

# Measuring the difficulty of activities for adaptive learning

Francisco J. Gallego-Durán<sup>1</sup> · Rafael Molina-Carmona<sup>1</sup> · Faraón Llorens-Largo<sup>1</sup>

Published online: 13 July 2017  
© Springer-Verlag GmbH Germany 2017

**Abstract** An effective adaptive learning system would theoretically maintain learners in a permanent state of flow. In this state, learners are completely focused on activities. To attain this state, the difficulty of learning activities must match learners' skills. To perform this matching, it is essential to define, measure and deeply analyze difficulty. However, very few previous works deal with difficulty in depth. Most commonly, difficulty is defined as a one-dimensional value. This permits ordering activities, but limits the possibilities of deep analysis of activities and learners' performance. This work proposes a new definition of difficulty and a way to measure it. The proposed definition depends on learners' progress on activities over time. This expands the concept of difficulty over a two-dimensional space, also making it drawable. The difficulty graphs provide a rich interpretation with insights into the learning process. A practical case is presented: the PLMan learning system. This system is formed by a web application and a game to teach computational logic. The proposed definition is applied in this context. Measures are taken and analyzed using difficulty graphs. Some examples of these analyses are shown to illustrate the benefits of this proposal. Singularities and interesting spots are easily identified in graphs, providing insights in the activities. This new

information lets experts adapt the learning system by improving activity classification and assignment. This first step lays solid foundations for automation, making the PLMan learning system fully adaptive.

**Keywords** Difficulty estimation · Difficulty measure · Learning activity · Adaptive learning

## 1 Introduction

Adaptive learning is a set of strategies to improve the learning process by adapting it to learners' progression. It is usually based on a technological platform that presents activities to learners in an adapted way, collects their responses and allows them to track their own learning progress. Present research in adaptive learning is focused on sequencing the curriculum in a progressive way, adjusting pace to learners' progression and learning style, taking prior knowledge into account and customizing presentation of lessons to learners' features. Most works base their adaptation on students' learning styles only. These learning styles are measured through standard tests that learners fill in advance. In some cases, learning styles are reconsidered during interaction with the system, though adaptation of pace to learners' progression is still a secondary feature in many systems.

An important aspect to consider when adapting pace is the concept of flow. Flow can be defined as a feeling of complete focus in an activity. Learners achieve a state of flow when their skills match the difficulty of the activity. In other words, difficulty is not too high to generate anxiety nor too low to produce boredom. The key research challenge here is properly measuring difficulty and learners' skills.

✉ Francisco J. Gallego-Durán  
fjgallego@ua.es

Rafael Molina-Carmona  
rmolina@ua.es

Faraón Llorens-Largo  
Faraon.Llorens@ua.es

<sup>1</sup> Cátedra Santander-UA de Transformación Digital,  
Universidad de Alicante, Avda. de la Universidad s/n,  
Alicante, Spain

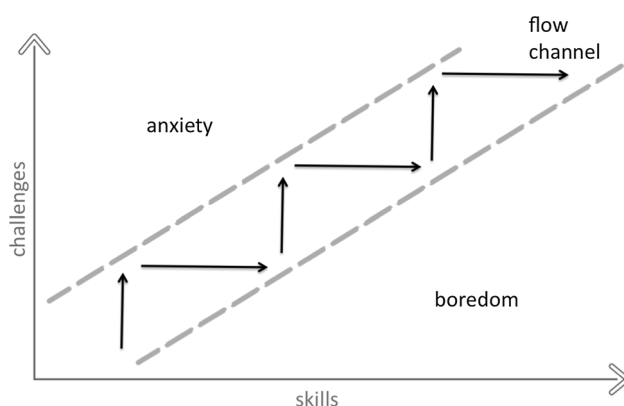
Some previous works propose measures of difficulty, many of them in the field of video games. Difficulty is usually understood as the effort required to successfully complete an activity. Although difficulty is considered a key factor to foster learners' motivation, existing definitions remain subjective. A formal definition and an agreed way to measure difficulty is yet to be proposed.

This work proposes a new mathematical definition for difficulty and a way to measure it. The definition arises from considering learning activities, learners' progress and the concept of flow. The measure is defined bi-dimensionally, in contrast to most previous works. It considers learners' progress over activity time, yielding much more expressiveness. When this measure is graphed, appearing features allow a richer interpretation of difficulty and learners' progress. Cost, hurdles, interest points and other singularities in learners' progression become self-evident in the graphs. This information expands new possibilities for adapting difficulty to learners' pace. Besides the formal definition, a practical case is presented to illustrate all these concepts. Measures and adaptation are shown using a custom learning system and PLMan [28], a game designed to learn computational logic.

A brief background about the research context is presented in Sect. 2. Then, Sect. 3 states the hypothesis and research objectives. Section 4 identifies information sources for measuring difficulty, states desired properties and limitations, proposes a mathematical definition and analyzes its advantages. Section 5 shows how to use proposed definition in an actual learning system. Finally, conclusions and future work are presented in Sect. 6.

## 2 Background

To understand the relation between difficulty, skills and learning, let us focus on the notion of *Flow Channel* (Fig. 1) [8, 26]. The Flow Channel represents the way



**Fig. 1** The Flow Channel (Csíkszentmihályi [3])

difficulty and skills of the learner relate to each other, as follows:

- When difficulty is much higher than learners' skills, anxiety appears. This is psychologically explained by learners perceiving their skills as insufficient, thus getting demotivated. They normally feel that the activity requires too much effort compared to their perceived capabilities. This often leads to early abandon;
- On the contrary, if learners' skills already include what the activity provides as learning outcome, boredom shows up. Having to invest time and/or resources to get an already possessed outcome is interpreted as lost time. Interest vanishes, motivation decreases and boredom appears;
- When skills and difficulty are balanced, learners enter a state of Flow. In Schell words [26], Flow is sometimes defined as *a feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment*.

This research assumes The Flow Channel theory as key for designing an adaptive learning system. Difficulty of the activities is adapted to match students' skills. The following sections present some relevant works on adaptive learning and its relation to difficulty and students' skills.

### 2.1 Adaptive learning

Adaptive learning is a research area whose aim is to improve learning by adapting contents to learners' needs. Adaptation is usually automatically performed by computers. This way, results may be scaled up to a virtually infinite number of students. This is becoming increasingly important as E-learning is growing and Massively Online Open Courses require improved ways of addressing heterogenous students' needs.

Most systems are based on the premise that there are different learning styles. They first take measures to infer students' learning styles, and then they adapt to each student. For instance, Yang et al. [29] propose a system that adapts user interface and content based on students' learning and cognitive styles. Cognitive styles are measured through standard tests that students fill in advance. Yang et al. defined Mental Load as a result of interactions between learning tasks, learning content and content characteristics. Mental Load could be considered another way to measure difficulty. They measured students' Mental Load and concluded that it was significantly decreased, while students' belief of learning gains was increased.

Another interesting work is the one of Sangineto et al. [25], The Diogene Platform. The platform performs automatic generation and personalization of courses. Diogene

gathers statistical information from users and constructs student models that include their skills and preferences. Diogene also takes into account current pedagogical knowledge on the didactic domain. These models are matched against learning objects that also have pedagogical descriptions. The match is performed using the pedagogical approach proposed by Felder and Silverman [5].

UZWEBMAT is an intelligent e-learning environment to teach probability [18]. UZWEBMAT uses an integrated expert system to determine learning styles and present most appropriate content. The system also takes into account performance and knowledge levels: different students with the same learning style may be subjected to different instructions. Learning styles are constantly reevaluated as learners interact. Initially, learning styles are determined using standard tests. The system considers three categories associated to the learning styles: visual, auditory and kinesthetic.

