being anonymous in a public ranking can avoid embarrassment when a player finds himself in a low position.

- **Rules**

  They determine how the player and gaming environment interact with each other through game resources, objects, and relationships. A part of the rules is the one that sets the conditions for progress: how can the player earn points, and what are the conditions for victory or defeat? These can easily be implemented in an educational setting, for example by transforming the votes into experience points, which are added by providing a score that gives the player a position in a ranking.

- **Levels**

  They can be implemented in many ways.

  They are usually represented by parts of the world where the game is being played. The levels, in this case, equate to the set of lessons that relate to a single subject.

  Levels can alternatively represent the player's level of experience, based on his score. In this case, the level may result in the final vote.

Finally, the levels can refer to the difficulty of the game. A high level would mean the skill required to overcome a difficult task.

It is believed that the easiest method to integrate the concept of levels in the educational field is to reshape them according to scores and votes.

- **Balance**

  In order to provide an enjoyable gaming experience, the various playful elements must be in balance.

  For example, the system that assigns a score for a given activity must provide an adequate score, which has to be neither too low nor too high depending on the activity being performed.

  The difficulty is also subject to balance: as players increase their skills, it is necessary to increase the difficulty of the course, in order to match the students' ability and to avoid boredom caused by too much easiness; at the same time, however, it is important to avoid creating situations that are too difficult to handle, to prevent anxiety.

- **Luck**

  Luck or randomness are part of the most famous gaming mechanics: some successful games are in fact based on randomness, such as Monopoly. Usually, however, players do not like being subject to fate, and would rather prefer their success being determined by their abilities. A discreet way to implement a Luck factor in a course would be to use a dice to select who would make a class presentation.

- **Collaboration**

  It's a form of interactivity. It can easily be implemented in a course, with a collective project, or a team challenge.

- **Competitiveness**

  Not all games require competition, and not all players love competing; in fact, a rash use of this concept can have toxic results. However, competition is considered one of the key factors in the entertainment that can provide a game, and ignoring its benefits can be a questionable choice.

  Competition is not commonly used in education. Examples of implementation of this concept are in the use of rankings, quizzes, or debates organized between students.

- **Game aesthetics**

  Many modern games have an extremely impressive graphic component. However, it is complicated to create an educational title that has the graphics of a “triple A game”, as you say in commercial jargon. It’s easier to stick to the minimalist graphics of titles created by independent developers, also known as “indie games”.

- **Storytelling**

  Games do not necessarily have to be focused on storytelling (such as chess, or any other virtual deployment of similar board games), but in some areas the wise use of this concept can be a discrete source of motivation. Transforming the course into a hero's

adventure is a method, as well as creating an avatar that evolves during the game.

- **Feedback**

  Unlike in video games, it is difficult for a teacher to provide feedback to students, unless they are simple, such as the "Successfully uploaded" message when uploading a homework on a professor's site. A practical solution consists precisely in moving part of the learning environment to a virtual platform.

- **Risk**

  Games entertain because they provide a safe environment where players can take risky actions: a player may in fact fail to complete a mission numerous times before completing a goal. This mechanics, however, can become overly accommodating if there are no incentives to improve, and change the player's behaviour. Examples of games that skilfully exploit punitive mechanics are those that require the use of limited lives, such as the famous Super Mario: the fear of having to start over a whole level acts as a deterrent, prompting the player not to make too many mistakes and at the same time, without denying the possibility of making mistakes more than once.

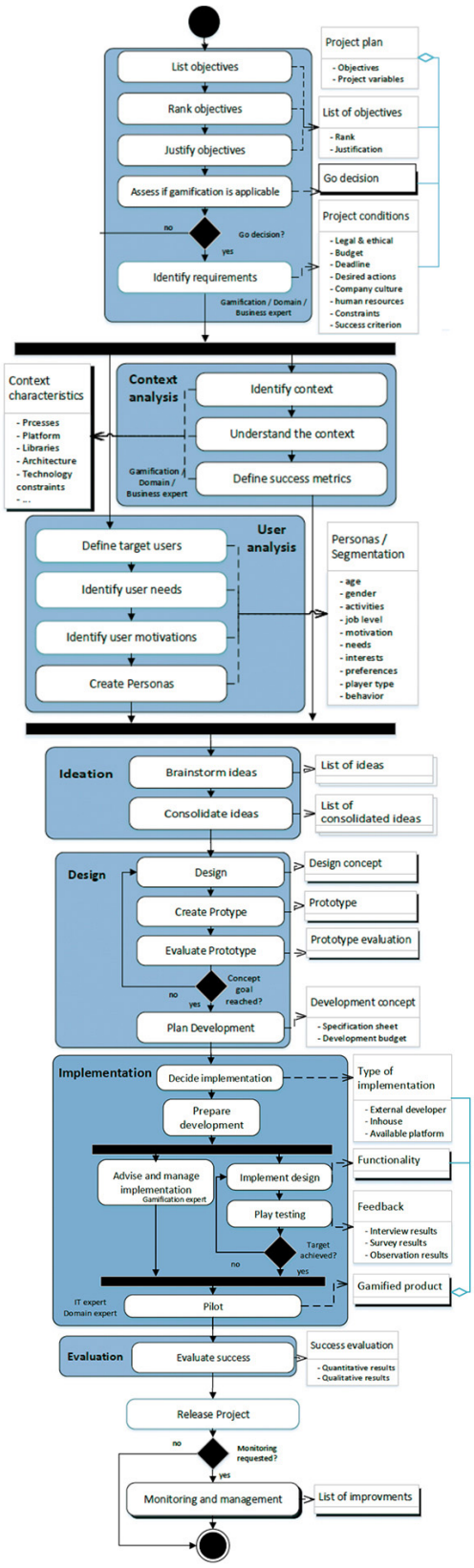
- **Game world**

  The game world is the setting. It can be functional to the story, and depending on the type of game can also be crucial. In the educational field it can be simulated virtually by using a dedicated web platform for education.

- **Immersion**

  Once the game objectives are clear and the activities are organized in an engaging way, participants can lose their sense of time and stop worrying about themselves. This type of immersion is typical of the best video games, and is the result of a wise synergy between the various components of the game: entertaining missions, eye-catching graphics, and an interesting story.

- A gamification workflow



After determining ultimately what these playful elements are, and what are the criteria for implementing them correctly, we can finally go to the design stage of our project. To work at best, a workflow model will be followed, as shown on the left (Morschenheuser et al., 2017). To make the diagram more understandable, every key step of the process will be briefly described.

1. **Project preparation**

   We clearly define the objectives of the project, classifying them according to their importance, and providing a justification for the choices made. It is therefore possible to implement the gamification in accordance with the requirements and material limits imposed on the project. These may be of various kinds, such as ethical and legal, associated with the budget or related to the project deadlines.

2. **Context and user analysis**

   We identify the context, the system where we want to apply gamification, and we study it in order to understand it: this includes the analysis of the development platform, its architecture, its technological limits, and so on. We define the conditions that must be met to determine the success of the project.

   We also outline the end-users to whom the project is addressed, trying to understand the needs and motivations that guide them. For this purpose, we use Persona models that identify user stereotypes based on features like age, gender, needs, interests, preferences, or **player type**.

   This latter feature in the gamification field is of crucial importance: therefore, we will shortly list these types in order to have a better understanding, referring to the standard nomenclature used in gamification studies.

- **Achievers**: These people like to solve the problems they face in front of them. A good way to motivate this type of player is to restrict access to some content in the game, prompting the player to unlock these content by demonstrating their abilities**.**

- **Explorers**: These people love to discover the secrets of sandbox environments, as well as to create new things. A good example of a game that draws in this type of player is Minecraft. Alternatively, in the educational field, Explorables are a great example of a game that attracts this category.

