Adaptive Educational Games Using Game Metrics

Nabila Hamdaoui(✉), Mohammed Khalidi Idrissi, and Samir Bennani

LRIE Laboratory, Computer Science Department, RIME Team Mohammadia School Engineers (EMI), Mohammed Vth University Agdal, AV. Ibn Sina Agdal, 765 Rabat, Morocco
nabilahamdaoui@gmail.com, khalidi@emi.ac.ma

Abstract. Video games are ostensibly adopted in the education field thanks to their engaging, immersive and adaptive capacities. The greatest problematic in educational games design is how to create a ludic and adaptive experience without neglecting the learning objectives. To create an adaptive educational game modeling the player/learner is a must. In fact, determining the playing/learning style and measuring the player's abilities and performances will help in adjusting the game content and difficulty. The aim of this paper is to explain how the Adaptive Mechanism for Educational Games (AMEG) adapts the game using metrics. We will define the kind of metrics that the mechanism collects and how they are used to define the learning/playing style as well as their importance in gameplay and learning content adaptation.

1 Introduction

To ensure an effective learning experience, learners must feel that what they learn is relevant and meaningful for them. Educational games have aroused great interest in e-learning thanks to their adaptive and engaging capacities. A good educational game should bear the capacity to adapt to the player's/learner's preferences in terms of game-play as well as his/her learning objectives. As a previous work [1] we proposed an adaptive mechanism for educational games that would not only adapt the gameplay but also the learning content, based on the learning/playing style of the learner. In this paper, we will explain how and why the mechanism collects properties and metrics, while the player is playing the game.

In video games, players are seen as clusters of data. Players interact with the game continuously. Throughout the actions they take and the decisions they make, the game system generates and accumulates a lot of data susceptible of being used to understand the player's behaviors, needs and preferences. Such data is collected as quantitative variables that are called metrics. The recorded metrics can be used to define the player's playing and learning style. By modeling the learner, the game can be adapted. In fact, the gameplay can be adapted taking into consideration the player's progress through the game and his/her preferences. In parallel, the learning content will also be adapted using the values of the properties that have been expressed by the pedagogues and that have been filled during the game, thus, ensuring an effective and adaptive learning experience.

A. Abraham et al. (eds.), *Proceedings of the Third International Afro-European Conference for Industrial Advancement—AECIA 2016*, Advances in Intelligent Systems and Computing 565, DOI 10.1007/978-3-319-60834-1_21

In this paper we will firstly present the adaptive mechanism AMEG, its architecture and the kind of metrics it is susceptible to collect. Then, we will compare Kolb's learning style with Bartle's playing style to highlight the correlation between the two taxonomies. In Sect. 3, we will present the different kinds of metrics that can be collected in a video game and explain how they can be used to determine the learner's persona. In Sect. 4, we will see how game metrics can be used to adapt the gameplay and the learning content and, finally, we will conclude our work and present our future works.

2 AMEG: Adaptive Mechanism for Educational Games

To ensure efficient learning experiences, the learning content has to adapt to the learner's profile. Educational games should be able to adapt the gameplay to ensure entertainment and the learning content to guarantee efficient learning.

2.1 Presentation

In our previous work [1] we conceived an adaptive mechanism to adapt the learning content as well as the gameplay. The adaptive mechanism is constituted of two bricks: The first one is the AI brick; it contains the different AI algorithms that have been developed by the game developers' team and it is responsible for the adaptation of the gameplay. The second brick is the IMSLD (IMS learning design) brick; it is composed of conditions predefined by the pedagogues and is in charge of the learning content adaptation. The mechanism communicates with a database that contains all the learning activities that have been defined by the pedagogues, as well as a set of personas that represent the learners, their preferences, performance and knowledge. The mechanism collects different metrics and properties while the learner/player is involved in playing the game and then uses them to model the learner or fit him/her under an existing persona that matches his/her playing and learning style. We categorize those metrics in three groups: the first one are generic data that concern the learner such as his first name, last name, age and gender. The second one encloses the metrics that have been defined by the game design team, like the player session duration, his/her interaction with other players, levels completion and so forth. These metrics will help not only to determine the player's playing style but also reflect his/her learning style. The third group includes elements specified by the pedagogues; it consists of IMSLD properties that are in charge of sequencing learning activities, and monitoring the learner's knowledge and progression during the game. The collected metrics will help to define the learner's learning style and playing style.

