Designing Declarative Knowledge Training Games: An Analysis Framework based on the Roguelite Genre

Bérénice Lemoine[0000−0002−7608−3223], Pierre Laforcade[0000−0001−8498−2731], and Sébastien George[0000−0003−0812−0712]

LIUM Computer Science Laboratory, Le Mans Université, Laval, France
{berenice.lemoine,pierre.laforcade,sebastien.george}@univ-lemans.fr

Abstract. Designing learning games for the retention of declarative knowledge is a way to provide learners with a large variety of adapted training situations. Such training situations can be considered as game activities built upon questioned facts. Learners must then face various game situations wherein interactive elements and rules are a means to read and answer specific questions about these facts. This chapter is an extended version of [18]. We propose Roguelite as a relevant game genre for declarative knowledge training. Indeed, its core design principles tackle the needs of variety and challenging training situations. Additionally, we propose an analysis framework to help teachers and game developers in identifying the key elements to design training games. This framework includes a set of questions to consider during the preliminary design of any training game for declarative knowledge. We identified and used this proposal in a specific research context about the training of multiplication tables. Following an iterative and prototype-centered approach, we illustrate two iterations about applying the analysis framework to guide the design and development of playable prototypes.

Keywords: Declarative Knowledge · Serious Game · Analysis · Training.

1 Introduction

Declarative knowledge is part of the knowledge necessary to perform a task. Anderson and Lebiere [2] define it as knowledge of "things we are aware we know and can usually describe to others". It includes factual information, such as multiplication tables, historical dates or geographical information. Repetition is necessary to encourage the memorization, generalization, and retention of declarative knowledge [15,25].

Retrieval practice, a concept in cognitive psychology, suggests that the act of recalling and retrieving concepts or facts enhances their long-term retention. Retrieval practice includes low-stakes and no-stakes writing prompts, brief quizzes, flashcards, etc. Research has demonstrated that this learning strategy can significantly improve long-term retention [5,25]. However, it is important to avoid making repetitive or redundant serious games, as well as those that present an imbalanced challenge relative to the players' skills, as these can lead to boredom [29]. To reduce this feeling of repetition, serious games focused on declarative knowledge should offer a wide range of tailored training activities.

Many existing serious games designed to train multiplication tables (an example of declarative knowledge) simply present questions for the players to answer, often accompanied by game mechanics such as time pressure, rewards, scores, and currencies. While there are a few exceptions that introduce advanced interactions and gameplay elements, such as platform-game mechanics where players control an avatar that must move and jump to make choices, these gameplay features can sometimes conflict with the training objectives, leading to failures despite correct answers. Overall, training games often lack engaging, long-term activities. Insufficient gameplay can quickly bore students and reduce their motivation, frequency, and duration of training sessions. It therefore seems important to allocate as much importance to gameplay as to educational content when designing learning game activities [21].

In this article, we propose to highlight the Rogue*lite* genre as a potential solution to declarative knowledge training. This genre is built upon several game design principles for which we discuss their compatibility with the requirements for effective and tailored training of declarative knowledge. Our proposal introduces an analysis framework that guides the design of Roguelite-oriented training games. This framework includes practical steps to be followed in each iteration of a prototyping-based design approach. The objective is to address design needs from both training and game dimensions, covering aspects such as technology (i.e., information required for the game engine and generation algorithm), game mechanics (i.e., how the core mechanics of the Roguelite genre operate), and game structure (i.e., the rules of the game). This framework is not confined to a particular declarative knowledge domain, nor is it designed exclusively for a specific target audience of learners. Nevertheless, to illustrate our proposal, we apply this analysis framework and discuss its implementation during two design iterations within the AdapTABLES project, a specific research context.

The structure of this chapter is as follows: Section 2 provides an introduction to our research context, including the AdapTABLES project and its focus on declarative knowledge related to multiplication tables. Additionally, section 3 introduces the Roguelite genre in video games and explores the suitability of Roguelites for designing training games. Following a concise overview of the current state-of-the-art in Section 4, section 5 presents a two-dimensional analysis framework, encompassing both gaming and training dimensions. This framework has been applied twice within our project's context. Finally, Section 6 presents the two applications of the framework and discusses the feedback gathered from evaluating a prototype aligned with the initial analysis.

2 Research Context: the AdapTABLES Project

The project aims to design and develop a serious game dedicated to the individual training of multiplication tables, targeting students ranging from grades 2 to 6.

From a teacher perspective, the training game to design will be adapted, prior to its use, to reflect how teachers consider the training: for example progress, difficulty, source facts to consider, etc. This training *structure* can be set up for the entire classroom's students, for a group, or for individuals having specific needs.

From a student perspective, the training game will follow the learners' progress, proposing facts according to their previous training sessions and results. From a player perspective, the training game will offer game levels that take into account their preferences. From a game perspective, a same training task should be tackled through different gameplays with different game elements. Finally, at runtime, the training game will have to generate varied training game activities, adapted, both in terms of gameplay and educational content, to the teachers and learners-players perspectives.

We followed an iterative co-design and prototyping approach, involving teachers, didacticians of mathematics, and game experts during the design and evaluation phases. At first, two initial steps were necessary: 1) specifying the knowledge to be trained, and 2) choosing a game genre that suits the training of declarative knowledge. These contextual elements are necessary to start designing at a high level the main key concepts and rules of the training game.

2.1 Declarative Knowledge Training

An exploratory research [17] has been conducted with the help of a user group composed of teachers and mathematics experts. The objective was to specify the adaptations to take into account when considering the training of multiplication tables from a teacher perspective: what to consider (i.e., source and targets of the adaptations) and how to realize these adaptations. The main two results are: a model of the training organization into training paths, and the specification of five detailed training tasks.

A training path, see Figure 1, is represented by a set of objectives ordered by prerequisite relationships. An objective (e.g., "*Work on the table 2*") is broken down into progressive levels of difficulty. Each level is itself broken down into training tasks (e.g., "*Level 1: Completion 1 with search for the result, Identification by choice of the correct facts*"). A task is defined by its type and parameters. The levels' achievements are considered from both a percent of encountered facts and a percent of achievement to reach.

2.2 Different Tasks Objectives and Parameters

The teachers have identified different questions that served as a basis for identifying five distinct tasks:

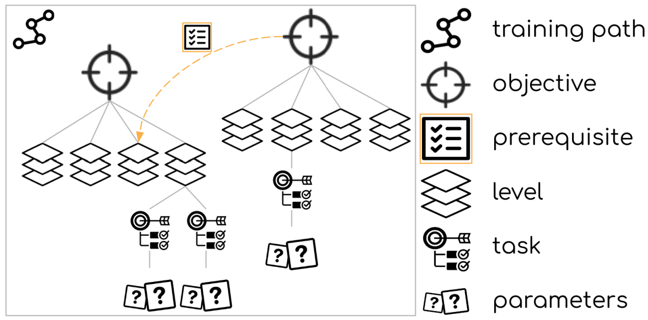


Fig. 1: Knowledge Structure [18]

- Completion 1: i.e., complete an incomplete fact that has one missing element (e.g., $3\times? = 15$, $15 =? \times 5$, $3 \times 5 = ?$);
- Completion 2: i.e., complete an incomplete fact that has two missing elements (e.g., $?\times? = 15$ with a set of given choices $[3, 6, 5, 10]$, $? \times 5 = ?$ or $3\times? = ?$ also with sets of given choices);
- Reconstruction: i.e., replace, in the correct order, all important elements of a fact (e.g., $?\times? = ?$ with a set of given choices $[3, 6, 5, 10, 15]$);
- Identification: i.e., identify the correctness or incorrectness of one or several facts (e.g., $3 \times 5 = 15$, true or false?);
- Membership Identification: i.e., identify the elements that share or do not share a given property (e.g., $[3, 5, 9, 14, 21]$ which are results of table 3?).