- **Socializers**: These people love to build relationships with other players. A good way to attract this type of player is to implement multiplayer mechanics that allow cooperative play.

- **Killers**: This ominously ambiguous label identifies people who love competitiveness, which can be offered by the game itself, or by competing with other players. As mentioned earlier, it is rare to implement such challenges in the educational field; however, their benefits should be considered to attract this category of players.

It should be considered that these categorizations are not mutually exclusive: it is more normal for a player to have several different types of interests. It is therefore convenient to design trying to satisfy every possible target.

3. **Ideation**

At this stage, ideas are gathered through a process known as Brainstorming. To this end, gamification experts recommend that you use the tools listed below at your own discretion.

- **Table Games and Video Games:** Trying games and discussing their mechanics can stimulate the mind, bringing new ideas.
- **Design Lenses and Design Cards**: they offer a particular perspective on the design process, directing the work in a precise direction.
- **Visualizations**: they're used to understand and show relationships between users and their behaviour in a particular environment.
- **Game Design Patterns**: quite common gaming components. They're fundamental to the development of gamification related ideas.
- **Story Cubes**: dices with different faces, used to support the creation of a story, which in turn can be a source for design ideas.
- **Canvases**: Canvases can help to structure the ideas for gamification systemically. They allow to easily communicate ideas, identify any flaws, and compare different approaches.
- **Decision Trees**: they help to make choices, and guide developers in selecting game elements.
- **Good-Practice Gamification Patterns:** Examples of recurring gamification patterns, used as the starting point for the process of design.

After conceiving a list of ideas, the developers must choose which of them can be implemented, and then be consolidated for the design phase.

4. **Design**

   At this stage the design takes shape: a design concept is developed, a prototype is created and evaluated. If the final product respects the objectives of the initial concept, the development phase proceeds; otherwise, the design cycle repeats until a satisfactory concept is made.

5. **Implementation**

   This phase is considered to be a continuation of the prototyping process. Decisions are made regarding which developers will be employed, and which environment will be used for development; therefore, under the supervision of a gamification expert, the design discussed in the previous step is implemented.

   Thereafter, the gamified product is tested, until the testers feedback reaches a positive reception rate. Upon reaching these conditions, a pilot product can be finally released.

6. **Evaluation**

   At this stage, the product's success is evaluated quantitatively and qualitatively: it is checked whether the product attracts the designated user and meets the set goals. These checks are usually performed through interviews, surveys, or comments made on players.

7. **Monitoring**

   At this stage, after the product release, we observe the released product in order to detect defects, and identify further improvements to be implemented with subsequent patches.

• Summary of the requirements for proper gamification

Based on what has been said so far, and thanks to the contributions of various experts (Morschenheuser et al., 2017), it has been possible to produce a set of recommended requirements for good gamification:

- Understanding the needs of the user, the motivation that guides him and his behaviour, as well as the context characteristics.
- Identifying the objectives of the project, and defining them clearly.
- Testing gamification design ideas as soon as possible.
- Following an iterative design process.
- In-depth knowledge of game-design mechanics and human psychology.
- Checking whether gamification is the best way to achieve the desired goals.
- Stakeholders and organizations involved must understand and support gamification.
- Focus on the needs of the user during the ideation phase.
- Defining clear reference values for the evaluation and monitoring phase in order to properly determine success of the gamification.
- Checking that the system does not have any flaws that allow cheating.
- Managing and monitoring continuously to optimize gamification design.
- Keeping in mind the ethical and legal limits during the design phase.

- Involving users in the ideation and design phase.

• Towards game development

In light of the above, it appears that **the best approaches to education through video games are the COTS approach (which usually uses existing Serious Games) and Gamification**. Since in our case the goal is to create a functional game editor meant ideally for educators, our first aim will be to design an easily adaptable game structure, and proceed subsequently with the development of the editor that will allow teachers to build the aforementioned game, keeping in mind the educators' skills and limits.

We will try to create a hybrid model between the serious game / gamified system that fits in the most universal way to the needs of the users and the educators.

Let's first outline some of the requirements set by the stakeholder: **the game in question must allow students to make assessments about their knowledge through the subterfuge of gaming**. Although I personally feel that it is more appropriate to use games for learning rather than for the assessment of students, keeping the two concepts separate, it is undoubtedly of interest and challenging to be able to make the latter an entertaining experience.

The game is meant to be released and played on a platform, named DEV, a web based e-learning system that allows personalization and adaptivity through student modeling. This means that the game mustn't require too many resources, or will otherwise become too

heavy to be handled on the end-user's systems, and it must follow particular standards.

Last of all, **the final product must follow the requirements set in all the previous sections as much as possible**.

As a side note, before further proceeding, it must be noted that the skills required to build a decent game often do not belong only to programmers, but also to artists, and possibly writers, and sound experts. As a matter of fact, Game Developers and Game Designers are two distinct figures in the gaming industry. In the case of educational videogames, it should also be considered the support of an educator.

It is important to understand that video games are a multimedia product; while the idea of the single creator with many skills is romantic (*Undertale* created by Toby Fox, or *Minecraft* created by Markus "Notch" Persson), these rare cases seems to be discretely inaccessible according to market mechanics and the enormous development times required.

In this project, my goal will be to conceptualize and develop a proper, entertaining educational game within the boundaries of my own skills.

Last but not least: an insight on Game Editor Technology

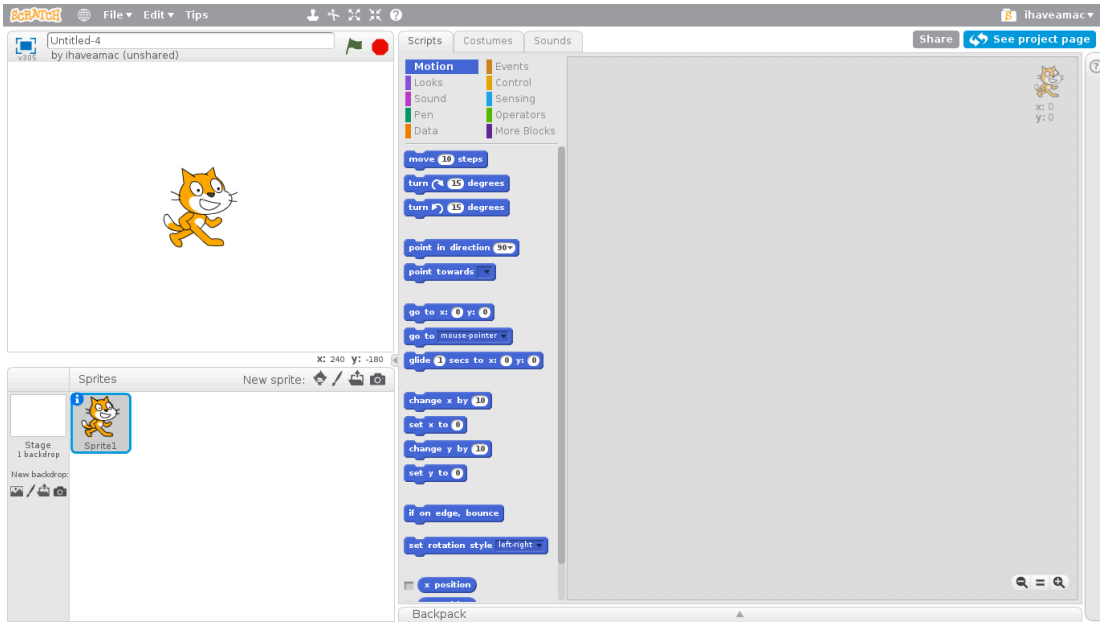
As a quite obvious matter of fact, to develop an environment meant for creating educational games not only is it necessary to know how to make a video game, but is also advisable to see how already existing editors work. With such a goal in mind, our last step before delving into the depths of software development will be to analyse such aforementioned editors. Before further proceeding, though, the reader must also acknowledge a simple concept: **almost any environment can actually be good to develop educational video games, as even a modded version of Pokemon can be educational, with proper coding. The only limit may be time, skills and most important of all, imagination**. What must be considered here is that these three quintessential ingredients may not always be present, and their lack can jeopardize a product's success.