Protus [13] is a programming tutoring system that adapts to the interests and knowledge levels of learners. Protus identifies patterns of learning style and habits by mining learners' interaction logs. First, it assigns learners to clusters based on their learning styles. Next, it analyzes habits and interests of learners by mining frequent sequences. Finally, it completes personalized recommendation of learning content according to the ratings of frequent sequences. It also introduces a collaborative filtering system that predicts usefulness rating for a learner using other learners' ratings. Prediction is generated as a weighted average of other learners' ratings. The recommender system uses these predictions to select sequences for learners.

Computer Games is an important field regarding adaptation. Solano et al. [27] analyze learning styles in Game-Based Learning, particularly the fluctuation of the learning style during the learning process. Before the experiment, participants were asked to answer a standard learning style questionnaire. During the game, interaction between participants and the game was recorded automatically to identify the participant's learning style. Results showed that learning styles detected by the questionnaire were not always consistent with those detected during gameplay. It was also shown that learning styles varied during gameplay.

Finally, Sampayo-Vargas et al. [24] use adaptive educational games to study the behavior of learners. They use an adaptive version of a previously developed game. Actions to be performed in the game are classified by difficulty and curriculum category. When the student provides three consecutive correct responses for the same curriculum category, it is considered as learnt. The game increases difficulty level when all the curriculum categories are learnt. Conversely, three incorrect responses decrease the difficulty level. This way, the difficulty is adapted.

They state that difficulty refers to the effort required to overcome challenges presented by learning activities. That relation between difficulty and effort is a key point of this work, even though difficulty of the activities is manually assigned by game designers instead of induced from data.

Most works center their efforts on adapting learner paths to learning styles. They focus on delivering content depending on learning styles, habits and/or preferences. However, there are few examples that consider the difficulty of the activities as a key element of the adaptation process. This may be due to the inherent complexity of measuring difficulty. The following section presents selected works that deal with this complexity.

## 2.2 Difficulty

Difficulty is quite a diffuse concept referring to something that is laborious, not easy to do or understand, which requires an effort to be accomplished [17]. Some research work has been carried out on difficulty calibration by analyzing student historical data [21], or using linear regression to estimate difficulty based on user data [2] or even on generating exercises automatically with a given established difficulty [20, 22]. However, these studies are spread, discontinued and seem to be disconnected from each other. In general, the concept of difficulty within the academic world does not seem to capture too much attention.

More studies related to difficulty can be found changing the focus to the field of Computer Games. The parallelism with academic learning is quite straightforward: if a level of a game is too difficult or too easy, players tend to stop playing the game. Therefore, it is vital for a game to have a well designed progression of difficulty, if willing to catch the attention of the players. Most studies in this field try to develop methods to dynamically adjust difficulty to match the player's skills [10, 11, 14, 15]. All these studies use existent levels of difficulty proposed in present Computer Games and focus on selecting the most appropriate for each player and game being played. Hunicke and Chapman [10, 11] take measures of performance of the player and try to predict if the player is going to fail to anticipate and adjust the level of difficulty. The proposal is completely specific to First Person Shooter (FPS) games [23], as measures are defined for this specific type of gameplay. Mladenov and Missura [15] use data collected from previously played games to analyse a set of gameplay characteristics and input this data to a supervised Machine Learning algorithm. The goal is to have an offline prediction of the level of difficulty players are going to select in their next game. Missura and Gartner [14] take a different approach for automatically selecting difficulty for a given player among a finite set of difficulty levels. They divide

the game into play-review cycles. They measure the performance of the player in the play cycles and change difficulty level on review cycles accordingly to their estimations.

Herbrich et al. [9] present a very interesting work on measuring players' skills comparatively. Their system, called TrueSkill, is based on chess' Elo rating system [4]. Just like the Elo rating system, players have a one-dimensional value ranking that predicts their probability of winning against other players by logistic comparison. Although this work is not directly based on difficulty, it is indirectly valuing players' skill with similar intention: match players against those with similar abilities to foster balanced games.

Another interesting work is that proposed by Mourato and dos Santos [16]. Their goal is to procedurally generate content for Platform Games similar to Super Mario Bros [19]. The problem with this kind of content is how to classify the generated content with respect to difficulty. They propose a way to measure difficulty in Platform Games by measuring players' probability of failing after each individual obstacle in the game. The concepts are interesting though lack a practical result with actual players and ready-to-be-played generated content.

Finally, Aponte et al. [1] present one of the most interesting reviewed works. In their work they state that their goal is "to evaluate a parameter or a set of parameters that can be considered as a measure of a game difficulty". They start by measuring the difficulty of a reduced Pac-Man game with one ghost. In their Pac-Man game, speed of the ghost is a configurable parameter to make the game more difficult at will. They measure the score of a synthetic player as the number of eaten pellets and then show a graph with the evolution of this value depending on the speed of the ghost. This approach lets them show the progression of difficulty depending on the selected level (speed of the ghost). Based on that result, they define a set of properties that a general definition of difficulty should have, and propose a general theoretic definition of difficulty as the probability of losing at a given time  $t$ . They only propose this definition, however, do not perform any kind of test or mathematical proof. It ends up as a simple proposition based on their arguments.

All these previous works demonstrate the incipient interest of the research community for measuring difficulty. This trend is confirmed by the growing focus on measuring learning in general. The NMC Horizon Report: 2016 Higher Education Edition [12] states that there is a renewed interest in assessment and the wide variety of methods and tools to evaluate and measure the elements related to the learning process.

### 3 Hypothesis and research objectives

A review of previous research yields several interesting conclusions:

- There have been attempts to measure the difficulty of learning activities and learners' skills;
- Difficulty is related to effort demanded from learners to complete an activity;
- Balancing difficulty and learners' skills greatly improves the probability of learners staying within the flow channel;
- This research field is in its preliminary stages.

Potential improvements shall be expected when learners are kept inside the flow channel. To prove it, objective measures for difficulty and learners' abilities are required. Only proper measures can guarantee that their matching is meaningful. In order to get an accurate match, the more expressive the measures are, the better. Moreover, more expressive measures may give new insights into the learning process. Present measures for difficulty are limited in their expressiveness, mainly due to being one-dimensional. That raises some questions: could difficulty be redefined in a multidimensional space? And then, in which ways would it improve present definitions? Which limitations would it have? Could it be applied in practice?

To answer these research questions, the first step is to state a hypothesis that could be tested:

*Difficulty can be measured in a multidimensional space, improving its expressiveness and accuracy.*

Although there are several measures to consider, this work is focused on difficulty, and so does the hypothesis. To validate this hypothesis, an empirical methodology will be followed. This methodology will proceed through three objectives:

1. Propose a new definition of difficulty;
2. Provide an objective way to measure it;
3. Test the proposal with a practical case.

Section 4 will cover the first two objectives. It will start by selecting an appropriate source for measuring difficulty. Then it will proceed to analyze desired properties of the difficulty measure. This analytical process should establish a solid link between desired definition and the actual phenomena of difficulty. Intrinsic limitations derived from source and properties will be considered. This will thoroughly describe the framework for the new definition. Finally, the definition will be proposed mathematically. The proposal will be theoretically analyzed to set solid understanding.

This new two-dimensional definition of difficulty will also be shown graphically. Graphs will improve

understanding of the proposed definition. Then bases for deep and accurate analysis of graphs will be considered. There are many technicalities to take into account for proper understanding and comparison between graphs. This will also illustrate the potential gain of using a two-dimensional definition.

Section 5 will introduce the PLMan learning system [28]. This will be used to empirically test the proposed definition, addressing the third objective. After introducing and thoroughly explaining the PLMan learning system, the definition will be adapted to measure difficulty in the PLMan game. This will also provide insight on how to connect the theoretical definition with a practical situation. Finally, some real outcomes from the PLMan learning system will be analyzed graphically. This will give empirical evidence on how the new definition improves previously existing ones.