The training tasks are defined based on teachers' opinions and preferences, allowing them to choose and configure these tasks for each {objective, level} combination. For instance, Table 1 presents the parameters for the *Completion 1* task, including the targeted multiplication tables, the position of the multiplicand and the result, the range of multipliers (minimum and maximum), the elements to search for, the order of the questions, the response modality, and the maximum response time.

Since we are using a prototyping design approach, the design process is conducted incrementally. Consequently, each new iteration incorporates additional information compared to the previous one. In this article, we focus on presenting the initial two iterations of our design, which do not yet encompass the complete knowledge structuring as outlined.

The primary prototypes employ a parameterization approach, wherein teachers are required to provide the parameter values (as indicated in Table 1) for each learner within the game.

Table 1: Examples of parameters for the *Completion 1* training task [18]

Adaptable element	Possible Values	Some Examples
Targeted Table(s)	From 1 to 12	
Multiplicand Position	Left $\vee$ Right	$1 \times 2, 1 \times 3, 1 \times 4..$ $\vee$ $2 \times 1, 3 \times 1, 4 \times 1..$
Result Position	Left $\vee$ Right	$1 \times 2 = 2 \vee 2 = 1 \times 2$
Multiplier Interval	Integer Min/Max in $[1, 12]$	$[1, 5] \vee [5, 10] \vee [1, 12]$
Element to search	Result $\vee$ Multiplicand $\vee$ Operand	$1 \times ? = 2 \vee ? \times 2 = 2 \vee$ $1 \times 2 = ?$
Questions Order	Ascending $\oplus$ Descending $\oplus$ Random	
Response Modality	Choice between propositions $\vee$ Input	
Max Response Time	Time in seconds	

3 A Relevant Game Genre for Training Purposes

3.1 The Rogue*lite* Genre

Over the past decade, the Rogue*like* and Rogue*lite* video game genres have experienced a significant surge in popularity. These genres originated from the groundbreaking game that introduced this style of gameplay, Rogue [31]. Rogue was a turn-based dungeon crawler where players had to navigate through levels of a dungeon, battling enemies, collecting items, and progressing further. The Berlin Interpretation [10] defined eight key factors that characterize Rogue*likes*, including:

- Random generation: Each playthrough is unique due to the random or semi-random generation of levels, enemy placements, item locations, and environmental conditions. This randomness adds an element of surprise and unpredictability, requiring players to adapt their strategies on the fly rather than relying on memorized patterns. Procedural generation is typically employed to avoid unwinnable situations.
- Permanent death: When the player's avatar dies, all progress is lost, and they must start the game from the beginning. There is no carryover of progress between runs.

While many Roguelike games adhere to these eight key aspects, some games deviate from certain elements. As a result, these games have been referred to as "roguelike-like" or "rogue*lite*".

The Roguelite genre emerged as a way to differentiate these games from traditional Roguelikes. Roguelites introduce macro-level objectives by allowing players to carry over certain items or upgrades between attempts. This persistent progression system enables players to gradually become stronger across multiple playthroughs, increasing their chances of success in subsequent *runs*.

Some well-known commercial Roguelite games include *Hades*, *Enter the Gungeon*, *The Binding of Isaac*, *Rogue Legacy*, *Children of Morta*, and *Dead Cells*. These games offer diverse gameplays, game styles, lores, and features, as well as permanent elements that contribute to achieving cross-run objectives (e.g., weapons, currencies, upgrades, characters, etc.). Collectible resources, for example, can persist between deaths and be used to unlock permanent upgrades, enhancing the player's chances of success.

Failure is then an integral part of Roguelites. Players often face new mechanics, traps, challenging enemies, bosses, and various features that require learning, resulting in multiple failures and deaths before achieving their first complete playthrough or run. Despite the repeated losses, Roguelites typically feature fast restart times, swiftly immersing players back into the action. With each subsequent run, players gain a deeper understanding of the game's underlying mechanics, enabling them to progress further.

Replayability is also another significant aspect of Roguelites. Every run offers a distinct experience. Players can adapt their strategies and objectives based on the dynamically generated environments. The ever-changing nature of levels and encounters enhances the replayability of Roguelite games, as no two runs are identical. Additionally, many Roguelites incorporate replay value beyond simply reaching the game's end. This can take the form of a *new game+* mode (e.g., *Rogue Legacy*) or a storyline that requires defeating the final boss multiple times (e.g., *Hades* or *Dead Cells*). In both cases, players embark on a new, more challenging playthrough, relinquishing their progress towards the primary game objectives while retaining their persistent items and upgrades.

When playing a Roguelite game, several *moments* can be distinguished (Figure 2):

- A *play session* is a temporal session that begins when the player starts to play and ends when they stop playing. According to the duration of a play session, one or several runs can be concerned.
- A *run* or *playthrough* encompasses a complete game experience, starting from the game's beginning and ending either at game over or upon reaching the conclusion of the game (often involving a final boss battle integrated into the game's storyline). A run can be initiated within the current play session or be a continuation from a previous session if the game supports saving the player's progress.
- A game level session corresponds to the completion of a specific level within the game. Depending on the game, players may encounter a single large level to complete or multiple levels. Procedural generation can be applied to one or multiple game levels simultaneously. For example, in *Rogue Legacy*, a run involves the completion of a single large game level, structured into four areas (castle, forest, dungeon, and tower), all generated at the beginning of the run. In cases where a run consists of a sequence of consecutive game levels, each level is generated one after the other, often with an increasing level of difficulty.

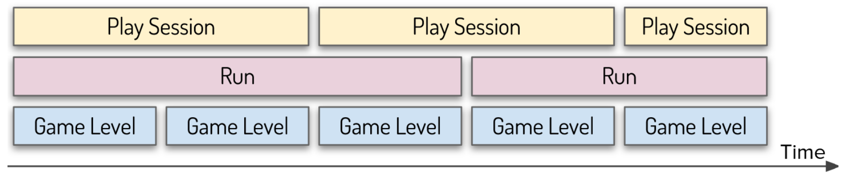


Fig. 2: Different times when considering playing a Rogue*lite* game [18]

3.2 Adequacy of Declarative Knowledge Training with Rogue*lite* Genre

The training of declarative knowledge requires learners-players to accomplish multiples training sessions, repeating the training activity, but with adapted content to train and varied ways in presenting the activity.

The procedural generation feature can be leveraged to generate diverse training scenarios and content, allowing for the training of declarative knowledge in a more engaging and varied manner. The Permadeath Mechanics can be utilized to reinforce the importance of knowledge retention and encourage players to learn from their incorrect answers or lack of knowledge. It can motivate players to actively acquire and retain declarative knowledge. The Difficulty Progression feature also consists in delivering a gradual increase in difficulty as players progress into the generated levels. This natural difficulty curve can be leveraged in serious games to provide a scaffolded learning experience. The game can start with simpler declarative knowledge challenges and gradually introduce more complex concepts as players improve, ensuring a suitable learning pace and maintaining engagement. The Replayability characteristic aligns well with the training of declarative knowledge, as repeated exposure and practice are key factors in knowledge retention. By designing a serious game with Roguelite elements, learners can revisit and reinforce their declarative knowledge through multiple playthroughs.

The Roguelite genre, with its challenging gameplay, strategic decision-making, and unpredictable nature, can provide an immersive and motivating learning environment. By integrating declarative knowledge challenges into its gameplay, such training games could enhance learner motivation, focus, and overall learning outcomes. Overall, the Roguelite genre offers a range of features and mechanics that can be effectively harnessed to design serious games for training declarative knowledge. Its procedural generation, permadeath mechanics, difficulty progression, replayability, and ability to captivate and motivate players make it a relevant and promising choice for creating engaging and effective training experiences. Therefore, Rogue*lite* seems to be a suitable genre for declarative knowledge training, where the training game activities generated are game levels.