As we've seen before in the Student-made Games section, some well-known environments for building games are *Scratch*, *GameMaker*, *Gamestar Mechanic*, *Python*, *Alice*, *Adventure Maker*, *Inkle*, *Pixel Press*, and *Tynker*. While

some of them require a discreet knowledge about programming, others do not. Pros and cons will be taken into account in the analysis of some of these environments, and requirements will be outlined to develop the easiest-to-use environment possible based upon best-practice cases.

- **Scratch**

  Scratch is a free programming environment, based upon a **graphical programming language**. It was made by Lifelong Kindergarten, a group from the MIT Media Lab, lead by Mitchel Resnick, in 2003. The language, inspired by the **theory of constructive learning** and designed for **teaching programming through simple visuals**, is suitable for students, teachers and parents, and can be used for **pedagogical and entertainment purposes**, ranging from studying mathematics to science projects, allowing the realization of simulations, visualization of experiments, animations, music, interactive art, and simple games with mere ludic purposes. Scratch provides an object-oriented approach called Sprite. The idea of this language is that children or people with no experience of programming languages can learn it through simple interactive interfaces.



While not diving deep into its mechanics, a simple look to the interface is given in this dissertation to get a proper overview.

The area on the top left is the "stage area", where the results appear and the code comes to life (animations, graphics, etc., all in small, normal or full screen sizes) while the Sprite, the graphic objects on which the code acts, are listed below.

By selecting a Sprite the user can associate the code blocks, selectable from the Script area, where they are grouped by topic. If a Sprite has associated code, this is displayed in the area on the right and the user can edit it. Any block can be tested by double clicking so the user can preview the action. Next to the Script folder, there are two more for Costumes and Sounds. An expandable bar on the right is dedicated to the tutorial.

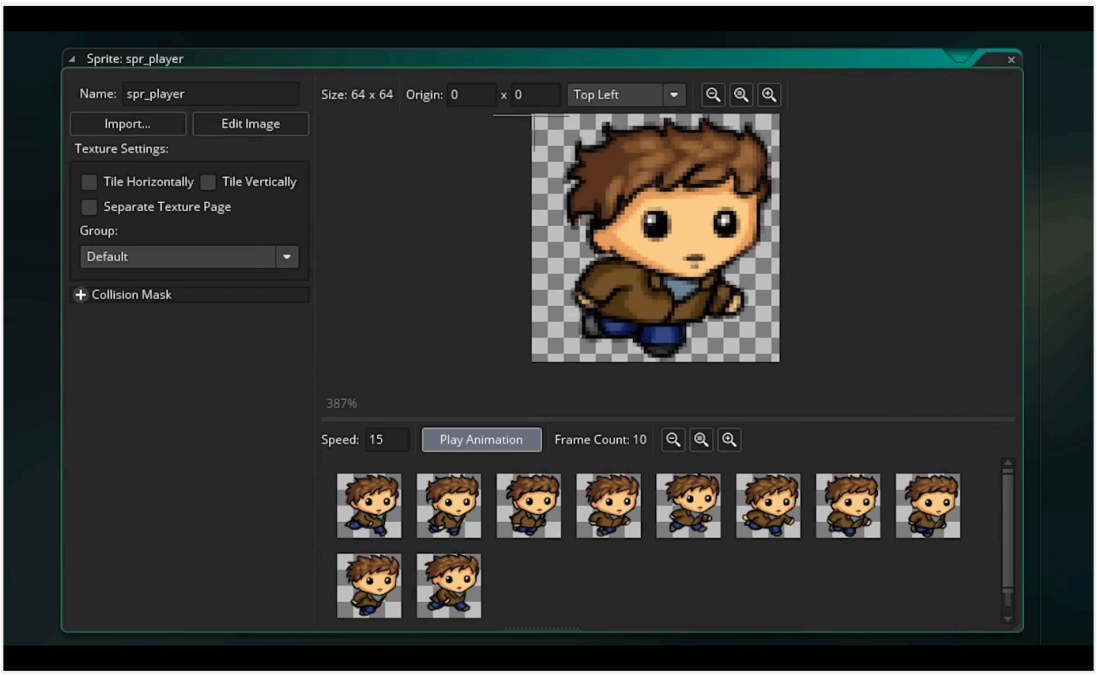
While Scratch indeed represents a particularly powerful tool, it is way far from being an immediate environment for building

games, as it requires time to be learned, whilst not assuring that the final product is a successful and interesting game that follows the requirements outlined in the previous sections. Given so, it is nonetheless a worthy tool considering its potentiality.

- **Game Maker**

  Game Maker is an IDE (integrated development environment) for video game development, originally created in 1999 by Professor Mark Overmars, and later developed by YoYo Games.

  Game Maker provides two programming methods: icon based and code based. **The first is aimed at beginners where thanks to icons to drag-and-drop with the mouse one can create games even without too much coding nor knowing any programming language**. The second method provides us with the Game Maker Language, a programming language with a syntax based on the union of Delphi, Java, Pascal, C and C ++ languages. The latter approach allows to increase the potentiality of Game Maker to levels almost identical to other, more highly professional environments. The functions of the program can be extended using the Dynamic-link library (DLL) files. It is also possible to create GEX, aka libraries written in Game Maker Language to be used exclusively in Game Maker.



Examples of well-known games made with the Game Maker tool are *Hotline Miami*, *Spelunky* and *Undertale*. Although Game Maker's potential is doubtlessly tangible, while it can surely be used to make educational games (as any other IDE), these games can't be made as easily as they're made in Scratch, nor can guarantee to be successful for the same reasons mentioned in the previous paragraph: **without proper knowledge of programming and video game mechanics, Game Maker's power can be wasted**.

❖ **Gamestar Mechanic**

Gamestar Mechanic is a game-based digital learning platform geared at nine to fourteen years old students that is **designed to teach the guiding principles of game design and systems thinking in a highly engaging and creative environment**.

Gamestar Mechanic allows students to learn about how systems work and how they can be modified or changed. Students learn to think analytically and holistically, to experiment and test theories, and to consider other people as part of the systems they create and inhabit. Through the game design process, students cultivate skills involving: system-based thinking, creative problem solving, art and aesthetics, writing and storytelling, science, technology, engineering, mathematics.

The design of the game is based on research by some of the leading academics in the field including Katie Salen (Executive Director of the Institute of Play and curriculum author for the New York City Public SchoolQuest To Learn) and James Paul Gee (author of What Video Games Have to Teach Us About Learning and Literacy).