## 4 Defining and measuring difficulty

### 4.1 Sources for measuring difficulty

Let us consider difficulty as a cost: in order to successfully finish an activity, any learner has to pay a cost in time and effort. Measuring time is trivial from a conceptual point of view. The problem comes from measuring effort. How can we measure effort? Do we have an objective definition of what effort is?

It will be considered that effort is indirectly related to progress. The more progress is achieved, the less effort is required to finish. Although this logic consideration is not a concrete definition of effort, it has many advantages:

- For many kinds of activity, progress is relatively easy to define and measure objectively;
- A measure for progress is also closely related to learning outcomes: most activities yield learning outcomes even when not fully completed. In fact, these learning outcomes become clear when success ratio increases out of repeating the activity;
- As progress to success is one of the key factors in motivation, measures taking progress into account also foster motivation.

Therefore, this research will consider an activity “more difficult” when less progress is done. In the sake of rigor, progress will be considered with respect to time: progress percentage per unit of time will be an inverse measure for difficulty. So, an activity being “more difficult” will imply that less progress is made per time unit. This will let us measure difficulty in an intuitive, understandable and objectively measurable way.

### 4.2 Desired properties for difficulty

There are several ways of defining difficulty as a relationship between time and progress. It is important to have guidance for selecting an appropriate measure from such a huge set of potential definitions. So, establishing a set of desired properties will ensure that the selected definition is useful under defined criteria. These desired properties will act as restrictions, reducing the search space.

Let us consider the next set of properties, having present that measuring and comparing learning activities is the final goal:

- Difficulty should always be positive. Progress and time are always positive or 0 when measuring a learning activity. A negative difficulty coming out of these two values is impossible and would have no meaning;
- Difficulty should have a minimum value. A difficulty value of 0 would mean that no time/effort is required to finish a given activity. That would correspond to an activity that is already done;
- Difficulty should also have a maximum value. Making difficulty unbound would imply that any value could be “not so difficult” compared to infinite. Having a maximum value lets us fix impossible activities, which is desirable. An unbound upper limit that should be labeled as infinity makes formulation more complicated and has no advantage on comparisons;
- Fixing 1 as the maximum value for difficulty has advantageous properties. That bounds difficulty in the range  $[0, 1]$ , which lets us consider it as a probability. That makes sense and is compatible with previous considerations. Moreover, that enables the probability theory as a valid set of tools for working with difficulty, which is very desirable;
- Difficulty should not be a unique value but a function over time. While an activity is being done, difficulty keeps changing as progress is being made;
- Difficulty must be a continuous function over time. It makes no sense for a moment in time not to have a difficulty associated;
- Difficulty must be a non-strictly decreasing function. Every time a learner makes progress on a given learning activity, difficulty decreases by definition as less progress is required to meet success.

Let us consider an example of activity: “scoring five 3-point shots in a basketball court, in less than 5 minutes”. This is a training activity whose expected learning outcome is an improvement in shooting precision to basket.<sup>1</sup> This activity will take at most 5 min, and at least the time

<sup>1</sup> Although other learning outcomes can be considered from this activity, let us consider it just as a precision improvement exercise.



required to shot 5 times: time cost is straightforward. Regarding effort, it will depend on previous conditions. A trained, muscular player may complete the activity fast, without much effort, whereas a weak novice could require many attempts to finish it successfully. Moreover, novice players may waste much more energy because they lack adequate technique. This could also be considered more effort.

The activity could be analyzed many times and from different perspectives, and many definitions for “effort” could be found. Before entering an endless debate on what “effort” is or should be, let us consider a useful point of view with respect to our goal of measuring difficulty. An indirect measure for “effort” could be derived from the intrinsic failure/success measures of the activity. When 5 min are over, a player that scored 4 baskets is closer to success than other who only scored 1. It can be considered that having scored 4 baskets leaves out less progress to be done for succeeding than scoring just 1. Under this consideration, there is less effort pending to succeed when more percentage of the activity has been completed.

Let us compose a function with this properties, for the basketball example. Let us imagine a player that scores 5 baskets at times  $t_i \in \{15, 40, 62, 128, 175\}$ ,  $i \in \{1, 2, 3, 4, 5\}$  in seconds. Difficulty could be represented as shown in Fig. 2: whenever the player scores baskets, difficulty decreases. Decreasing difficulty can be considered as a step function, maintaining its value except on scoring events. It can also be considered as a linear function, resulting on a much smooth shape. Moreover, a linear function seems to inform better about the pace of the player.

As it can be deduced from Fig. 2, these properties configure a very powerful definition of difficulty: it goes far beyond a simple scalar quantity, defining a representative function. This function represents progress of the player over time which gives much more information about the process. This new information will also be useful for visual comparison of activity profiles as well as individual or group profiles.

### 4.3 Intrinsic limitations

The selected properties limit the way activities should be defined. Not every possible activity will fit for this model. This is both a limitation and a design guide. Activities designed for this model of difficulty will have the following set of properties:

- Activities require progress to be measurable (i.e., they should have a score). For instance, an activity defined as “selecting the proper answer from a set of 4” has no way of measuring progress. Although time to answer

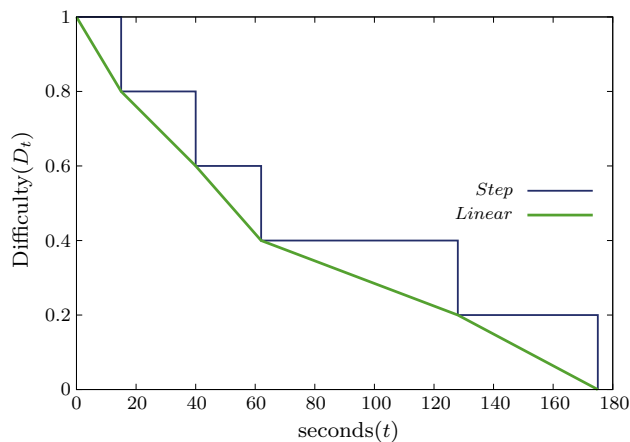
and success can be measured, there is no progress toward success. Resulting functions would represent either a full square or a line, depending on model selected;

- Score (i.e., progress) has to be non-strictly increasing function over time. As score is measuring progress to an end it does not make sense for it to decrease. General score measures having punishments or negative score events would not be appropriate. However, almost any score measure could be transformed in an equivalent non-strictly increasing measure for this purpose;
- Activities must have a measurable success status or, at least, a maximum score. This is required to define difficulty within its limits. Progress can be measured in unbounded activities, but cannot be scaled to a  $[0, 1]$  range;
- Activities must be considered over time. For instance, an activity about creating a program cannot be considered just as its final result. Having a single point of evaluation is similar to not being able to measure progress. It is also very important to measure the time required to do the activity. If all learners hand the result of an activity at the same time and no measures have been taken previously, no data will be available for the model.

These intrinsic limitations are part of the selected set of properties and ought to be assumed. They may represent a drawback for traditional activities such as questionnaires or written problems. These activities are not considered over time, nor is there a measure of their progress. However, these limitations may be also thought of as an opportunity to redesign and improve activities. Potential learning gains may be achieved. Having a progress measure informs learners and can improve their engagement. Moreover, a percent score yields more insight on the status of the learning process than a pass-or-fail measure. Therefore, adapting traditional activities to the limitations may represent an improvement in their learning potential.

While new designs for traditional activities are devised, some simple adaptations may be of help. For instance, a traditional questionnaire may be computerized and the score would easily be related to the number of right answers. Then, the exact evolution of the score over time may be considered. It should be considered that negative marks should not affect the percentual measure score: the one to be considered for difficulty. With this simple adaptation, it would be possible to use the proposed difficulty definition for a questionnaire.