3.3 Targeted Adaptations

The adaptation of generated game *and* training activities is not straightforward. It requires to be characterized from the game perspective as well as the learning perspective.

Adaptation is often characterized by three concepts [32]:

- the source (i.e., to what do we adapt?);
- the target (i.e., what is adapted?);
- the pathways (i.e., what methods are used to adapt the target to the source?).

Foremost, in our context, the adaptation targets generated game level (e.g. dungeons composed of interconnected rooms) and their elements (i.e., what is adapted). Therefore, it takes place during the generation of an activity (i.e., when it is adapted). In the spectrum of adaptation defined by [23], our targeted adaptation can be positioned in-between adaptivity and adaptability, as it uses user's data previously collected to automatically generate an activity that is adapted to the user.

In the literature, the gaming adaptations are mostly based on players/personalities profiles [30,22,9] or players characteristics, such as age and genre. In our context, adaptation from a game perspective seeks to take into account player preferences to choose the game elements (i.e., source). The main idea is to represent preferences as game elements that can be activated/deactivated by the player. From the learning perspective, our intention is to use knowledge of the learner (e.g., actual level, previous mistakes) from his/her learning path (source) to adapt the dungeons' difficulty in terms of educational content.

Since the adaptation is an integral part of the generation, in our context, this article does not dissociate them (i.e., the generation criterion includes adaptation, see Section 5).

3.4 Research Question

Designing a Rogue*lite* oriented game for training declarative knowledge requires to answer several design questions: *What is generated? How and when does the avatar die? What are the consequences? What varies? What indicates a progression?* And so on.

Furthermore, these questions need to be answered from both an educational and a game perspective. Moreover, in a prototyping design approach, the answers may change from one prototype to another.

Therefore, our research question is: *How can the design of Roguelite-oriented training be facilitated in a prototyping design approach?* Our proposal is an analysis framework that helps designers ask the right questions and make their choices explicit during each design iteration.

4 State of the Art

Numerous frameworks and methods for games, serious games, and game-based scenario design can be found in the literature. However, most of these frameworks are primarily focused on analyzing existing games [12].

One notable framework proposed by [7] is a learner-centered framework consisting of four dimensions: Representation, Context, Pedagogy, and Learner. This high-level design approach aids in the design of game-based learning scenarios but does not facilitate the transition from educational content to concrete game elements. Another framework, the DPE framework introduced by [33], extends the Mechanics, Dynamics, Aesthetics (MDA) framework [11] for serious games. The DPE framework is divided into three categories (design, play, experience) and is described by four criteria: Learning, Narrative, Gameplay, and User Experience.

Additionally, [26] presents a method involving a series of questions covering various aspects to consider during game design. This method is more generic and not specific to a particular type of knowledge or game genre. [1] describes the GOM II framework, an extension of the Game Object Model (GOM), which considers educational games as compositions of elements described by abstract and concrete interfaces. However, this work is theoretical and focuses on the general design of games rather than their specific implementation.

The LM-GM framework proposed by [3] enables the association of Learning Mechanics with Game Mechanics through the use of Serious Game Mechanics (SGM). However, this framework leans more towards game analysis rather than game design. [6] presents the ASTMG conceptual model based on activity theory, aiming to provide a better understanding of the relationships between serious game elements and learning objectives. Similarly, this framework is also more focused on game analysis.

Many other framework and methodologies exist [34,4,20,14,27]. Some of these approaches offer methods that are more closely aligned with specific game genres, such as adventure games or story-oriented games [8]. However, these works are primarily generic and not tailored to a specific type of knowledge or game genre.

Considering the specific context of the Roguelite genre, none of the existing frameworks fully meet the requirements. Nonetheless, these frameworks are not mutually exclusive and can be used in conjunction with each other. For example, [26] could be used to describe the general design of the game, while ASTMG [6] could be employed to ensure coherence in each prototype.

In summary, existing frameworks that share the goal of assisting in the design of learning games focus on different pedagogical objectives, knowledge types, and game genres. None of these frameworks specifically cater to situations where the game genre is already identified due to its relevance to a specific learning objective. Our analysis framework is specifically dedicated to Roguelite games for declarative knowledge retention, making it a highly specific scope. However, multidisciplinary teams also require tailored frameworks to guide them in designing relevant, adapted, and well-balanced learning games.

5 Analysis Framework for a Rogue*lite* Learning Game

Although the design tends to focus on the training and learning dimensions, the game aspect must not be neglected. Indeed, as Prensky noted [24], the main reason for learning game failure lies in their lack of gameplay. To this extent, the proposed framework aims to provide a means of analyzing the design needs of Rogue*lite*-oriented learning games by specifying both dimensions through specific criteria.

To design a Rogue*lite* game, the initial step involves defining game mechanics. This includes determining how the game world is generated (e.g., what is generated and how it is generated), when permanent death occurs, and how progression works (e.g., which elements are carried over). The generation mechanism, within this context, encompasses specifying the aspects that should vary during gameplay. Similar to learning, an important concept is the progression of difficulty, where it becomes crucial to define how difficulty increases and when it does so. From both perspectives, these five mechanisms (**Generation**, **Death/Hurt**, **Variety**, **Progress**, and **Difficulty**) serve as criteria for analyzing the design requirements. Each criterion comprises a set of questions related to its respective mechanism. Answering these questions is essential for clarifying the design needs of the educational game.

The following are the questions for each criterion:

1) Generation
- *Q1.* What elements are generated?
- *Q2.* When are these elements generated?
- *Q3.* Based on what criteria are they generated? (i.e., sources of generation)

2) Death/Hurt
- *Q4.* Under what circumstances can the avatar be injured or die?
- *Q5.* What are the consequences of being injured or killed?
- *Q6.* Where can the avatar sustain injuries or be killed?

3) Variety
- *Q7.* Which elements exhibit variation?
- *Q8.* How do these elements vary? (i.e., are the variations triggered by player action? Are they random? Is it a combination of both? Are they guided by heuristics?)

4) Progress
- *Q9.* What is preserved or carried over between each death? (i.e., which elements?)

5) Difficulty
- *Q10.* What factors contribute to increasing or decreasing the difficulty?
- *Q11.* How is the difficulty progression designed? (i.e., if multiple elements affect the difficulty, what is the sequence in which they occur?)

Table 2 provides a structure for conducting a needs analysis. Each criterion is represented by a row, which is further divided into X sub-rows, with each sub-row corresponding to a specific question. The columns represent different dimensions, one for the game aspect and another for the educational aspect. If there is shared information between both dimensions, it can be combined and specified in the related cells. This framework is not restricted to any particular didactic field and can be applied to design training games within the Roguelite

Table 2: Grid for Design Needs Analysis [18]

Criteria		Educational Perspective	Game Perspective
Generation	Q1		
	Q2		
	Q3		
Death/Hurt	Q4		
	Q5		
	Q6		
Variety	Q7		
	Q8		
Progress	Q9		
Difficulty	Q10		
	Q11		

genre. The subsequent section illustrates the application of this framework in the context of the AdapTABLES project.