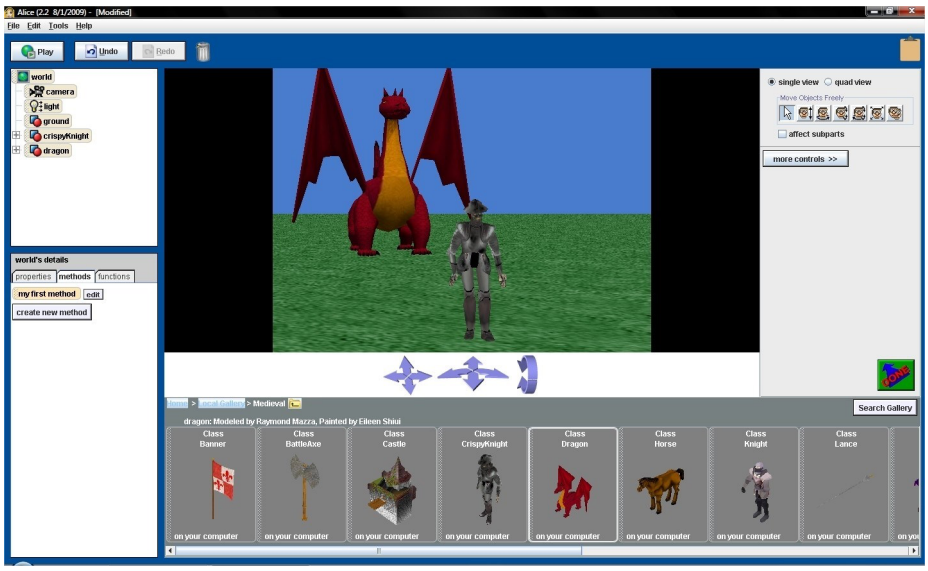
Gamestar Mechanic, whilst being especially meant to teach principles of game design, represents an interesting environment

for educational game development (Games, 2009). Unlike Scratch or GameMaker, **Gamestar Mechanic doesn't give too much freedom to the developer**, as it is not remotely meant to develop anything like the Serious Games we've seen before, but regardless, **the way it is built allows its users to create games according to hard-wired game design rules; unskilled developers have less chances of ruining their games because of their lack of experience,** as mostly everything is coded already, similarly to what happens in the drag-and-drop coding of GameMaker.

- **Alice**

Alice is a **block-based programming environment** that makes it easy to create animations, build interactive narratives, or program simple games in 3D. Unlike many of the puzzle-based coding applications Alice motivates learning through creative exploration. Alice is designed to teach logical and computational thinking skills, fundamental principles of programming and to be a first exposure to object-oriented programming.

The Alice Project provides supplemental tools and materials for teaching using Alice across a spectrum of ages and subject matter with proven benefits. Alice is used by teachers at all levels from middle schools (and sometimes even younger) to universities, in school classrooms and in after school and out of school programming, and in subjects ranging from visual arts and language arts to the fundamentals of programming and introduction to Java courses.



Research has shown that Alice has a measurable positive effect on performance and retention in computer science education (Conway, 1997).

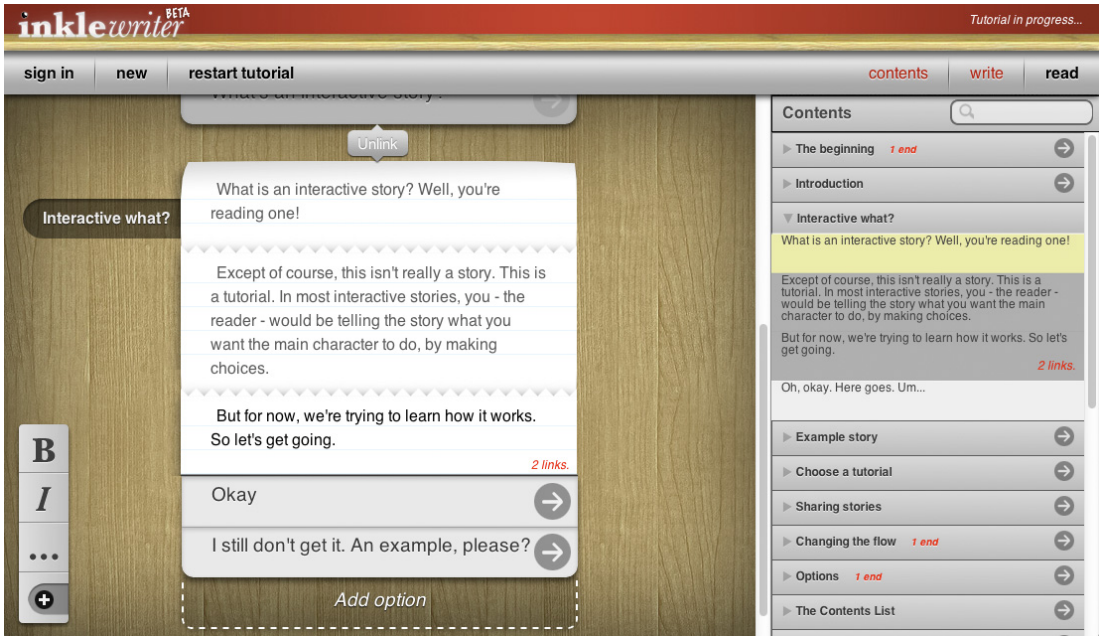
Being so similar to Scratch in the means of design and purpose, Alice doesn't give further insight on what we have discussed, yet it underlines the importance of "ready-made" chunks of code: **people with no knowledge of coding can easily take their first steps into game development if they're not facing the intricacies of real software programming**.

- **Inklewriter**

Inklewriter is a free tool designed to allow anyone to write and publish interactive stories. It's perfect for writers who want to try out interactivity, but also for teachers and students looking to mix computer skills and creative writing.

Inklewriter lets users write as they play, branching the story with choices, and then linking those branches back together again. It

keeps track of which paths have been finished, and which still need to be written.



There's no set-up, no programming, no drawing diagrams. Once written, Inklewriter allows to share the story with the world, because every story is given its own unique web-page that the user can share however he or she wants.

Inklewriter, among the examples seen until now, represents the less code-writing related approach to game-building, and its game design may indeed be considered hard-wired if not untouchable, similarly as what we've seen in Gamestar Mechanic.

Such hard-wired design though, although it limits the power of the developer within certain boundaries, **comes in handy as more users can use the editor**, and **the games made with the editor have less possibilities of being ill-designed**.

❖ **Pixel Press Games**

Pixel Press is a technology company focused on building engaging gaming experiences to develop analytical and creative skills through a simple formula, namely the "build your own video game" approach. Although Pixel Press has made more than one game, the similarity between one another makes it simpler to talk about them in a general way, as, concept-wise speaking, few changes occur between one game and the other, often most in terms of imagery and assets.

The graphical programming though, commonly present in all their games, **underlines yet again the importance of simplified programming for unskilled users. The less people have to code, the more they'll likely enjoy making a video game**. What makes

these games unique is that in some of these, level programming is made directly on paper, scanned and implemented through a special algorithm. A natural gesture such as drawing turns into complex coding in the blink of an eye, allowing users to try instantly the fruits of their imagination.

An insight that can be given upon the latter statement is that programming through natural gestures (or to be more correct, gestures related to previous experiences) **give a particular appeal to video game editors, and make human-machine interaction easier and simpler**. (Norman, 1988)

It is also worth mentioning that in a few of its titles Pixel Press exploits the previously mentioned idea of using famous brands to increase their own product's attractiveness through commercial partnerships: as a matter of fact, one of the games, "Adventure Time Game Wizard" is based upon Cartoon Network's homonymous show, while another one, "Bloxels Star Wars", clearly takes its inspiration from the famous movie series.

- ## Before further proceeding

The Explorables project, discussed in one of previous sections, offers another distinct approach towards game building meant particularly for educational purposes, which surely comes in handy in this dissertation. But being the project pointed towards different goals from ours, it won't be discussed here, as our aim is to study game editors for people who don't know how to code, and are probably not willing to know how to code, while the purpose of the Explorables project is to teach educators to make their games according to the

principles of good coding and game design. Further information can be found on http://explorabl.es/tutorials/.

• Conclusions on Game Editor Technology

Based upon what we've seen so far, a few things can indeed be outlined in terms of **editor complexity**, **user coding knowledge**, **user game design knowledge**, **user creative freedom**, and **final products effectiveness**, defined hereby.