Other traditional activities may be adapted in similar ways. For instance, a written problem may also be computerized and split into individually assessed sections. Progress then would also be measurable by the evolution of



**Fig. 2** Manually constructed difficulty function for basket example. Difficulty decreases as player progresses, scoring baskets in this example

marks over time. However, it would be better if problems could be redefined to have some sort of precision outcome.

For instance, let us suppose the problem of calculating the trajectory of a satellite to be launched into orbit. The problem could be translated into a simulation of the launch. The activity would consist in solving incidents during the launch process. Let us say that precision of the final orbit is affected by response from learners. They would have to calculate quick solutions for maintaining the optimal trajectory. The score would depend on the perfection of the trajectory. In a 5-min launch, a 3-s trajectory sampling would give a hundred samples. Awarding  $[0 - 1]$  points for trajectory perfection at every sample would give a final score in  $[0 - 100]$  points. As this score definition is cumulative, it is also non-strictly increasing. This final simulated problem would fit in the limitations and produce a continuous score. This example gives an idea on the improvements that may be achieved when fitting for the limitations.

#### 4.4 Mathematically defining difficulty

With all desired properties and limitations clarified, a working mathematical definition of difficulty can be constructed. Let  $A$  be the set of all possible activities, and  $L$  the set of all possible learners. Let  $\alpha \in A$  be a concrete learning activity. As an activity,  $\alpha$  can be performed by any learner  $l \in L$ . Each  $l$  performs  $\alpha$  a number of times  $N_l \in \mathbb{N}$ . So let  $\alpha_l^i, l \in L, i \in \mathbb{N}, i \leq N_l$  represent the  $i$ -th realization of the activity  $\alpha$  by the learner  $l$ .

Each  $\alpha_l^i$  takes an amount of time  $t_l^i \in \mathbb{R}$ , measured in seconds. Let us consider, for simplicity, that each  $\alpha_l^i$  starts at time 0 and ends at  $t_l^i$ . Then, let  $S_t(\alpha_l^i) \in \mathbb{R}$  be a function that measures the score got by learner  $l$ , at time  $t$  on its  $i$ -th realization of  $\alpha$ . So,  $S_t(\alpha_l^i)$  is the function that measures the

progress toward success of a learner that performs an activity.

The score function is expected to be explicitly defined for each activity. In fact, many different score functions can be defined for each activity. Therefore, let us assume that activities and their score functions are defined by activity designers. Also, for clarity reasons, let us assume that activities and score functions meet the desired properties and limitations exposed on Sects. 4.2 and 4.3.

In previous sections, difficulty has been defined as the inverse of progress. However, this cannot be defined exactly this way. Difficulty must be defined in a  $[0, 1]$  range, and the score function could have a much broader range. However, the score function should be non-strictly increasing, and should have an upper limit. Therefore, the score function could be safely assumed to start at 0, because the actual range of the function can always be moved to start at 0. Let  $S^*(\alpha)$  be the maximum score value for the activity  $\alpha$ ,

$$S^*(\alpha) \in \mathbb{R}, S^*(\alpha) \geq S_t(\alpha_l^i) \quad \forall l \in L, i \in N_l \quad (1)$$

This lets us define the “easiness function” as a scaled version of the score function over time in the  $[0, 1]$  range:

$$E_t(\alpha_l^i) = \frac{S_t(\alpha_l^i)}{S^*(\alpha)} \quad (2)$$

The function defined in Eq. 2 is called “easiness function” as it is exactly the inverse of the initial definition of difficulty. Therefore, the definition of difficulty follows:

$$D_t(\alpha_l^i) = 1 - E_t(\alpha_l^i) \quad (3)$$

This definition of difficulty is tied to the concept of progress. It represents an advantage over estimating difficulty with just a single scalar value: the resulting graph shows an evolution over time which informs of the whole realization of the activity. It also yields instant values for difficulty at any time of the realization. These values intrinsically represent the percentage of progress remaining to finish the activity. They could also be interpreted as the probability of failing the activity.<sup>2</sup>

However, these values are quite plain: they are instant values that do not capture information on the progress by themselves. The result is similar to considering any instant  $t$  to be independent from the others that compose the timeframe of the activity. For instance, this is like considering in the basketball example that scoring at first shot is equally probable to scoring after 4 baskets, or at a last attempt, when time is finishing. Nevertheless, a more accurate definition should consider that events occurring at

<sup>2</sup> This interpretation is bound to discussion about its real meaning as a probability.

time  $t$  are influenced by all events happened in the range  $[0, t]$ .

Experience shows that influence of a timeframe over next time steps is strong on humans. It is convenient to consider how human factors relate over time, i.e., psychological status, strength, fatigue, motivation, etc. Time steps in the timeframe of any learning activity, performed by a human learner, are best considered to be strongly interdependent. Therefore, they can be improved by making  $D_t$  depend on a function of all  $t' \in [0, t]$ , to make final values express this interdependency.

There are many approaches to make  $D_t$  dependent on the set of all past values of difficulty  $\{D_{t'}/t' \in [0, t]\}$ . Moreover, there is no theoretical way to determine the appropriate way to weight all the possible factors. What is more, different activities and learners will have different influence factors. This makes extremely difficult, if at all possible, to design a theoretical relation covering such a chaotic landscape. This suggests using an experimental approach instead. Therefore, this research starts modeling influence in a very simple way. This first model can be used as a benchmark to test other different approaches and experimentally determine better ways of defining difficulty.

Assuming that  $D_t, \forall t$  should depend on  $\{D_{t'}/t' \in [0, t]\}$  and  $0 \leq D_t \leq 1$ , let us define  $D_t$  as the area of the curve above  $E_t$  related to the maximum possible area up to the instant  $t$ ,

$$D_t(\alpha_i^j) = 1 - \frac{1}{t} \int_0^t E_t(\alpha_i^j) dt \quad (4)$$

Equation 4 defines difficulty  $D_t$  as a value depending on all previous history of the  $i$ -th realization of an activity  $\alpha$  by a learner  $l$ . The dependency is made indirect, using the easiness function as a proxy for difficulty. This makes definition easier, eliminating recursive references and associated problems.

Using the new definition stated at Eq. 4 the graphical layout of  $D_t$  varies greatly, as Fig. 3 shows. Compared to Fig. 2, the new definition for  $D_t$  results in a function that responds much smoothly to score events. This new behavior shows an interesting feature. Let us assume that  $t \in [0, t^*]$ . Using Eq. 4,  $D_{t^*}$  will directly depend on the performance shown by the learner during the realization of the activity (being  $D_{t^*} > 0$ <sup>3</sup>). In the basketball example, the faster baskets get scored, the lower  $D_{t^*}$  will be, and vice versa. Therefore, after completing an activity, the lower the residual difficulty value  $D_{t^*}$ , the greater the performance shown by the learner.

<sup>3</sup> Unless  $D_0 = 0$ , which would only happen on activities completed at start time. That is a degenerate case with no interest in practice. Thus, it can be safely ignored.

The interesting property shown by  $D_{t^*}$  is a direct consequence of its cumulative definition. So, this property will be shown by  $D_{t'}, \forall t' \in [0, t^*]$ . Therefore,  $D_t$  can now be used as a performance measure with more information than  $E_t$ , as it integrates information about score and time / frequency in one single value. Careful analysis of  $D_t$  for different learners and realizations of the same activity could lead to establishing correlations with abilities learnt and degree of mastery.

#### 4.5 Understanding easiness-difficulty graphs

Defining easiness and difficulty as a function of time yields a powerful analyzing tool. Resulting graphs (for example Figs. 3, 4) show learners' progress over time. Progress can show many different layouts, and a careful analysis gives insight into intrinsic characteristics of the activity.