6 Framework Application: AdapTABLES Project

This section is divided into five subsections. Subsection 6.1 provides a detailed analysis focused on the design requirements of the initial prototype. It outlines the necessary considerations and factors to be addressed during the design phase. The first prototype is then described in subsection 6.2. This subsection offers an overview of the existing implementation. In the subsection 6.3, feedback gathered from real-life conditions is presented. These informal feedback provides insights into how the prototype has been received and the observations made by users during its usage. The design needs analysis for the subsequent prototype is detailed in subsection 6.4. It specifies the requirements and improvements that need to be addressed to enhance the design and functionality of the game. The second prototype, currently in development, is presented in the subsection 6.5.

6.1 First Analysis

In our iterative process, the first step involved conducting a design needs analysis for the initial prototype. The primary objective was to set aside the knowledge structure and concentrate on a single task, specifically *Completion 1* for multiplication tables. This task was manually integrated into the game, ensuring that the information remains persistent across different training sessions. An overview of the design needs analysis for the first prototype is provided in Table 3.

Generation. The generated element (**Q1**) in the game design is a dungeon level comprising various elements, such as rooms, their order, contents, positions, and values. Each room can be categorized as either a question room or a no-question room. A question room is associated with a specific training task defined by the teacher and configured before training. On the other hand, a no-question room is designed for entertainment purposes and includes enemies, traps, and other game elements. The inclusion of non-question-based rooms aims to prevent learners-players from perceiving the game as merely a disguised questionnaire. Following the game flow depicted in Figure 2, a new dungeon level is generated (**Q2**) when the player requests it during gameplay.

As discussed in Section 3.3, our game adaptations are tailored to cater to players' preferences. To identify these preferences, we examined existing Roguelite games and discovered that many of them incorporate a purchasing mechanism for items such as equipment, upgrades, and skills. Collaborating with game designers, we categorized the preferences into three types: 1) Content, 2) Rules, and 3) Visuals & Audio (**Q3**). Content preferences encompass additional objects that exist during gameplay or elements that modify the activity's structure when activated. Examples include extra lives or different dungeon modes, such as linear or labyrinthine layouts. Rules preferences involve elements that impact the players, the avatar, or the Non-Player Characters (NPCs) behavior. Examples of rules preferences are increasing the speed of enemies or introducing a game goal where completing an activity without mistakes earns +10 coins.

Regarding the game dimension, the generation process takes into account the three types of preferences mentioned earlier (**Q3**). For example, if a player has purchased and activated the labyrinthine mode, the generation algorithm considers this preference. To ensure the tracking of learners' in-game progress, the generator also takes into consideration the last level number and state. If the previous level, let's say #5, was successfully completed, the next level generated will be #6. However, in the case of death, the next level will be reset to #1. The level number influences various aspects of the dungeon, such as its length, the number of rooms with questions, and the overall dungeon effects (levels above #4 are set in dark mode). In the educational dimension, the parameters for the tasks can vary based on the learners' level (as discussed in Section 2). Therefore, each learner has their own customized setup for *Completion 1* task, defined either individually or shared by the teachers. These parameter values are utilized to generate relevant questioned facts associated with the rooms that have questions (**Q3**). To avoid repeating successful questioned facts, the system takes into account the previously encountered facts and their outcomes.

It is important to note that conflicts can arise between the game and training dimensions. For instance, if a learner-player has activated the "labyrinthine" mode while the task setup requires encountering questioned facts in a specific ascending or descending order. In such cases, our recommendation is to prioritize the training dimension over the game dimension.

Death/Hurt. When the avatar is injured (**Q5**), it loses a life, and when all lives are depleted, it results in the avatar's death. The player (via the avatar) can get hurt (**Q4**) by coming into contact with enemies, falling into pits (game dimension), or by answering questions incorrectly (educational dimension). Additionally, running out of time, which is predetermined in the task parameters, can also result in the avatar getting hurt. Incorrectly answering a question or running out of time can only occur in question rooms (**Q6**), while encountering the wrong enemy or falling into a pit can happen in both types of rooms.

Variety. The game generation algorithm offers a range of choices for different elements. In the context of Roguelite games, variations mainly include selecting different object types, determining their positions, shaping the objects and dungeons, and controlling the quantity of elements present (**Q7**). Consequently, the positions of decorative objects and the shapes of rooms are selected "randomly" (**Q8**), while maintaining coherence to ensure that elements are not placed outside the room or inaccessible to the avatar. The gameplay itself represents how learners carry out the tasks. Having only one type of gameplay per task can lead to a sense of repetition. To avoid this, four distinct gameplay variations (**Q7**) have been identified for the Completion 1 task type: opening chests with the correct answer, passing through doors with the correct answer, touching enemies with the correct answer, or typing in the correct answer. This diversification of gameplay helps prevent monotony and adds variety to the learning experience.

In the learning dimension, the facts to be practiced are distinct until each fact has been encountered at least once (**Q7**). Furthermore, depending on the parameters specified in Table 3, the format of the facts (e.g., missing elements, position of the equals sign, etc.) varies (**Q8**). While there is an element of randomness involved in the variation of these elements, it is constrained by the game's preferences, educational considerations (i.e., choices made by teachers), and previous selections made by the algorithm.

Progress. Our approach involves implementing a purchase mechanism inspired by Roguelite games. Players have the ability to buy items and subsequently activate or deactivate them as desired. Therefore, game progression can be observed through the elements purchased and the number of coins accumulated (**Q9**). Some Roguelite games only retain progression when players successfully complete levels or dungeons without the avatar dying. As a result, coins are earned only upon completing a dungeon entirely. These coins can be obtained during the dungeon journey (randomly appearing when opening a correct chest) or at the end of the dungeon based on activated rules (e.g., +10 coins for completing the dungeon level without any mathematical errors). Training progression becomes apparent at the end of a dungeon run, whether by reaching the end or experiencing the avatar's demise, where statistics are presented showcasing the mistakes made, correct answers given, and areas that require further improvement. These results persist across subsequent runs.

Difficulty. In the game dimension, the difficulty increases within a run by progressively increasing the number of rooms with questions (e.g., starting with 5 rooms, then 7, then 9, then 11, and so on) (**Q10**). The total number of rooms may still vary due to the generation process, which can randomly include rooms without questions along with the overall structure of the dungeon, such as a tortuous but linear layout or a labyrinthine design (**Q11**). After successfully completing five levels without losing, a more challenging level is introduced where the player navigates in darkness, with only a torch illuminating the avatar (**Q10**). In the educational dimension, the difficulty increases based on parameters defined by teachers. As long as these parameters remain unchanged, the questioned facts will consistently reflect the established setup (**Q11**). These design choices regarding difficulty progression are open to debate and aim to establish an initial version of the game's difficulty curve, which is subject to further refinement.

Table 3: Design Needs of the prototype #1 [18]

Criteria		Educational Perspective	Game Perspective
Generation	*Q1: What?*	One task and one questioned fact per room-with-question	Dungeon + rooms + entry + exit
	Q2: When?	When a new game level is required	
	Q3: Based on?	"Completion 1" set-up Current progress among possible facts	Previous level number and state Activated game elements or rules
		Task parameters have priority on activated game elements if conflict	
Death/Hurt	*Q4: What?*	Incorrect answers or time out	Being touched by foes, falling into holes
	Q5: When?	Injuring causes heart lost, no more hearts causes death	
	Q6: Where?	Question rooms	Any room with foes or holes
Variety	*Q7: What?*	Facts	Rooms with gameplay and content
	Q8: How?	Progress and past results	Random
Progress	*Q9: What?*	Success or failure on met questioned facts	Coins collected during successful game levels + purchased elements
Difficulty	*Q10: What?*	Questioned facts	Dungeon level length + dark mode
	Q11: How?	In relation with the task parameters	According to previous level number and state

6.2 First Prototype

The first prototype of the game was developed using the Unity game engine, employing C# scripts to create a 2D game. It has been exported and deployed as a Web platform WebGL build, allowing for easy accessibility. The game incorporates an HTTP REST API, developed in .Net Core, to store data in a NoSQL MongoDB database. Additionally, a web application teacher dashboard has been created using .NET with Blazor. This dashboard enables teachers to monitor their students' progress, including the current multiplication parameter settings, achievements related to multiplication facts, and purchased game elements. Currently, the available version of the game is in French and can be played using a gamepad or a keyboard.