- **Editor Complexity**: this term, in this particular context, is used to define how much an editor's mechanism requires time to be learned, and mastered. This is considered in the perspective of someone who doesn't know anything about programming, or a coding amateur, and is calculated in terms of time to learn and feedback given by the users to self-assess their acquired expertise. The more the complexity of an editor grows, the more time it will require to be learned, and the less will it be appealing to people who don't know how to code. Regardless, a complex editor has its pros, as it usually hides more potentiality within its structure. A complex game editor is more keen to be the right tool to create a successful serious game, compared to a simpler editor. Nonetheless, if the development happens in an uncontrolled environment, the final product may be unsatisfying. A good editor should empower the user without allowing him or her to make too many mistakes in the development phase.

- **User Coding Knowledge**: with this concept we describe how much a user knows about programming. Although knowledge can't be properly quantified, this can be relatively defined through self-assessed evaluations by users, or tests about basic knowledge of computer science. The more a user knows about coding, the less he'll have to adapt to most of the editors. More programming knowledge often means games with more functionalities, but a programmer that lacks knowledge about game design may accidentally make mistakes during the design phase of the game, risking to create an ill-designed product. To avoid this kind of problem, an editor could wire game-designing choices within its structure.

- **User Game Design Knowledge**: like the aforementioned kind of knowledge, this kind of notion defines what a user knows about game designing. The more a user knows, the more likely he's going to make a successful product. But if the game designer lacks knowledge of good programming, he may not be able to make a good game, or may create an ill-coded product. To avoid this kind of situations, a good editor should allow users to create games without heavy knowledge of programming.

- **User Creative Freedom**: this term defines how creative can a user be, in terms of programming and design. Absolute freedom is what is given by any classical programming language such as C, Java, etc., while restricted freedom likely describes those environments where everything is limited by strict boundaries. Absolute freedom guarantees unimaginable potentiality, but can't guarantee that it won't get wasted. On

the other hand, limiting the user's freedom may not allow the users to create whatever they want, but avoids user-made mistakes in terms of coding or game design.

- **Final Product Effectiveness**: with this concept we describe an editor's product's effectiveness in function of the requirements for a game's success (such as positive feedback from the end-users, eventual incomes), requirements for proper gamification, and cognitive psychology's guidelines. To guarantee effectiveness, a game must at least be well-designed and well coded. If the editor's goal is to allow to create successful games, it must sacrifice the user's creative freedom, to avoid errors related to coding and designing.

We can finally establish all the requirements for our editor. As the end-users are ideally teachers who want to assess students, and may not have time to learn how to work with a new environment, the editor must be **simple**. As the end-users may lack knowledge about programming, the editor must point towards as **less coding as possible**. The teachers may lack knowledge about game design, thus **design choices must be hard-wired into the final product, so that the game follows the requirements of a successful video game**. A discreet amount of **creative freedom has to be sacrificed** in order to accomplish so, but nonetheless it will be granted as much as required for the teachers' objectives.

QuBE

The Quiz Battle Editor

- ## Game Concept

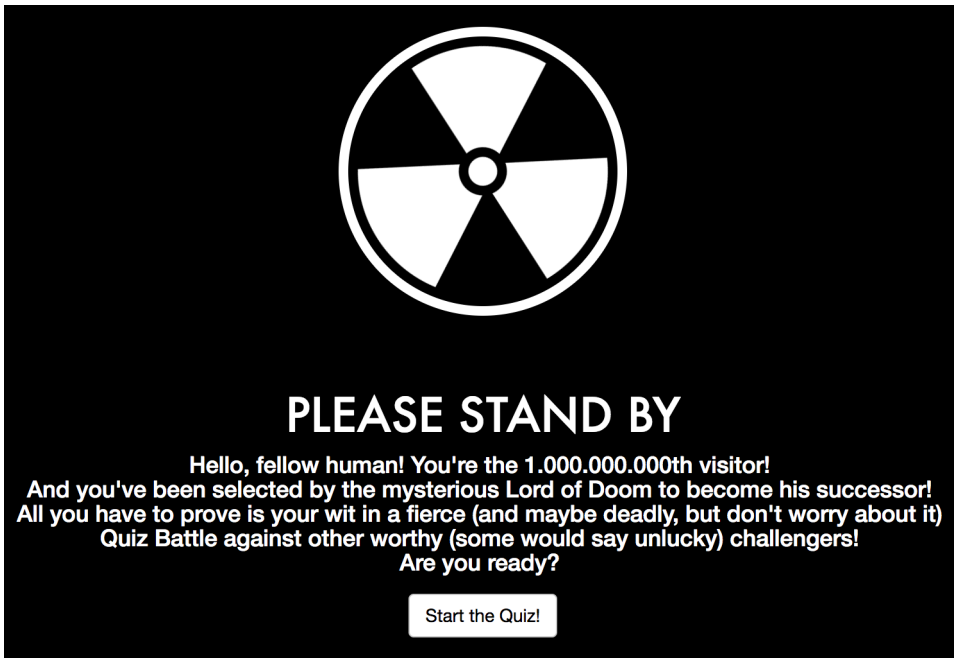
The concept for the game itself is quite simple, and takes heavy inspiration from games such as Buzz!, a quiz game where the player, facing other players, has to answer as fast and correctly as he can in order to win. In the games made with our editor, the Quiz Battles, the player faces a singular opponent, an AI with different degrees of difficulty, expressed in terms of probability to answer correctly, and a time slot within which the enemy player can answer randomly given the total number of seconds required to answer (established by whoever made the game). The higher the difficulty, the faster will the enemy answer, and the more chances will it have to answer correctly.

- ## Game design choices and mechanics

As stated in a previous section, as these games have to run in an internet browser within a web platform named DEV, Quiz Battles have been made in a minimalistic style, in order to avoid requiring heavy resources on the client's machine. Nonetheless, every nook and cranny has been calculated precisely to fit every notion we've acquired until now. The first step that had been taken was to conceive

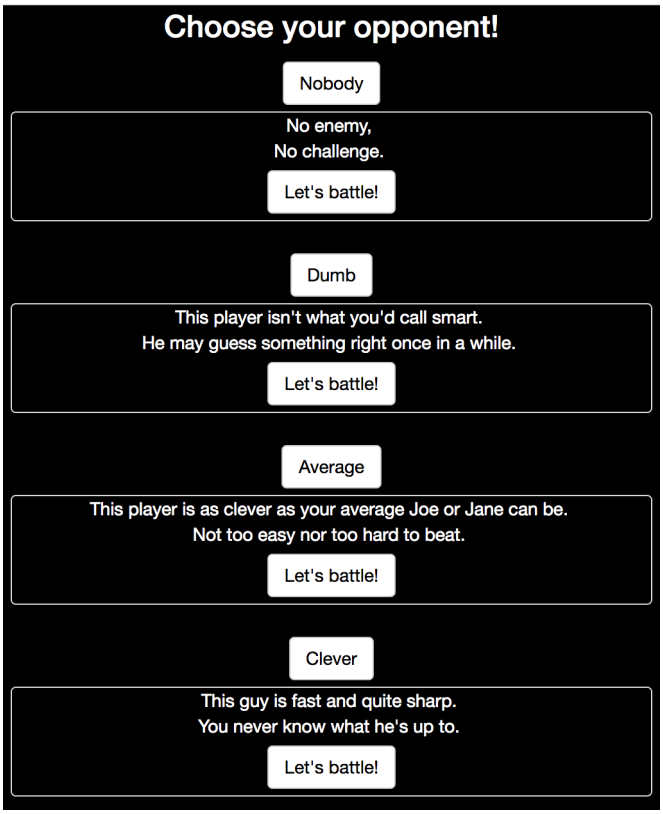
something that allowed to assess students, such as questionnaires. The next step has been pure gamification along the lines described in the game concept, following tests to see the effectiveness of the game. Having received a positive feedback, the game has been developed following the requirements.