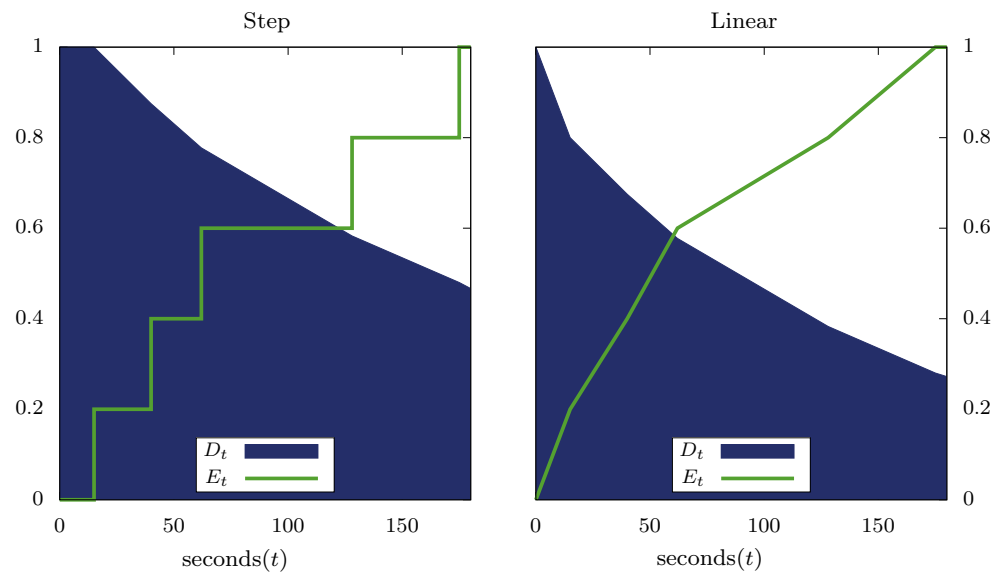
For instance, activities could present singularities like Activity X from Fig. 4.  $E_t$  captures the progress of a single learner that completes 30% of the activity in 50 s. Then the learner requires almost 200 s for completing another 10%. Finally, around 60 s are enough to complete the final 60% of the activity. This layout shows that the learner has struggled with some obstacle in the middle of the activity, while the rest of the activity has been straight forward. The graph shows the singularity and gives an insight into a potential hurdle in the middle of the activity.

$E_t$ - $D_t$  graphs are a powerful tool for analyzing activities. The information yielded can be used to design, classify or redesign activities to better adapt them to learners. It can also be used as evidence of learners' progression, degree of activity adaptation and hurdle detection. For these uses, the most important details to take into account are:

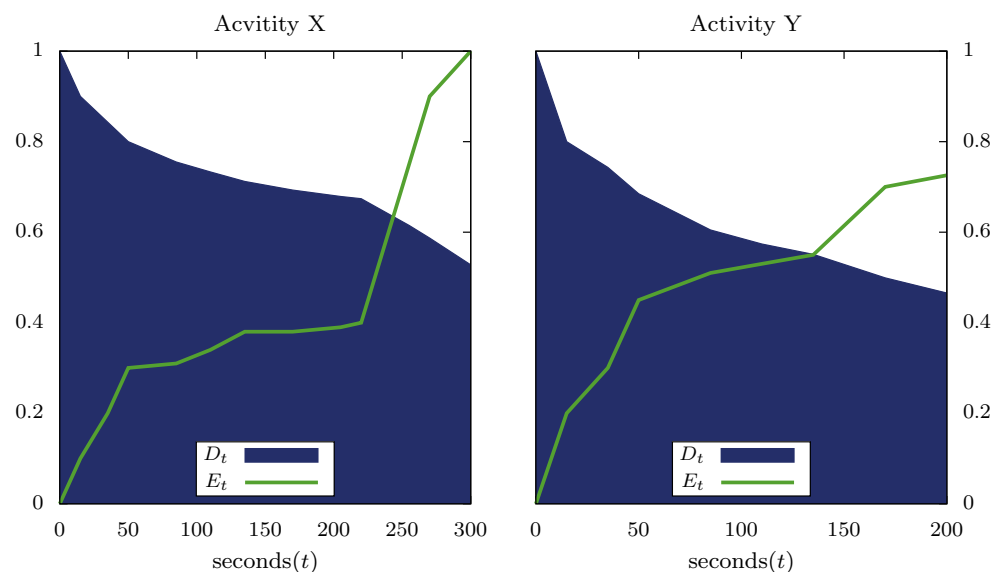
- *Data sources*: data can come from one single learner (Activity X in Fig. 4), or from an average of learners (Activity Y). Graphs show this difference: learner from Activity X achieved 100% at  $t = 300$ , while the group of learners from Activity Y had not passed 78% at  $t = 200$ . However, some of the learners from Activity Y could have achieved 100% at  $t = 200$ ; their 78% result is the average and sometimes could be confused with them all having the same value;
- *Time scale*: it is important to notice that Activity X and Activity Y have a different time scale. In order to properly compare difficulty and progress, similar time scales should be used. This is not always possible, since activities often have different completion times. Therefore, the time scale should be considered when analyzing graphs;
- *Singularities and interesting spots*: as already analyzed from Activity X, graphs show changes in their slope that may be taken into consideration. Pronounced



**Fig. 3** Behavior of  $D_t$  using Eq. 4 with data from the basketball example from Sect. 4.2. *Left*, exact definition for  $E_t$  with step value changes. *Right*, linear interpolation for  $E_t$



**Fig. 4**  $E_t$ – $D_t$  graphs for two activities. Activity X shows progress for a single learner while Activity Y averages a group of learners



changes will usually be due to characteristics of the activity or particulars of the execution. Also, the duration over time of this changes gives additional information on their relative relevance;

- *Cumulative nature of  $D_t$* :  $D_t$  is useful for single-point comparison in time. As  $D_t$  accumulates all past events in its instant value, it gives better instant estimation of difficulty. Comparing activities X and Y, their  $D_t$  values for  $t = 200$  are 0.68 and 0.46, respectively. These instant values inform that progress up to  $t = 200$  shows much resistance for Activity X than for Activity Y. In other words, Activity X has been more difficult up to  $t = 200$ .

Taking all these details into account, the analysis of  $E_t$ – $D_t$  graphs yields a great amount of information highly valuable for adaptation purposes.

More details and uses of this definition of difficulty are explained in our previous works [6, 7].

## 5 PLMan: a practical adaptive learning system

The PLMan learning system [28] is a custom-made automated adaptive learning system, which gives support to the first-year subject of computational logic. This work shows how the definition of difficulty can be used to make learning systems adaptive, using The PLMan learning system as example.

The PLMan learning system has two major components: a web application and a game. The web application implements the learning system itself and lets students and

teachers interact. Students access their progress status, select difficulty levels, get new activities assigned, upload their solutions, are automatically assessed and obtain their marks. Teachers can monitor students' progress, manage activities, analyze results and enhance students' assessments whenever required.

Activities in The PLMan learning system are based on a game called PLMan. PLMan is a custom-developed game aimed at teaching Logic Programming and Reasoning. PLMan is the core of the activities, and also the center of difficulty measures and adaptation. The following section briefly introduces the main details of PLMan. A more comprehensive description can be found in [28].

### 5.1 Learning activities: PLMan game

PLMan is a game that challenges students to solve some *Pac-Man*-like mazes by means of logic programming in Prolog language. Students control Mr. PLMan, a *Pac-Man*-like character whose aim is to eat all the dots in the maze. To control Mr. PLMan, students develop automated controllers (i.e., Prolog programs).

The automated controllers select the actions Mr. PLMan does during gameplay (see Fig. 5). These decisions have to dodge many different perils in order to eat all dots and succeed. Controllers are made of rules to reason about Mr. PLMan surroundings. These rules formalize students' reasonings, encoding patterns in the form *conditions*  $\rightarrow$  *action*. Final rules that solve a maze come at the end of a process by which students learn logic programming and reasoning.