Figure 3 showcases six screenshots from the prototype. Screen 3a displays a portion of the "hub" area, featuring four accessible elements: statistics ("Stats Générales"), progress ("Progression"), educational settings ("Réglages"), and the purchase panel ("Achats") for game preferences). The hub area serves as the starting point for each run and is where players can review their progress or manage collectibles, such as purchases and activation/deactivation of game elements. Screen 3b presents a section from the educational settings panel, as outlined in Table 1. Screen 3c showcases a segment from the item purchase panel.

Screens 3g, 3e, and 3f illustrate examples of the currently implemented gameplay mechanics within the dungeon levels, including selecting the correct foe, door, and chest, respectively. Finally, Screen 3h displays an example of a room where the player needs to type in their answer.

The screenshots provide a visual representation of the prototype, demonstrating various aspects of the game's interface and gameplay mechanics.

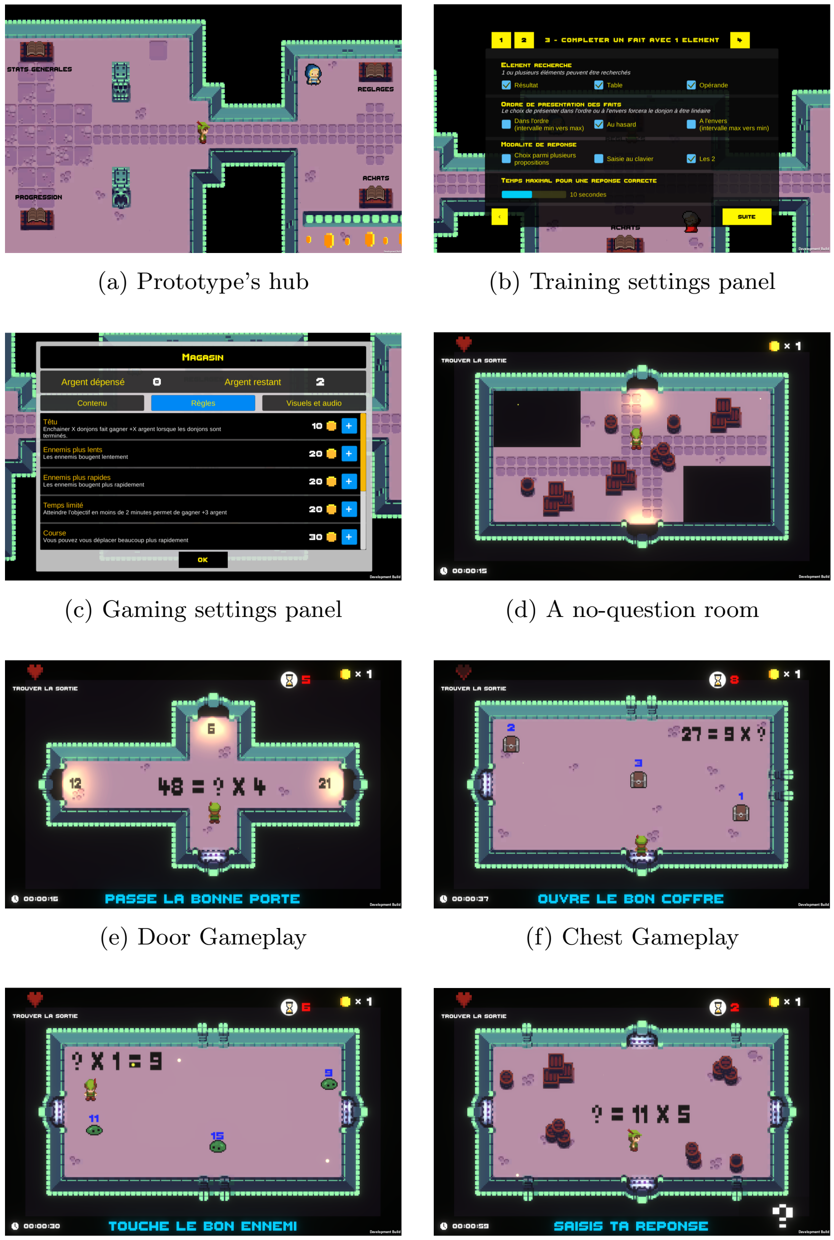
6.3 Experiment Feedback

The design of the prototype underwent three iterations, during which it was tested in real conditions and refined based on feedback from teachers and students. The empirical feedback encompassed various aspects, including ergonomic concerns related to keyboard versus gamepad usage, the overall playability experience, the inclination for replayability, and the motivation to play and practice multiplication tables.

Regarding the death/injury criterion, several issues were identified. Firstly, children sometimes made unintended choices due to the current touch-oriented interactions that did not require the use of buttons or keys. Although this gameplay problem is tied to ergonomics, it could lead to a sense of unfairness in the reward/punishment system. Secondly, some rooms had foes positioned randomly in close proximity to the avatar's entry area. This resulted in children losing hearts without sufficient time to avoid them. Additionally, certain rooms with questions also contained holes to avoid. Teachers pointed out that these game elements could distract children when they should be focused on answering the questions.

In terms of the variety criterion, children appreciated the three different gameplays for selecting an answer (door, chest, and foe). However, it was noted that the prototype lacked sufficient variation. The gameplay involving touching the correct foe was considered confusing by both children and teachers. In some rooms, foes needed to be avoided, while in others, players had to guide their avatar to touch the foe corresponding to their chosen answer. This counter-intuitive approach led to teachers suggesting that associating a correct answer with a negative action (i.e., killing foes) should be avoided. The prototype also included rooms with questions that required directly typing in the correct answer on the keyboard. Based on the correctness of the answer, the correct door or chest would open, or all foes would die. An incorrect answer resulted in opening an empty chest, leading to a door that led to a dead end, or having no effect on the touched foe. In all three cases, however, the avatar would be injured. These situations were initially designed to vary the response method while maintaining similar room content but ultimately proved to be confusing.

In terms of progression, it was found that collecting coins was unbalanced. Only gameplay modes focused on chests (choosing or typing answers) had a chance to randomly contain '+1' or '+3' coins (or a 'heart'/life). While certain purchased and activated rules could provide alternative ways to earn coins, early successful dungeon levels might not result in earning any coins. Additionally, the initial items purchased primarily fell under the content category (such as extra



(a) Prototype's hub
(b) Training settings panel
(c) Gaming settings panel
(d) A no-question room
(e) Door Gameplay
(f) Chest Gameplay
(g) Foe Gameplay
(h) Answer entry

Fig. 3: Screenshots of the features from the first prototype (text in French) [18]

hearts), which were preferred over rules. Furthermore, some teachers were unconvinced about allowing learners to freely buy and activate rules that pressured them to act faster, potentially causing stress to answer quickly.

As mentioned earlier, the generation process could disable certain activated rules that did not align with the current configuration of the *Completion 1* task. This could lead to feelings of confusion when encountering the generated dungeon levels.

6.4 Second Analysis

This second analysis was conducted to inform the design of the upcoming prototype for multiplication tables training. Building upon the feedback received from teachers and learners, we collaborated with game designers to identify solutions and determine further directions for both the game and training dimensions.

Similar to the first prototype's functional scope, the knowledge structure is still not considered, although all five task types (presented in the subsection 2.2) are now considered. The prototype will allow manual parameterization of the current training configuration for individual children based on their specific progress, represented by the {objective, level} pair in the learner's training path (refer to Figure 1).