- **Plot**: although simple, Quiz Battles have a sort-of-plot that's meant to work as an eye-catcher: it's supposed to be humorous, exaggerated, but entertaining in its own small way, in the belief that even the slightest emotion can make the experience a bit more pleasant and thus, memorable. The fictional danger has been created to appease those who seek risk (without actually there being one, obviously).



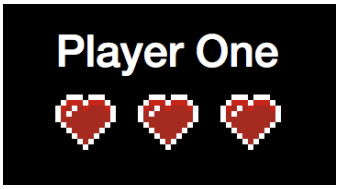
- **Enemy AIs**: these have been made to give a sense of competitiveness. This feature implements a discreet amount of luck in the game (as the probability of correct answering by the

AI is calculated randomly within certain bounds), and it's pointed towards Achievers (to give a sense of satisfaction by beating a hard-to-beat AI), and Killers (as the AI also incarnates a fictional player to compete against). To avoid creating an excessively hard challenge though, different levels of difficulty have been added, so that anyone can play according to its own pace.

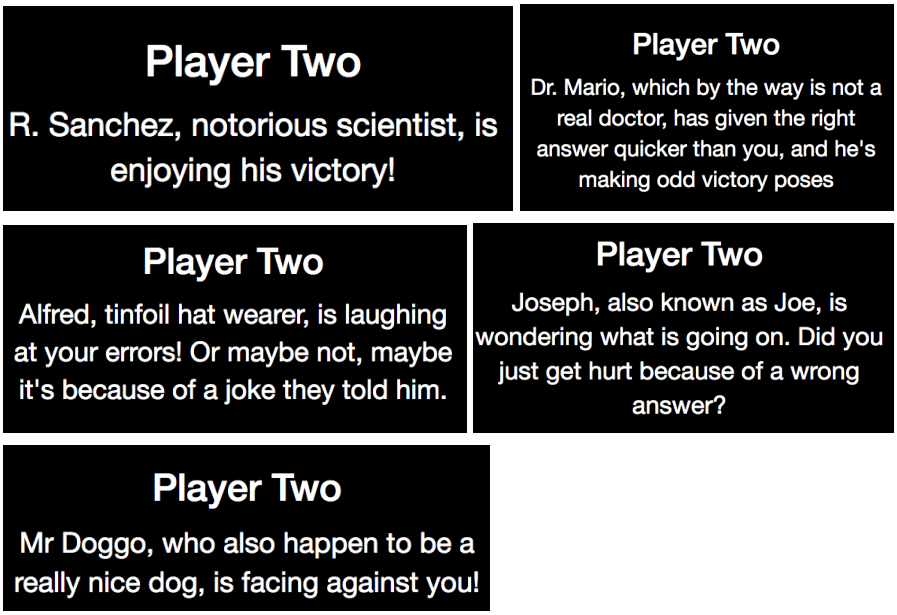


- **User Freedom**: beyond being able to select the difficulty, a user can also select the number of questions he or she wants to face. This grants power to the player, allowing him to choose the kind of challenge he wants. It's also a feature meant to appease Explorers. The player creates his own experience.

- **Health Bar**: the hearts displayed in the Player HUD (head-up display) are a gamified way to express how many errors can the player make within the boundaries of reason: if the player can't answer correctly to at least half the questions, he may not be ready to take the test.



- **Enemy Feedback**: Enemy AIs give feedbacks according to the player's behaviour. If the player answers correctly, skips the question, doesn't answer, gets the question wrong or loses all his lives, the AI is going to act in a particular manner. Every enemy has a unique comedic behaviour, following a style that resembles a formula applied in cult games such as Nintendo's Earthbound.

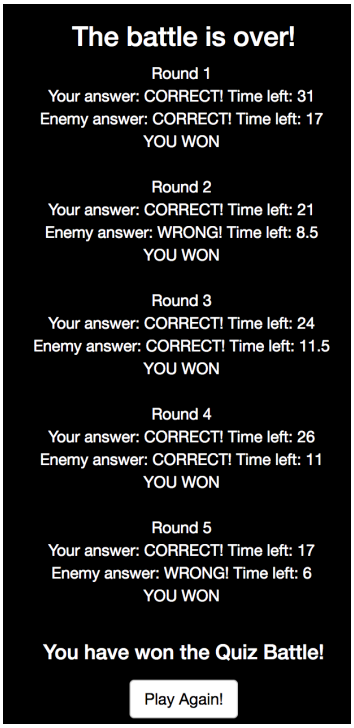


- **Timer**: the timer is a necessary mechanism, although not dearly beloved by those who have to stand against it. Were time-to-answer a calculable asset, it wouldn't be required from editor users, as it can be bothersome to guess a reasonable amount of time within which a player would have to answer to a particular question. But as the AIs have to answer in a reasonable manner, without either being too slow or too fast, it is ultimately considered peculiar to know how much time is it needed to answer. Showing the timer is important as well, as it guarantees in-game fairness, as the player must know within which limit he or she has to answer.

- **Game Over**: the old-school screen that pops up when you fail. Losing all lives (basically getting at least half the questions wrong) triggers the Game Over screen, prompting the user to consider if his or her knowledge is enough to try the test.

Although the game isn't meant to be excessively punitive, this kind of mechanism is quite mainstream, as it shows up in many modern game titles. It's part of the fictional risk in a controlled environment.

- **Score**: the score is the ultimate feedback, that allows the student to see his or her own performance once the game is over. While the game has its own way of calculating the score according to the rules of the game, the actual assessment of the student (academically speaking) is calculated separately with another function. Regardless of what the AI has done, what matters is that the player has answered correctly.



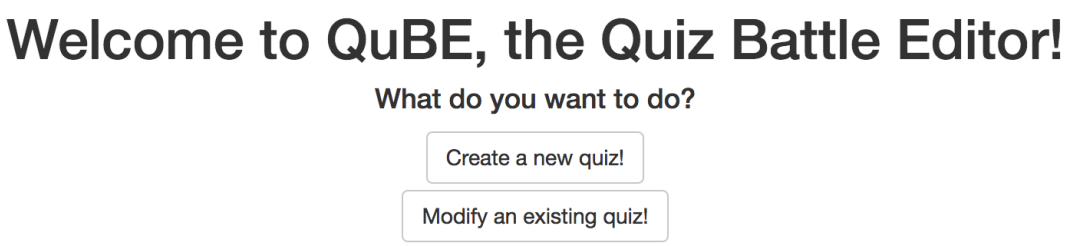
- # Editor design choices

As it has been said previously, the editor has been built in a really minimalistic way, in order to allow as less client-side coding and game designing as possible. Since most of the game mechanics can be hard-coded without fear of tampering by ingenuous users, the editor has been built to resemble a classical questionnaire spreadsheet. This way, the educator using QuBE has the feeling of doing something "natural", already related to his or her previous experiences, since it is normal for a teacher to write tests for students.

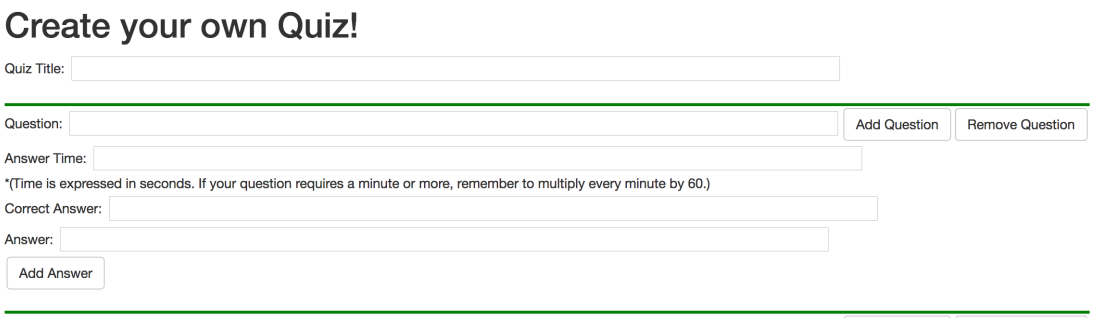
Nonetheless, there is one feature that can be dangerous if mishandled, and ruin the game: the time required to answer to each question. As it has been explained before, issues regarding this feature can't be solved without a bit of hope in the judgement of the educator using this tool.