The process is as follows: students write a minimum set of rules, try them executing the game, observe and analyse results, understand how rules produce results, modify their rules and start over again. This iterative process guides them into constructing the knowledge and abilities required to solve the mazes and advance to new stages. Figure 5 shows a maze along with a set of rules that guide Mr. PLMan to eat all the dots (i.e., solves the maze).

PLMan is turn-based. During each turn, Mr. PLMan is only allowed to perform one single action, which can be one of the following:

- `move(Direction)`: move one cell toward the direction.
- `get(Direction)`: get the object placed at the contiguous cell.

- `drop(Direction)`: drop the current object (reverse of `get`).
- `use(Direction)`: use the object toward one the direction.

For each selected action, an orthogonal *Direction* (up, down, left, right) must be specified. The game ends when Mr. PLMan succeeds (eats all the dots) or fails (it comes across an enemy or bomb, the limit of turns is reached or there is a time-out during execution).

### 5.2 Automated assessment with PLMan

PLMan is automatically assessed throughout its score. The score is defined as the percentage of dots that a controller eats, minus some punishments for incorrectly performed actions. This definition reflects the progress that students perceive as they develop their controllers. Score punishments are required to highlight incorrect actions like trying to move into a wall, trying to use an object not having one, trying to get an object where there is none or failing to select an action for a given situation (rule failure).

Each time students test a new controller by running PLMan, they get an automated assessment at the end of the execution. A detailed report is shown including final status, dots eaten (as percentage) and incorrectly performed actions. When students are satisfied with their present results, they submit the controller to the web application. The web application executes their controller internally and uses results to update student progress and marks. Each given maze has a value: when students achieve a 100% score in the maze, the value is added to their global marks. Also, when their score is less than 100%, they get the corresponding proportion of the value.

Students can send as many controllers for a given maze as they wish. Whenever students submit a controller that achieves more than 75% of the total score for a given maze, the next level is unlocked. Each level has a set of mazes, classified internally by difficulty. Initially, students select the difficulty they want for the new level, and the web application selects one maze among those available. The maze that gets assigned to a student cannot be assigned to another student in the same classroom. Therefore, in the same classroom, students have different mazes, even within the same level and difficulty.

**Fig. 5** Example maze and rules that control Mr. PLMan (@) to eat all the dots (.) dodging the enemy (E). # represent walls

```
#####
# ..... #
# ..... #
#@.....E.....#
# ..... #
# ..... #
#####

:- use_module('pl-man-game/main').
plman :- see(normal, up, '.'), doAction(move(up)).
plman :- see(normal, down, '.'), doAction(move(down)).
plman :- see(normal, left, '.'), doAction(move(left)).
plman :- see(normal, right, '.'), doAction(move(right)).
plman :- see(normal, down-left, 'E'), doAction(move(left)).
plman :- see(normal, down-left, '.'), doAction(move(down)).
plman :- doAction(move(right)).
```

In this progress scheme, difficulty is the key. If difficulty is well measured and matches students' abilities, students are placed in the channel of flow. Letting students select difficulty levels during the process is a step in this direction, that let students inform the system. However, any clever adaptation requires difficulty to be measured accurately. The following section shows how our definition of difficulty is applied to PLSMan.

### 5.3 Difficulty measurement in PLSMan

Initially, teachers design mazes and classify them in increasing levels of estimated difficulty. In the first mazes, simple rules in the form "If you see an enemy to your left, move right" are enough to construct successful controllers. As students progress and get more difficult mazes, higher-level programming constructs and reasonings are required. Teachers estimate the difficulty of newly added mazes by weighting the kind of programming constructs required and the projected cost in time for students to solve mazes. Although this may sound reasonable, actual difficulty for students may be very different.

When students develop controllers, the system logs all development progress. This progress includes all developed versions of the controller, their execution results, and the time required to develop them. All this information lets us construct a progression graph over time, using partial scores as main measure. If we only consider the maximum score achieved at every time, it is a valid easiness function ( $E_t$ ). Therefore, applying definitions of easiness and difficulty from Sect. 4.4 yields accurate difficulty measures.

After adding progress logged from different students for each maze, results show average difficulty. Figure 6 shows added results for maze 1–31. This maze was classified by teachers as level 1, difficulty 1 (out of 4 levels and 5 difficulties). Students solve mazes like 1–31 at the start of the

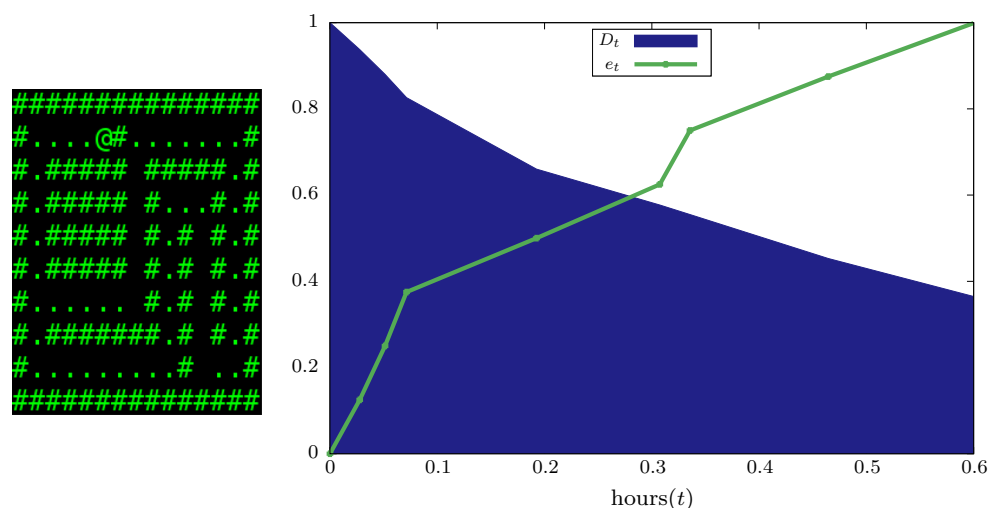
course. Figure 6 shows that all students solved 100% of this maze, taking 0.6 h at most. Their progress was more or less linear, as  $E_t$  shows. This means that there are no apparent hidden problems while solving the maze.

A little bit more complex example is shown in Fig. 7. This second example was classified as level 1, difficulty 5 by teachers: they needed 0.40 h on average to achieve 100% and estimated that students would take 4 times that. Again, all students achieve 100% score for this maze, but taking up to 5.63 h of work. Interestingly, students develop controllers that eat up to 76% of the dots rapidly (0.45 h) though after that, hours of work are required to find improvements to that.

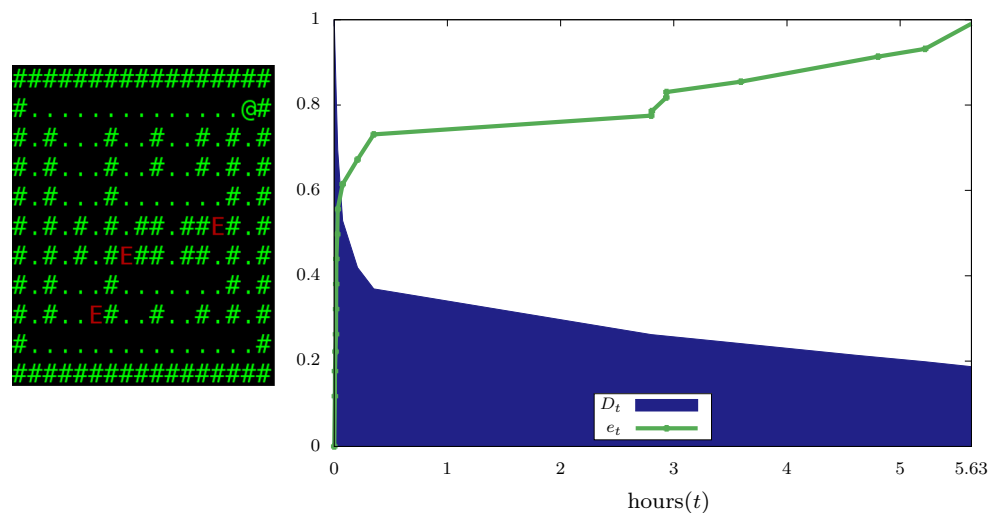
If we look at the structure of the maze, there are 3 enemies that move up to down. With many sets of simple rules to follow the dots, eating exterior and some interior dots is easy and fast. However, more interior dots are much more difficult to eat without coming across an enemy. Much more developed strategies and rules are required for that. Sometimes, even redesigning solutions from scratch to try following a different path are required. This explains the short period required for the initial 76% score, and the long period for the additional 14%. Explanations like this are not possible out of a standard unidimensional difficulty value.