Table 4: Design Needs of the Future Prototype (**Bold** describes changes from the first analysis).

<table>
<tr><th colspan="2">Criteria</th><th>Educational
Perspective</th><th>Game
Perspective</th></tr>
<tr><td rowspan="4">Generation</td><td>Q1: What?</td><td>One task and one questioned fact per room-with-question</td><td>Dungeon structure + rooms</td></tr>
<tr><td>Q2: When?</td><td colspan="2">When a new game level is required</td></tr>
<tr><td rowspan="2">Q3: Based on?</td><td>All tasks set-up
Current progress among possible facts</td><td>Previous level number and state
Equipped items</td></tr>
<tr><td colspan="2">Task parameters have priority on activated game elements if conflict</td></tr>
<tr><td rowspan="3">Death/Hurt</td><td>Q4: When?</td><td>Incorrect answers or time out</td><td>Being touched by foes, falling into holes</td></tr>
<tr><td>Q5: What?</td><td colspan="2">Injuring causes heart lost, no more hearts causes death</td></tr>
<tr><td>Q6: Where?</td><td>Question rooms</td><td>Only rooms with no question</td></tr>
<tr><td rowspan="3">Variety</td><td>Q7: What?</td><td>Facts</td><td>Different types of rooms, types of gameplays, types of elements</td></tr>
<tr><td rowspan="2">Q8: How?</td><td>Progress and past results</td><td>Based on the available equipments, gameplays, elements,</td></tr>
<tr><td colspan="2">and in relation to the tasks $\Longleftrightarrow$ gameplays mappings</td></tr>
<tr><td>Progress</td><td>Q9: What?</td><td>Success or failure on met questioned facts</td><td>Coins collected during successful game levels + purchased items</td></tr>
<tr><td rowspan="2">Difficulty</td><td>Q10: What?</td><td>Questioned facts</td><td>Dungeon level length
+ curses</td></tr>
<tr><td>Q11: How?</td><td>In relation with the task parameters</td><td>According to previous level number and state</td></tr>
</table>

Generation. The generated element (**Q1**) remains a dungeon level consisting of organized rooms categorized as either question-free or rooms with one question associated with a specific task type as per the training setup. Each room contains various interactive elements. The generation of a new game level occurs when the player requests it, either from the hub-room to start a new run or after the debriefing screens following a successful dungeon level (**Q2**). From the game dimension, the generation process continues to take into account the last level number and state (**Q3**), as well as the features that have been purchased and activated. However, the purchasable and activatable elements have been modified, and these elements are further explained in the following categories. From an educational perspective, each generation considers the learner's current configuration for all task types (ranging from 1 to 5) and takes into account the previously encountered questioned facts and their results.

Death/Hurt. The player continues to experience injury (**Q4**) when interacting with foes, falling into pits (game dimension), answering questions incorrectly, or running out of time (educational dimension). However, a significant difference is that question rooms will no longer contain traps or game elements that can harm the avatar (**Q6**). Incurring injuries in question rooms will solely result from providing incorrect answers or exceeding the time limit. The consequences of sustaining an injury remain unchanged (**Q5**), leading to the loss of a life or resulting in the avatar's death if there are no more hearts left.

Variety. In the initial prototype, various types of rooms combined game elements and gameplays. However, to introduce greater room variety (**Q7**), we have adopted a new approach, as depicted in the conceptual class diagram presented in Figure 4. This approach involves two key aspects. Firstly, the different *types of tasks* are mapped to specific *gameplay types* based on the current task parameters (mapping work presented in [19]). Each gameplay type requires a *quantified* number of elements possessing the specified *ability*. Secondly, the *types of rooms* are defined by their *positions*, which can accommodate different elements possessing specific *abilities* (and sizes). Consequently, different *types of game elements* are associated with the *abilities* they can manage.

These combined elements will play a crucial role in determining the generation of rooms and their respective built-in elements (**Q8**). Consequently, the purchase mechanic now involves players selecting items that, when equipped (i.e., activated), unlock new types of gameplays that can occur in specific rooms, as long as they align with the associated task. By providing variants of game elements that share certain abilities, game developers can enhance the potential for variations within the game.

Progress. In terms of progression, equipment items can now be purchased and retained across multiple runs. The coin mechanism remains, but it will be adjusted so that learners earn one coin for each correct answer. Furthermore, questioned facts encountered and their associated results will be saved even after the avatar's death. As a result, progression can be observed through the availability of equipment items, the number of coins accumulated, and the stored results of facts, which are accessible outside of a game level (**Q9**).

Difficulty. The educational difficulty remains consistent with the first prototype, determined by the parameters of the tasks set by the teachers (**Q11**). From a gameplay perspective, the progression based on the length of the dungeons is maintained, but a new gameplay mechanism called “curses” is introduced (**Q10**). The game progression will be structured around different minimum thresholds, with each threshold unlocking curses that may or may not occur during the generated dungeon level (**Q11**). For instance, if we consider thresholds every three dungeon levels, the generation of level #10 could involve a maximum of three curses or, with some luck, none at all. The inclusion of curses adds an additional layer of challenge and unpredictability to the gameplay, increasing the overall difficulty and providing new obstacles for players to overcome.

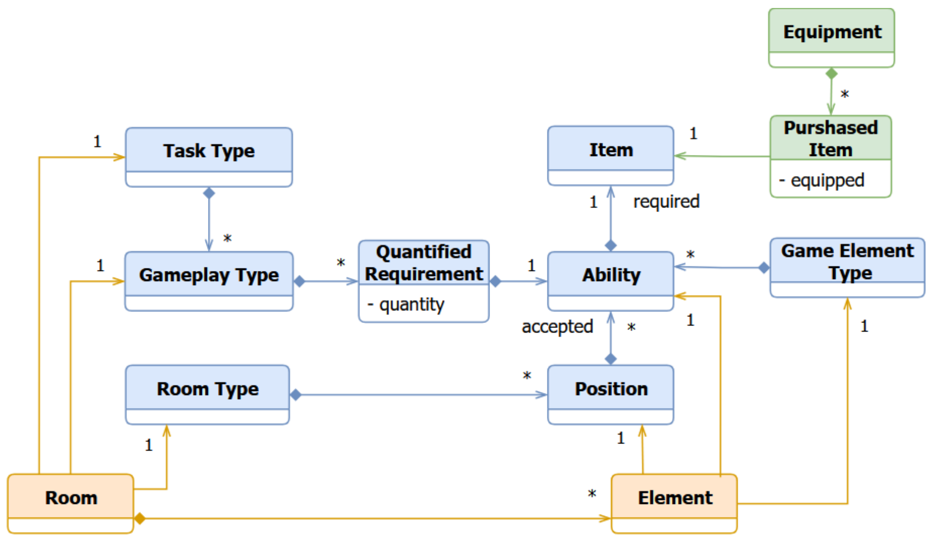


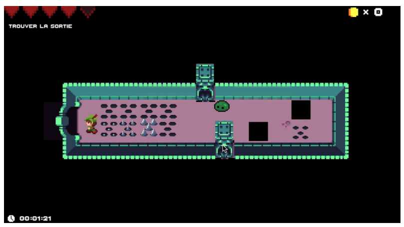
Fig. 4: Conceptual class diagram illustrating the domain elements and relationships involved in considering a wide variety of rooms with question. Yellow concepts are to be generated; blue concepts are specifications of the game and didactic elements available; green concepts concern each learner-player.