The editor, pretty simple in its structure, will be described hereby.

- **Editor Main Page**: the first page of the editor, quite simple in its structure. The user can choose either to create a Quiz Battle from scratch, or modify an existing quiz created previously by the user.

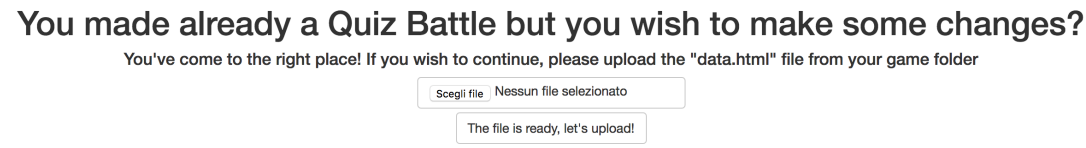


- **Creator Interface**: if the user clicks on the first option from the previous page, he'll find him or herself in page structured as a dynamic HTML5 form with a list of empty input text areas. At the top of the form goes the Quiz Title, which can be for example a subject studied in a course. Questions, structured as elements (Document-Object-Model-wise speaking), can be added or removed by clicking on the buttons to the right of each question, in a pretty intuitive manner. The answers to each question, which recursively follow the same concept, can be dynamically added to the form with the same principle. Following discussions with the stakeholder, the number of questions for a Quiz Battle has been set to a minimum of 20. These questions represent a pool from which the game draws randomly the questions the player sees during a Quiz Battle, thus making the game a bit more random and complex. The more the questions are, the more is the game going to guarantee replayability.

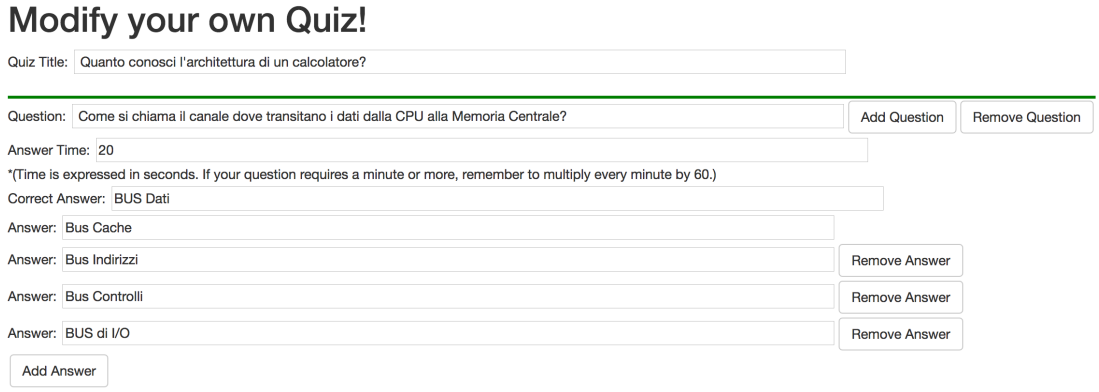


- **Modifier Interface**: If a user wishes to edit a Quiz Battle instance he or she has created previously, it can easily be done using the "data.html" file from the game folder that is generated using the creator interface. "data.html" is basically a Base64 encrypted file that contains the questions in a particular

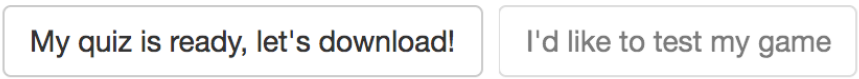
format. The encrypting has been done in order to avoid cheating.



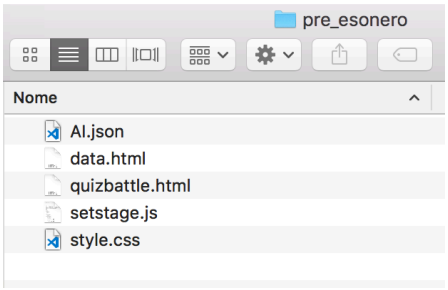
Once the file has been uploaded, the file is parsed by the editor which automatically generates a web page with a pre-compiled form that resembles the creator interface.



- **Download and Test**: last but not least, at the bottom of the page are two distinct submit buttons. The first one allows to download the quiz (if all the required fields have been properly filled).



Once the game has been downloaded, the “Test” function is unlocked, allowing the user to try out the game he or she has just created.



As can be easily seen, the editor is incredibly simple to use, and the game itself is almost lightweight. This guarantees basically no game loading time, no time to upload the game to the site where it's supposed to run, and heavy modifiability.

• Discarded Ideas

Some programming choices, as well as game-related mechanics, have been ultimately discarded for one reason or another, among which was the misunderstanding of how DEV, the container website, is structured or how it works. Nonetheless these ideas are briefly discussed here, with proper justification for the choices that weren't made.

- **Quiz Data Base**: the first version of the Quiz Battle was conceived as a stand-alone game on its own site, without requiring external container sites. Users could have chosen the Quiz they wanted to play from a list of Quiz Battles. This list, updated through a separate web portal, is based upon MySQL. This "alpha" version is still existing and available for show, or future use.

- **Default Music and Images**: although they could've been easily implemented (and have been tried as well), these assets have

been discarded as the web platform within which the game has to be put in would have generated a useless quantity of duplicated assets, since the games created with the editor wouldn't have had a shared assets folder. The sole idea of having so many redundant files felt discouraging, ultimately pushing towards discarding the use of assets.

- **Items**: this idea has never been implemented, as it was too distracting (more info in the Attention section on Cognitive Psychology). Ideally, it was about adding "Mario Kart" styled items in the game. These were meant to be used both by the player and the enemy AI, with their effects being, for example, an ink spot on screen for a few seconds, time freezing, an extra heart, or anything reasonable that could've been imagined. But, as it has been said, it hasn't been implemented as ultimately it lead to players being too distracted and unable to concentrate, especially on complex and tricky questions. Being there already sort-of distracting elements such as the timer, the player lives and the enemy behaviour display in the HUD, items have been discarded.

- **High score**: the idea takes heavy inspiration from old-school arcade cabin games. Players, at first, were meant to be able to register their game score, along with a nickname, as an incentive to play again and try to get a higher score, by maybe facing a tougher opponent. This could've been possible it the game ran in PHP. Due to DEV's nature though, PHP files can't be allowed to run for security reasons, thus making this idea available for use only on the stand-alone version.

- # Game feedback

Game testing made with the alpha version of the Quiz Battles allowed the gathering of a few interesting opinions by the users, a small pool of Computer Science students, with their ages ranging from 21 to 25 years old. Quiz data has been extracted from a Computer Engineering course material, by courtesy of a teacher, to simulate an accurate use case of the product.

First of all, it has been noticed that if the time-to-answer has been wrongly calculated by the game creator, a player may not be able to waste any time in looking at the enemy in-game messages, needing some extra-time, which has subsequently been added by default; if timing is totally ill-calculated though, the player may be unable to answer in time anyway, thus being led to frustration.