Figure 8 shows measures taken for maze 2–48, classified as level 2, difficulty 2. In this case, teachers needed 0.33 h to achieve 100% and estimated that students would take between 4 and 5 times that. Real measures of difficulty show that not all the students achieved 100% score in this maze, and some of them took up to 4.44 h for their final controller. In this case, the failure of some students to achieve 100% is explained by the wall that blocks some dots, that has to be destroyed getting and using the white ball. Some students did not managed to get the ball, so they ate as much dots as possible, though not all.

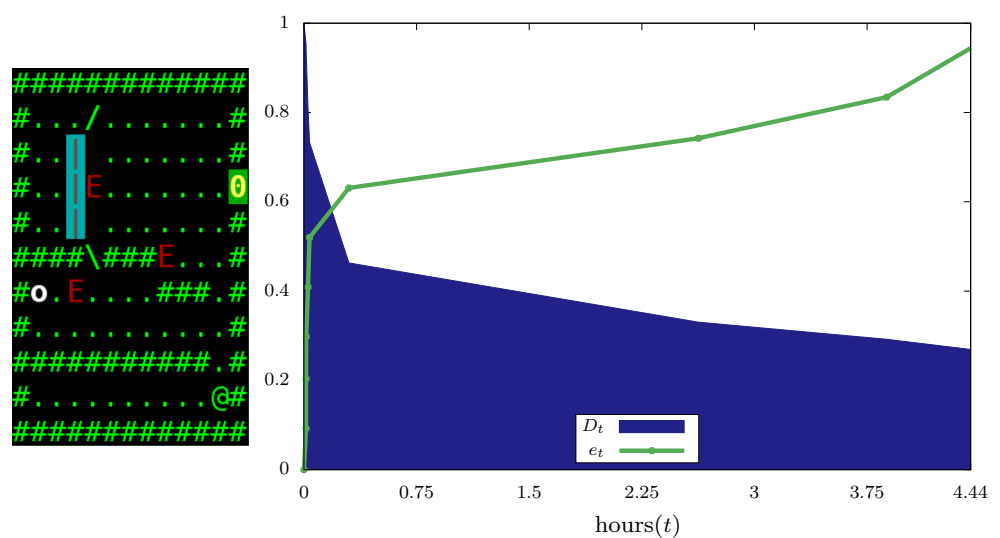
**Fig. 6** Maze 1–31, estimated as level 1, difficulty 1 by teachers. 16 students were assigned this maze and all achieved 100% marks after 0.6 h



**Fig. 7** Maze 1–59, estimated as level 1, difficulty 5 by teachers. 17 students were assigned this maze and all achieved 100% marks after 5.63 h



**Fig. 8** Maze 2–48, estimated as level 2, difficulty 2 by teachers. 18 students were assigned this maze and all achieved 94.5% marks after 4.44 h



It is also interesting to note the curve described by student progress. Students face a level 2 maze like this after solving three level-1 mazes. Therefore, they have experience and code from previous mazes when they start solving this one. That explains the ultra-fast 50–60% achievement. They use code from previous controllers for this maze and, without having to develop anything new, their code gets all that dots. However, solving the rest of the maze is much harder, as the  $E_t$  progress clearly shows.

#### 5.4 Making the PLMan learning system adaptable

The PLMan learning system includes more than 400 different mazes. From all these mazes, around 200 have enough data to add and generate representative graphs like those from Sect. 5.3. All these mazes have been graphed. These graphs have been and are being analyzed by teachers to better understand what the main problems and

difficulties are. This first step has led to most of these mazes to be re-classified.

This manual adaptation is also used as a first step for future automatic adaptation. Careful analysis yields better characterization of the features that describe the maps. This expert information will be matched against difficulty graphs to tag them and to train Machine Learning algorithms. Trained algorithms will be able to automatically tag mazes for potential problems directly out of graph information.

As a final step in the process, mazes will be clustered using difficulty graphs and generated tag information. These clusters will then be sorted into levels. Finally, whenever students choose their desired difficulty for a level, Machine Learning algorithms will learn to assign them the most promising mazes in that cluster, using students' progress information, clusters, difficulty graph information and generated tags.



## 6 Conclusions and further work

This paper stated as its hypothesis (Sect. 3) that there are better ways to define and measure the difficulty of learning activities. To prove this hypothesis, we proposed a new definition and a way to measure difficulty. We also used a practical case to illustrate the value of the proposal.

The proposed definition has been designed mathematically with a list of desired properties in mind. The definition relates difficulty to progress over time. Effort is modeled as the required time to achieve a specific score value. Difficulty can then be measured, graphed, analyzed and compared visually, yielding much new insights in the process. The proposed definition takes into account progress toward solving a learning activity, based on the score a learner achieves when performing the activity.

The proposed definition has intrinsic limitations: activities have to meet some requirements to be measurable. Activities must be performed and measured over time, and a score function is required to measure progress. The score function must have upper and lower boundaries and be non-strictly increasing: achieved score cannot be lost.

The proposed definition has also many interesting advantages. Being drawable, it can show progress over time. Graphs let teachers quickly and easily detect singularities and hurdles of learning activities, and also skills, problems and features of learners. Different parts of learning activities can be identified: most difficult parts produce valleys and easy parts produce pronounced slopes, all of them becoming measurable. Activities can be compared using their graphs, yielding a much accurate knowledge about which ones require more effort, and differences in the distribution of effort over time. These advantages make the proposed definition of difficulty a powerful tool for analyzing and comparing learning activities.

Although formal, the definition is also practical. PLMan Learning System has been used as a practical case to illustrate the use of the definition. The main element of the system is PLMan: an educational game to learn computational logic. PLMan is a *Pac-Man*-like game in which learners use Prolog programs to control the main character. The score depends mainly on the number of eaten dots. So, more difficult mazes are those that require a higher effort from learners to construct a Prolog program that eats the maximum number of dots. This score fulfills the required properties to apply the proposed definition of difficulty. After measuring difficulty, resulting graphs yield predicted rich interpretation of hurdles, singularities and features of the mazes. Some examples have been shown to illustrate

how easy and clearly graphs yield these and other insights into activities and learners' progress.

The final aim of the definition is the construction of an adaptive learning system that adjusts difficulty to the learners' skills. Present adaptations in the context of the PLMan learning system are made manually, however, show their value. Further work is required to automate these adaptations and to develop proposed and new ones. Next steps will aim to automate all these proposals inside the PLMan Learning System.