6.5 Second Prototype

The second version of the game is being created using the Unity game engine and is accessible through a Web platform WebGL build. However, there has been a change in the game’s structure compared to the first prototype. The software responsible for generating customized and diverse learning activities is no longer integrated within the game itself. Instead, it is now being developed separately. This separation allows us to focus on designing the generator as a distinct research subject, while the game interprets the levels generated by the generator.

(a) New shop

(b) Equipment panel



(c) A no-question room with traps

(d) A question room in dark mode

(e) Completion 1 Gameplay

(f) Reconstitution Gameplay

(g) Identification Gameplay

(h) Membership Id. Gameplay

Fig. 5: Screenshots of the features from the second prototype (some texts are in French)

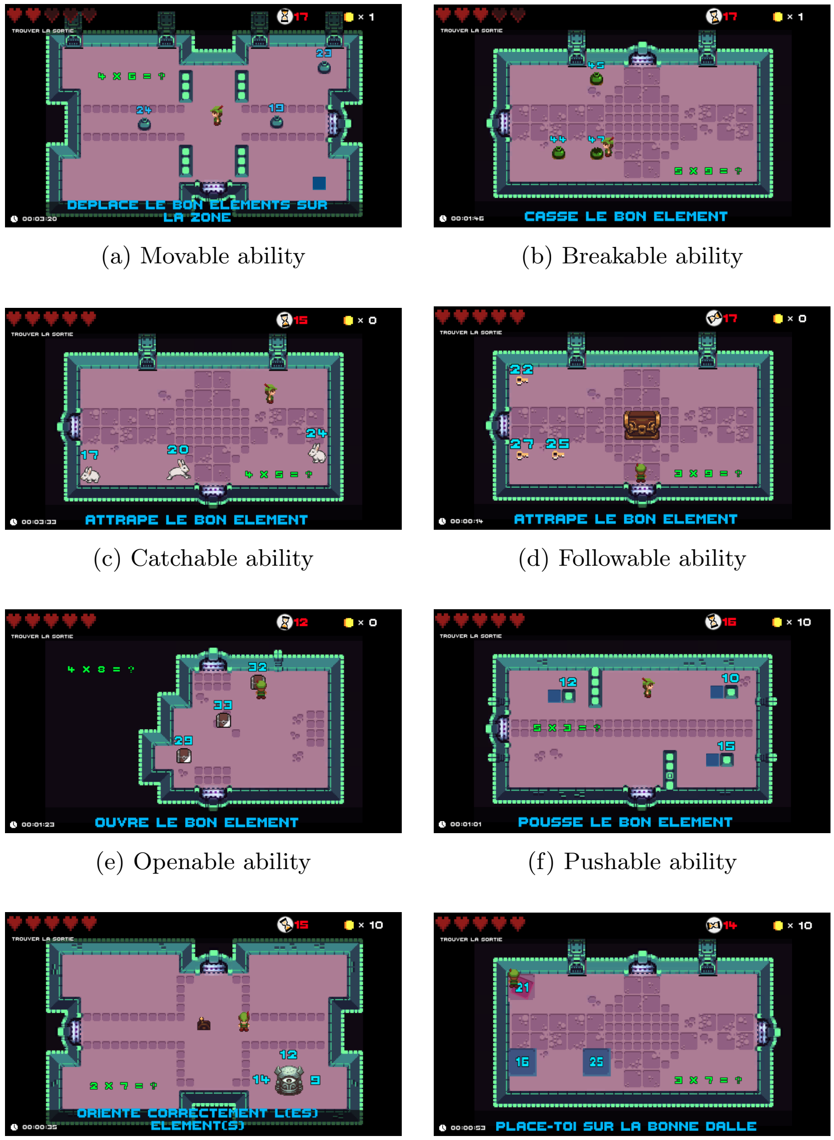
To create the generator component, we are using Java and drawing upon theories from the Model-Driven-Engineering (MDE) field [13]. Additionally, we are using the practical tools provided by the *Eclipse Modeling Framework* [28] and the *Epsilon Framework* [16]. We have developed a servlet-based HTTP REST API to facilitate communication with the generator. The Unity game receives the levels in the form of XML files. As for user input, the game can still be played using either a gamepad or a keyboard.

We illustrate the major changes from the first prototype in Figure 5. Screens 5a shows how learners-players can purchase equipment and activated or deactivated them in 5b. Some items allow to augment the avatar's hearts. Other items concern the abilities and allow to extend the encountered gameplays. Last items can ease the progression through the potential curses, e.g. the lantern that surround the avatar with a light, making easier the play through the darkest level, or the encompass that unlock the display of a map of the current dungeon level, useful with a labyrinthine dungeon to distinguish rooms already seen from others. Screens 5c and 5d are examples of random traps and dark mode situations that players can met. Four of the five tasks types are then illustrated. Screen 5e is about the *Completion 1* task. It uses a *statue* having the *rotable* ability to propose 3 choices, validation of a choice being made by triggering the *switch*. Screen 5f illustrates the use of various *floor interrupts* to do/undo (*selectable* ability) make choices of values for reconstituting the fact (*Reconstitution* task). Screen 5g is also using a *rotable* ability with a *statue* to let learners choose if the proposed fact is correct or incorrect (*Validation* task). Last screen 5h illustrates the *Membership identification* task. The learner-player have to push to the left all blocks proposing a correct proposition (a result for table 10). Figure 6 illustrates different abilities, and related gameplays and elements, for a same task (*Completion 1*).

This second prototype has not been evaluated yet. However, it was presented during The Science Festival 2023, known as "la fête de la science" in French, which is an annual event aimed at promoting scientific knowledge and discovery among the public in France. In addition, many experiments are planned in order to collect qualitative feedback from both teachers and learners viewpoints (from different school grades).

7 Conclusion

In this article, two main points were covered. Firstly, it discussed the suitability of the Rogue*lite* game genre for training declarative knowledge. Secondly, it introduced a framework for conducting a design needs analysis specifically tailored for Rogue*lite*-oriented learning games. The proposed framework offers an initial understanding of the mechanisms and choices that benefit both the game and training aspects. The core idea is to enable a two-dimensional design approach that considers both the play and learning dimensions separately, while ensuring their compatibility through verification and maintenance. This framework was applied in the context of the AdapTABLES project, which aims to develop a multiplication tables training game. The article describes how the framework was utilized in two iterations.



(a) Movable ability (b) Breakable ability

(c) Catchable ability (d) Followable ability

(e) Openable ability (f) Pushable ability

(g) Rotable ability (h) Standable ability

Fig. 6: Different abilities, gameplays and elements for a same Completion 1 task

The first advantage of the framework is the traceability of design choices. Indeed, at each iteration, the choices are explained and then summarized. This traceability facilitates the evolution of the design without the risk of involuntarily going backwards. On the other hand, the visual synthesis (cf. Table 2) makes it possible to check that none of the dimensions has been neglected (i.e., presence of an empty box in case of neglect). In a prototyping approach, iterations are essential to fix certain settings. However, it would seem that the use of such a framework (i.e., allowing traceability as well as visual verification of the non-neglect of a dimension) could reduce the number of iterations required. Finally, this tool provides a support that can be understood by all stakeholders of the design process.

However, the proposed framework takes into account a rather precise context: training or retrieval practice in the context of the Rogue*lite* video game genre. Moreover, the criteria used are those that we consider essential. Consequently, other criteria might be considered essential by other researchers or game designers. In particular, some criteria depending on the application domains might be interesting to add.

Looking ahead, we are interested in applying this framework to other fields beyond mathematics. Currently, they are actively working on implementing it for history and geography topics, such as historical dates and countries of the European Union.