As the game doesn't relatively differ all that much from a classical timed Quiz, a serious person who doesn't have time (or feel like) playing games may play at its own pace against the non-existent-AI, basically an AI that is rigged to always answer late and not correctly. A player who isn't keen on having a laugh may as well ignore any of the messages displayed in the Player 2 section, and keep on enjoying the Quizzes as if they were boring school tests. The game works both ways, regardless of the player interest in having fun. The player shapes his own experience, and the game can go only so far in being appealing to people who do not like to enjoy things. This odd fun-hating behaviour by end-users (also seen in the studies on the educational use of Making History) has also been considered in the development phase, thus leading to the implementation of a "No enemy" option.

Comedic in-game messages have shown positive feedback by users who took time to see what was going on, when the answering time hadn't been miscalculated. Thanks to the amount of possible behavioural pattern that can be displayed by all four kinds of AI and the ever-changing questions (within the bounds of the question pool from which these are drawn), the game can also guarantee a discreet replayability.

• Editor feedback

The editor has been tested as well, although by an intermediary person in contact with an educator. While it was found indeed easy to use, writing a minimum of twenty-five questions was felt as a tiresome activity; the minimum has thus been decreased to twenty.

Calculating the time to answer has also been found as a complex activity, especially in the case of questions that required logical thinking rather than the use of memorized notions, which require less time. Nonetheless, as the platform upon which the game is implemented allows other users to download and mod other users' games, this problem of time may be solved by an active community of users and amateurish "game developers", people who may correct timer-related issues through feedbacks, or game importation and modification.

Integration with DEV, the system within which the game is supposed to run, has been successfully tried out as well, although it has been found less simple than creating the game. Since DEV integration has been found relatively difficult, the editor has been modified in order

to allow to make games that may run everywhere, whether they are on the platform upon which they were meant to be… or not.

The dissertation is over, have a nice day.

References

Atkinson, R.C., Shiffrin, R.M.: The control of short-term memory. Scientific Am. 225, 82-90 (1971)
Baddeley, A.D.: Working Memory. Oxford University Press, New York (1986)
Case, N.: Explorable Explanations, http://blog.ncase.me/explorable-explanations/
Chen, J.: Flow in Games (and Everything Else). Commun ACM. 50, 31-34 (2007)
Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Perennial (1990)
Deci, E.L., Ryan, R.M.: Intrinsic motivation and self-determination in human behavior. Plenum (1985)
Desurvire, H., Caplan, M., Toth, J.A.: Using Heuristics to Evaluate the Playability of Games. Extended Abstracts CHI '04. ACM, New York, NY, 1509-1512 (2004)
Ebbinghaus, H.: Über das Gedchtnis. Untersuchungen zur experimentellen Psychologie. Duncker & Humblot, Leipzig (1885)
Fullerton, T.: Game design workshop: A playcentric approach to creating innovative games. CRC Press (2014)
Henry Jenkins, *Cultura convergente,* Milano, Apogeo, 2007, ISBN 978-88-503-2629-7.
Hodent, C.: Developing UX Practices at Epic Games. Presented at the 2014 Game Developers Conference Europe, Cologne, Germany. Retrieved from http://www.gdcvault.com/play/1020934/Developing-UX-Practices-at-Epic(2014a)
Hodent, C.: Toward a Playful and Usable Education. In Blumberg FC (Ed.) *Learning by playing: Video Gaming in Education*. Oxford University Press, New York (2014b)
Hodent, C.: 5 misconceptions about UX (User Experience) in video games. Gamasutra (2015)
I. Bogost, "**Persuasive Games: Exploitationware**," *Gamasutra: The Art & Business of Making Games,* May 3, 2011.
Isbister K., Schaffer, N. (Eds.): Game Usability. Elsevier, Burlington (2008)
Ivan A. Games: 21st century language and literacy in gamestar mechanic: middle school students' appropriation through play of the discourse of computer game designers
Johnson, J.: Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines. Elsevier, Burlington (2010)
Kahneman, D.: Attention and effort. Prentice Hall, Englewood Cliffs, NJ (1973)

Kawashima, Ryuta, *Train Your Brain: 60 Days to a Better Brain* (London, 2006)
Laitinen, S.: Usability and Playability Expert Evaluation. In Ibister K, Schaffer N (Eds.) Game Usability. Elsevier, Burlington (2008)
Lesgold, A.M.: The nature and methods of learning by doing. Am Psychol. 56, 964-973 (2001)
Mack, A., Rock, I.: Inattentional blindness. MIT Press, Cambridge, MA (1998)
Matthew J. Conway: **Alice: Easy-to-Learn 3D Scripting for Novices (PDF)**
Morschenheuser, B.; Werder, K.; Hamari, J.; Abe, J.: How to gamify? A method for designing gamification (2017)
Mott, Tony (2010). *1001 Video Games You Must Play Before You Die*. Universe Publishing. p. 22. ISBN 978-0-7893-2090-2.
Nielsen, J.: Heuristic evaluation. In: Nielsen J, Molich RL. (Eds.) Usability Inspection Methods. John Wiley & Sons, New York (1994)
Norman, D.A.: The Design of Everyday Things. Doubleday, New York (1988).
Norman, D.A.: Emotional Design: Why We Love (or Hate) Everyday Things. Basic Books (2005)
Norman, D.A., Miller, J., Henderson, A.: What You See, Some of What's in the Future, And How We Go About Doing It: HI at Apple Computer. Proceedings of CHI. Denver, Colorado (1995)
Nouchi, Rui; Taki, Yasuyuki; Takeuchi, Hikaru; Hashizume, Hiroshi; Akitsuki, Yuko; Shigemune, Yayoi; Sekiguchi, Atsushi; Kotozaki, Yuka; Tsukiura, Takashi; Yomogida, Yukihito; Kawashima, Ryuta (2012). "Brain Training Game Improves Executive Functions and Processing Speed in the Elderly: A Randomized Controlled Trial". *Plos One*. Institute of Development, Aging and Cancer, Tohoku University.
pp. e29676. doi:10.1371/journal.pone.0029676. Retrieved 8 February 2013.
Novak D.: Handbook of Research on Holistic Perspectives in Gamification for Clinical Practices (2015)
Przybylski, A.K., Rigby, C.S., Ryan, R.M.: A Motivational Model of Video Game Engagement. Rev Gen Psychol. 14, 154-166 (2010)
Schell, J.: The Art of Game Design. Elsevier/Morgan Kaufmann, Amsterdam (2008)
Simons, D.J., Chabris, C.F.: Gorillas in our midst: Sustained inattentional blindness for dynamic events. Perception. 28, 1059-1074 (1999)
Sheldon, L.: The Multiplayer Classroom: Designing Coursework as a Game [Hardcover], 1st edn., p. 304. Cengage Learning PTR (2011)
Stroop J.R., (1935) *Studies of interference in serial verbal reactions.* Journal of Experimental Psychology, 18, 643-662.
Sillaots, M.: Gamification of Higher Education by the Example of Course of Research Methods, Tallinn University (2014)
Sweetser, P., Wyeth, P.: GameFlow: A Model for Evaluating Player Enjoyment in Games. ACM Computers in Entertainment. 3, 1-24 (2005)

Sweller, J.: Cognitive load theory, learning difficulty, and instructional design. Learning and Instr. 4, 295-312 (1994)
Swink, S.: Game Feel: A Game Designer's Guide to Virtual Sensation. Morgan Kaufmann Publishers (2009)
Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless… Educase Review, 41,2, 1-16.
Watson, William R.; Mong, Christopher J.; Harris, Constance A. (2010)：A case study of the in-class use of a video game for teaching high school history
Wertheimer, M.: Untersuchungen zur Lehre der Gestalt II, Psychol Forsch. 4, 301-350 (1923) Translation published as Laws of Organization in Perceptual Forms, In Ellis WA Source Book of Gestalt Psychology, 71–88. Routledge and Kegan Paul, London (1938)