Before adapting the game, the mechanism checks if the current learner fits under an existing persona; if it is the case, it will adapt the game by displaying the right learning activities and the right gameplay. If none of the existing personas matches the learner, then the mechanism will use the collected metrics to conceive a new persona before adapting the game. Thus, even if the playing/learning style of the learner changes through the game, the mechanism will ensure an adaptive experience. In AMEG, we propose the use of a common first level, a sort of short stage that will expose different learning activities with different difficulty levels as well as different kinds of gameplay. The collected data through this stage will help to determine the learner's knowledge as well as his/her preferences in term of gameplay and, thus, modeling his/her playing/ learning style.

2.2 Playing Style vs. Learning Style

People learn differently; they have different needs, strengths and weaknesses for the same learning content. By identifying the learner's learning style, it is possible to determine the methods by which s/he prefers to receive, assimilate and organize information. A significant number of categorizations of learning styles exist in the literature. Many researchers believe that learning styles are not determined by inherited traits; but can change over time and they are developed through experience. Among these researchers we cite Kolb whose model of learning styles is based on experiential learning, the kind of learning that is found in educational games.

In fact, educational games present quite an array of problem solving activities; each level exposes new challenges that the player needs to solve to become an expert [2]. Kolb's experiential learning style theory is built on the concept of a learning cycle. Such a cycle is made of four stages: Concrete Experience, Abstract Conceptualization Reflective Observation and Active Experimentation [3]. Experiential learning progresses through these four stages; the learner reflects on an experience to constitute concepts that are used in an active experimentation to test hypotheses and can serve as guides to create new experiences. As a learner proceeds in an experience, s/he goes through these four stages in a unique order and time. Thus, s/he will have his/her own learning style. Learners tend to prefer one stage of the cycle more than others. Using these four stages, Kolb introduced four learning styles (see Table 1).

Table 1. Kolb's learning styles traits

Learning style	Traits
Divergent (CE/RO)	- Emotional and imaginative, interested in people - Prefer to work in group, like to gather information
Assimilative (AC/RO)	- Find it more important that a theory bears logical soundness than practical value - Want to have time to think things through - Less focused on people, more interested in ideas
Convergent (AC/AE)	- Find practical uses for ideas and theories - Have the ability to solve problems and make decisions based on finding solutions to questions or problems - Less focused on people
Accommodative (CE/AE)	- Involve in new and challenging experiences - Act on "gut" feelings rather than on logical analysis - Rely more on people for information than on their own logical analysis

People do not only learn differently; they also play differently. Many taxonomies of playing styles have been proposed these last decades. We have chosen Bartle's types [4] as they are considered the simplest, earliest and more referenced models of gamer types. In Bartle's, there are four kinds of players: Socializers, killers, achievers and explorers (see Table 2). A socializer player likes to interact with others; s/he is interested in communicating and having relations with others. In this case, the game environment must be a network of friendship. Killers are attracted by rewards and they usually feel great by killing/destroying elements in the game. For this kind of players the gameplay must offer high level of challenges with interesting rewards. An achiever is interested in rising in game's levels. An interesting game for him/her must expose different situations/challenges susceptible of allowing him/her to accumulate points and make fast progresses. Explorers like to discover every component and/or detail that the game has to offer such as its mechanics, short cuts and tricks. The ideal gameplay for them is the one that exposes all the internal machinations of the game.