References

1. Amory, A.: Game object model version II: A theoretical framework for educational game development. Educational Technology Research and Development **55**(1), 51–77 (Jan 2007). https://doi.org/10.1007/s11423-006-9001-x
2. Anderson, J.R., Lebiere, C.J.: The Atomic Components of Thought. Psychology Press, 0 edn. (Jan 2014). https://doi.org/10.4324/9781315805696, `https://www.taylorfrancis.com/books/9781317778318`
3. Arnab, S., Lim, T., Carvalho, M.B., Bellotti, F., de Freitas, S., Louchart, S., Suttie, N., Berta, R., De Gloria, A.: Mapping learning and game mechanics for serious games analysis: Mapping learning and game mechanics. British Journal of Educational Technology **46**(2), 391–411 (Mar 2015). https://doi.org/10.1111/bjet.12113
4. Barbosa, A.F.S., Pereira, P.N.M., Dias, J.A.F.F., Silva, F.G.M.: A New Methodology of Design and Development of Serious Games. International Journal of Computer Games Technology **2014**, 1–8 (2014). https://doi.org/10.1155/2014/817167
5. Brame, C.J., Biel, R.: Test-Enhanced Learning: The Potential for Testing to Promote Greater Learning in Undergraduate Science Courses. CBE—Life Sciences Education **14**(2) (Jun 2015). https://doi.org/10.1187/cbe.14-11-0208
6. Carvalho, M.B., Bellotti, F., Berta, R., De Gloria, A., Sedano, C.I., Hauge, J.B., Hu, J., Rauterberg, M.: An activity theory-based model for serious games analysis and conceptual design. Computers & Education **87**, 166–181 (Sep 2015). https://doi.org/10.1016/j.compedu.2015.03.023
7. De Freitas, S., Jarvis, S.: A Framework for developing serious games to meet learner needs. Interservice/Industry Training, Simulation & Education Conference, I/ITSEC (Jan 2006)
8. De Lope, R.P., Medina-Medina, N., Soldado, R.M., Garcia, A.M., Gutierrez-Vela, F.L.: Designing educational games: Key elements and methodological approach. In: 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games). pp. 63–70. IEEE, Athens, Greece (Sep 2017). https://doi.org/10.1109/VS-GAMES.2017.8055812

9. Goldberg, L.R.: An alternative" description of personality": The big-five factor structure. Journal of personality and social psychology **59**(6), 1216–1229 (1990)
10. Harris, J.: The Berlin Interpretation. In: Exploring Roguelike Games, pp. 37–43. CRC Press (Sep 2020). https://doi.org/10.1201/9781003053576-9
11. Hunicke, R., LeBlanc, M., Zubek, R.: MDA: A formal approach to game design and game research. In: Proceedings of the AAAI Workshop on Challenges in Game AI. vol. 4. San Jose, CA (2004)
12. Junior, R., Silva, F.: Redefining the MDA Framework—The Pursuit of a Game Design Ontology. Information **12**(10) (Sep 2021). https://doi.org/10.3390/info12100395
13. Kent, S.: Model Driven Engineering. In: Integrated Formal Methods. pp. 286–298. Springer Berlin Heidelberg (2002)
14. Kiili, K.: Digital game-based learning: Towards an experiential gaming model. The Internet and Higher Education **8**(1), 13–24 (2005). https://doi.org/10.1016/j.iheduc.2004.12.001
15. Kim, J.W., Ritter, F.E., Koubek, R.J.: An integrated theory for improved skill acquisition and retention in the three stages of learning. Theoretical Issues in Ergonomics Science **14**(1), 22–37 (Jan 2013). https://doi.org/10.1080/1464536X.2011.573008
16. Kolovos, D., Rose, L., Paige, R., Garcia-Dominguez, A.: The Epsilon Book. Eclipse (2010)
17. Laforcade, P., Mottier, E., Jolivet, S., Lemoine, B.: Expressing adaptations to take into account in generator-based exercisers: An exploratory study about multiplication facts. In: 14th International Conference CSEDU. Online Streaming, France (Apr 2022). https://doi.org/10.5220/0011033100003182
18. Lemoine, B., Laforcade, P., George, S.: An Analysis Framework for Designing Declarative Knowledge Training Games Using Roguelite Genre. In: 15th International Conference on Computer Supported Education. vol. 2, pp. 276–287. SCITEPRESS - Science and Technology Publications, Prague, Czech Republic (Apr 2023). https://doi.org/10.5220/0011840200003470, `https://hal.science/hal-04093534`
19. Lemoine, B., Laforcade, P., George, S.: Mapping task types and gameplay categories in the context of declarative knowledge training. In: Proceedings of the 15th International Conference on Computer Supported Education, CSEDU 2023, Volume 2, Prague, Czech Republic, April 21-23, 2023. pp. 264–275. SCITEPRESS (2023). https://doi.org/10.5220/0011840100003470
20. Marne, B., Wisdom, J., Huynh-Kim-Bang, B., Labat, J.M.: The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library. In: 21st Century Learning for 21st Century Skills, vol. 7563, pp. 208–221. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33263-0_17
21. Marty, J.C., Carron, T.: Hints for improving motivation in game-based learning environments. In: Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches, pp. 530–549. IGI Global (2011)
22. Nacke, L.E., Bateman, C., Mandryk, R.L.: BrainHex: A neurobiological gamer typology survey. Entertainment Computing **5**(1), 55–62 (2014). https://doi.org/10.1016/j.entcom.2013.06.002
23. Oppermann, R., Rashev, R.: Adaptability and adaptivity in learning systems. Knowledge transfer **2**, 173–179 (1997)
24. Prensky, M.: Computer Games and Learning: Digital Game-Based Learning. Handbook of Computer Game Studies (2005)

25. Roediger, H.L., Pyc, M.A.: Inexpensive techniques to improve education: Applying cognitive psychology to enhance educational practice. Journal of Applied Research in Memory and Cognition **1**(4), 242–248 (Dec 2012). https://doi.org/10.1016/j.jarmac.2012.09.002
26. Schell, J.: The Art of Game Design: A Book of Lenses. CRC press (2008)
27. Silva, F.G.M.: Practical Methodology for the Design of Educational Serious Games. Information **11**(1), 14 (Dec 2019). https://doi.org/10.3390/info11010014
28. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. Addison-Wesley Professional, Boston, Massachusetts (2008)
29. Streicher, A., Smeddinck, J.D.: Personalized and Adaptive Serious Games. In: Dörner, R., Göbel, S., Kickmeier-Rust, M., Masuch, M., Zweig, K. (eds.) Entertainment Computing and Serious Games, vol. 9970, pp. 332–377. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46152-6_14
30. Tondello, G.F., Wehbe, R.R., Diamond, L., Busch, M., Marczewski, A., Nacke, L.E.: The Gamification User Types Hexad Scale. In: Proceedings of the Annual Symposium on Computer-Human Interaction in Play. pp. 229–243. ACM, Austin Texas USA (Oct 2016). https://doi.org/10.1145/2967934.2968082
31. Toy, M., Wichman, G., Arnold, K., Lane, J.: Rogue [digital game] (1980)
32. Vandewaetere, M., Desmet, P., Clarebout, G.: The contribution of learner characteristics in the development of computer-based adaptive learning environments. Computers in Human Behavior **27**(1), 118–130 (Jan 2011). https://doi.org/10.1016/j.chb.2010.07.038
33. Winn, B.: Handbook of Research on Effective Electronic Gaming in Education:. IGI Global (2009). https://doi.org/10.4018/978-1-59904-808-6
34. Yusoff, A., Crowder, R., Gilbert, L., Wills, G.: A Conceptual Framework for Serious Games. In: 9th IEEE International Conference on Advanced Learning Technologies. pp. 21–23. Riga, Latvia (Jul 2009). https://doi.org/10.1109/ICALT.2009.19