## References

1. Aponte, M.V., Levieux, G., Natkin, S.: Scaling the level of difficulty in single player video games. In: Natkin, S., Dupire, J. (eds.) *Entertainment Computing ICEC 2009, Lecture Notes in Computer Science*, vol. 5709, pp. 24–35. Springer, Berlin (2009). doi:[10.1007/978-3-642-04052-8\\_3](https://doi.org/10.1007/978-3-642-04052-8_3)
2. Cheng, I., Shen, R., Basu, A.: An algorithm for automatic difficulty level estimation of multimedia mathematical test items. In: *Eighth IEEE International Conference on Advanced Learning Technologies*, 2008. ICALT '08, pp. 175–179 (2008). doi:[10.1109/ICALT.2008.105](https://doi.org/10.1109/ICALT.2008.105)
3. Csíkszentmihályi, M.: *Flow: The Psychology of Optimal Experience*. Perennial Modern Classics. Harper & Row, New York (1990)
4. Elo, A.E.: *The Rating of Chessplayers, Past and Present*. Arco Publishing Company, New York (1978)
5. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Eng. Educ.* **78**(7), 674–681 (1988). <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf>
6. Gallego-Durán, F.J.: *Estimating difficulty of learning activities in design stages: a novel application of neuroevolution*. PhD thesis, University of Alicante (2015). <http://rua.ua.es/dspace/handle/10045/53697>
7. Gallego-Durán, F.J., Molina-Carmona, R., Llorens-Largo, F.: An approach to measuring the difficulty of learning activities. In: Zaphiris, P., Ioannou, A. (eds.) *Learning and Collaboration Technologies, LCT 2016, Lecture Notes in Computer Science*, vol. 9753, chapter 38, pp. 417–428. Springer Nature, Cham (2016). doi:[10.1007/978-3-319-39483-1\\_38](https://doi.org/10.1007/978-3-319-39483-1_38)
8. Getzels, J., Csíkszentmihályi, M.: *The Creative Vision: A Longitudinal Study of Problem Finding in Art*. Wiley, New York (1976)
9. Herbrich, R., Minka, T., Graepel, T.: Trueskill™: A Bayesian skill rating system. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 19, pp. 569–576. MIT Press, Cambridge, MA (2007)
10. Hunnicke, R.: The case for dynamic difficulty adjustment in games. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACM, New York, NY, USA, ACE '05, pp. 429–433 (2005). doi:[10.1145/1178477.1178573](https://doi.org/10.1145/1178477.1178573)
11. Hunnicke, R., Chapman, V.: Ai for dynamic difficulty adjustment in games. In: *Proceedings of AIIDE 2004* (2004). <http://www.cs.northwestern.edu/~hunnick/pubs/Hamlet.pdf>
12. Johnson, L., Adams Becker, S., Cummins, M., Estrada, V., Freeman, A., Hall, C.: *NMC Horizon Report: 2016 Higher*

- Education Edition. New Media Consortium, EDUCAUSE Learning Initiative, Austin Texas, [S.l.] (2016)
13. Klanja-Milievi, A., Vesin, B., Ivanovi, M., Budimac, Z.: E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Comput. Educ.* **56**(3), 885–899 (2011). doi:[10.1016/j.compedu.2010.11.001](https://doi.org/10.1016/j.compedu.2010.11.001)
  14. Missura, O., Gartner, T.: Predicting dynamic difficulty. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., pp. 2007–2015 (2011). <http://papers.nips.cc/paper/4302-predicting-dynamic-difficulty.pdf>
  15. Mladenov, M., Missura, O.: Offline learning for online difficulty prediction. In: *Workshop on Machine Learning and Games at ICML 2010* (2010)
  16. Mourato, F.J., dos Santos, M.P.: Measuring difficulty in platform videogames. In: *4a Conferencia Nacional em Interacao Pessoa-Mquina*, Grupo Portugues de Computacao Grafica/Eurographics, pp. 173–180 (2010)
  17. Nicholls, J.G., Miller, A.T.: The differentiation of the concepts of difficulty and ability. *Child Dev.* **54**(4), 951 (1983). doi:[10.2307/1129899](https://doi.org/10.2307/1129899)
  18. Özyurt, O., Özyurt, H., Baki, A., Güven, B.: Integration into mathematics classrooms of an adaptive and intelligent individualized e-learning environment: Implementation and evaluation of UZWEBMAT. *Comput. Human Behav.* **29**(3), 726–738 (2013). doi:[10.1016/j.chb.2012.11.013](https://doi.org/10.1016/j.chb.2012.11.013)
  19. Pedersen, C., Togelius, J., Yannakakis, G.N.: Modeling player experience in super mario bros. In: *2009 IEEE Symposium on Computational Intelligence and Games*, Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, USA, CIG'09, pp. 132–139 (2009). doi:[10.1109/cig.2009.5286482](https://doi.org/10.1109/cig.2009.5286482)
  20. Radošević, D., Orehovački, T., Stapić, Z.: Automatic on-line generation of student's exercises in teaching programming. In: Radošević, D., Orehovački, T., Stapić, Z. (eds.) "Automatic On-line Generation of Students Exercises in Teaching Programming", *Central European Conference on Information and Intelligent Systems*, CECIIS (2010)
  21. Ravi, G.A., Sosnovsky, S.: Exercise difficulty calibration based on student log mining. In: Mdrischer, F., Luengo, V., Lai-Chong Law, E., U, H. (eds.) *Proceedings of DAILE'13: Workshop on Data Analysis and Interpretation for Learning Environments*, Villard-de-Lans, France (2013). [http://www.dfki.de/~sosnovsky/papers/DAILE13\\_crv.pdf](http://www.dfki.de/~sosnovsky/papers/DAILE13_crv.pdf)
  22. Sadigh, D., Seshia, S.A., Gupta, M.: Automating exercise generation: a step towards meeting the MOOC challenge for embedded systems. In: *Proceedings of Workshop on Embedded Systems Education (WESE)* (2012)
  23. Saldana, J., Marfia, G., Rocchetti, M.: First person shooters on the road: Leveraging on aps and vanets for a quality gaming experience. In: *2012 IFIP Wireless Days*. IEEE, pp. 1–6 (2012). doi:[10.1109/wd.2012.6402812](https://doi.org/10.1109/wd.2012.6402812)
  24. Sampayo-Vargas, S., Cope, C.J., He, Z., Byrne, G.J.: The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Comput. Educ.* **69**, 452–462 (2013). doi:[10.1016/j.compedu.2013.07.004](https://doi.org/10.1016/j.compedu.2013.07.004)
  25. Sangineto, E., Capuano, N., Gaeta, M., Micarelli, A.: Adaptive course generation through learning styles representation. *Univ. Access Inf. Soc.* **7**(1–2), 1–23 (2007). doi:[10.1007/s10209-007-0101-0](https://doi.org/10.1007/s10209-007-0101-0)
  26. Schell, J.: *The Art of Game Design: A Book of Lenses*. CRC Press, San Francisco, CA, USA (2008)
  27. Soflano, M., Connolly, T.M., Hainey, T.: Learning style analysis in adaptive GBL application to teach SQL. *Comput. Educ.* **86**, 105–119 (2015). doi:[10.1016/j.compedu.2015.02.009](https://doi.org/10.1016/j.compedu.2015.02.009)
  28. Villagrà-Arnedo, C., Gallego-Durán, F.J., Molina-Carmona, R., Llorens-Largo, F.: PLMan: towards a gamified learning system. In: Zaphiris, P., Ioannou, A. (eds.) *Learning and Collaboration Technologies, LCT 2016, Lecture Notes in Computer Science*, vol. 9753, chapter 8, pp. 82–93. Springer Nature, Cham (2016). doi:[10.1007/978-3-319-39483-1\\_8](https://doi.org/10.1007/978-3-319-39483-1_8)
  29. Yang, T.C., Hwang, G.J., Yang, S.J.H.: Development of an adaptive learning system with multiple perspectives based on students' learning styles and cognitive styles. *Educ. Technol. Soc.* **16**(4), 185–200 (2013). <https://www.learntechlib.org/p/131570>