Table 2. Playing styles traits

Playing style	Traits
Socializer	- Interested in people and what they have to say, look for relationships
Explorer	- Like to discover the world of the game and the logic behind it - Interested in knowledge seeking, less focused on people - Likes to understand the principal behind the game, it is considered as an award
Achiever	- Focuses on accumulating materials and gathering points - Work hard to level up, less focused on people
Killer	- Impose him/herself on others, is considered as manipulator and born negotiator, looks for competition

We have chosen those two taxonomies (Bartle's playing styles and Kolb's learning styles) not only because they are efficient and widely referenced, but also because we have noticed a great correlation between the two of them. A socializer's playing style corresponds to the divergent learning style, since the two of them are interested in people and look for relationships. An explorer player matches an assimilative learner; they are both interested in knowledge and try to find the logic behind a theory. The achiever playing style resembles the convergent learning style, they focus less on people and work hard to solve problems and find solutions. The killer type goes well with the accommodative type since they both look for challenges and competition; they rely more on others and negotiate for information and points. Considering this correlation between the two taxonomies and defining the player's playing style are susceptible of generating a more accurate idea of the player's learning style(s).

3 Modeling Users Using Games Metrics

To be able to monitor the player's performances and motivations, we need to collect metrics that will contain information about the player and that will be used to customize the gameplay to the player abilities and preferences.

3.1 Games Metrics

As the player interacts with the game environment, data is continuously generated and recorded as quantitative variables. These variables are known as games metrics. Metrics are used to understand the players, their needs and their preferences, as well as their interaction with the world of the game and with other players/characters. To be able to set metrics during the gameplay, we need to express the attribute data that correspond for each metric and find a way to turn them into variables/features [5]. There are three kinds of player's metrics: customer metrics, community metrics and gameplay metrics. The kind of metrics that we are interested in are gameplay metrics as they focus on the player's behavior while s/he interacts with the game and its components. They monitor the player's progression through the game, the completion rate of missions/achievements and the player's flow, to see if the player has understood the game's objectives and gameplay or if s/he is struggling and needs help. Gameplay metrics are the most important metrics to evaluate the gameplay and the user's experience (see Table 3 for examples).

Table 3. Examples of games metrics

Metric	Operationalization	Representation
Number of weapons/ ammunitions	A variable that increment when the player picks up a weapon	If the player likes to collect weapons, then he might be an achiever or a killer
Total playtime per player	The sum of the duration of all played levels	The player is immersed in the game if the value of this metric is high
Number of quests/missions completed	A counter that increments each time the player finishes a mission or solves a problem	If the player completed most of the given quests, then he is an achiever and s/he has understood the game mechanics
Location of the player per time	A hash that stores the time spent per location	If the location of the player is changing constantly, then he might be an explorer
Health of the player	It increases each time the player picks up a power up and decreases when his character faces damages	The health of the player determines if the player is having a hard time. Thus, the game should adapt the difficulty level
Interaction with other characters	A hash that stores the sort of interaction that the playing is having with other characters and its duration	If the player has a sort of empathy relationship with other characters, then he is a socializer

Many researches have been conducted to prove how gameplay metrics customize and adapt the gameplay to the player preferences and abilities. Robin Hunicke and Vernell Chapman [6] proposed a dynamic difficulty adjustment system based on the monitored game data to create flexible experiences that adjust to the fly. Kiel M Gilleade and Alan Dix [7] proposed a different approach where the frustration of the player is measured and used to identify problematic play and design adaptive video games. Games metrics can be specific to one type of game, in Role Playing Games (RPG), for instance. Useful games metrics to collect are: number of victories, number of completed quests/missions, NPCs (Non Playing Characters) interaction and so forth. Games metrics can also be specific to one game to monitor features that are considered important. The collection of games metrics can be continuous or triggered by an event or can be recorded frequently. An event is anything that can happen in the world of the game; it can be an item clicked, an NPC killed, a conversation with another character, the completion of a level or any other more or less important action.

3.2 Games Metrics to Define Personas

Choosing the metrics to monitor is not a trivial task, since they are important to obtain meaningful models of players/learners and to adapt the game. Monitoring every possible game metric is not a wise thing to do. Collecting thousands of variables is considered a time consuming and an expensive operation. In that sense, the game designers/developers need to select the metrics to collect carefully. To be able to define the player's/learner's persona, the game must model his/her behaviors. For that purpose four kinds of metrics can be used [8]. Navigation metrics are about the position of a character, they gives us an idea of the preferred corners of the player, by tracking his/her spatial coordinates constantly and by tracking the camera view as well as his/her movements and speed. Many metrics can fit under this category, we could cite the type of movement of the player (walking, running, flying, staying still, lying down…). Movement can be tracked as a function of time. For instance, if the player is staying still for a long time we can conduct that s/he is not interested in the game or s/he is having trouble to figure out what do next. If s/he is constantly running and walking throughout the game world, then we assume that s/he likes to explore and discover the components of the game, thus, s/he is an explorer type. Interaction metrics are very useful to determine what the player is interacting with and what kind of interaction s/he is having. To exemplify more, if the player is often interacting with other characters and players, we can assume that s/he is more likely to be a killer type or a socializer type. If s/he likes to push other characters and use his/her weapons, then s/he is more likely to be categorized as a killer. If s/he is talking and using the chat system (another kind of interaction where the player is interacting with a game object instead of a character) then s/he bears characteristics of a socializer. Narrative metrics are about how the player interacts with the story and the objectives of the game, like the completion time of a certain level or a mission. Interface metrics are about the use of the graphical interface of the game and its menus.

In Table 3 we have selected some very well known and used metrics in game development, for each metric we explain how and when it is recorded and what it tells us about the game and the playing style of the player. The interpretation of a metric can differ from one game to another depending on the area of concern. For example, the player/learner with the shortest playing session will be considered the best. Although, it is not always the case, especially in educational games since the game is about solving problems, completing learning objectives and competency improvement. In this case, finding the best solution is more important than finishing first. Game metrics give us a clear idea of the player's playing style and preferences in terms of gameplay and because there is a great correlation between the playing style and the learning style of the learner (see Sect. 2). They will also be very effective in determining the learning style of the learner.

4 Video Games Metrics and Adaptivity

To ensure an effective adaptivity that matches the player's abilities, the game should adapt each level (the gameplay of the level, NPCs behaviors…). Metrics measure the difficulty that the player is facing and, thus, they are crucial for the adaptivity process.

4.1 Games Metrics to Adapt the Gameplay

To ensure a ludic and engaging game experience, the game should be able to adapt to the player's abilities by avoiding him/her to get bored if the game is too easy or frustrating or if the game is too difficult. The process of automatically changing the difficulty of the game is called Dynamic Difficulty Adjustment (DDA). DDA creates a custom experience for each player ensuring a fun and an interesting experience. DDA can affect many elements of the game; we could cite NPCs by reducing/increasing their number, frequency, speed and power or the gameplay by taking off some parts that are considered too difficult for the player's skills. The player him/herself can change via DDA by increasing/decreasing the number of power ups or health potions and first aid kits that will help the player to restore his health points. In all different DDA systems, it is important to measure the difficulty the player is facing to be able to adapt the game. This measure can be fulfilled via a heuristic function that maps a game state into a value that determines the difficulty that the player is facing at a specific moment [9]. Metrics/ heuristics that can be used for that purpose are: The life points of the player, the number of achievements and failures, completion time of some task and any other metric that can help to determine the player's score.

There are different approaches in DDA; we could cite dynamic level content adjustment that is in charge of the adaptation of the level content and its objects like enemies, weapons, health pickups and any other object that the player interacts with. An important system that implements the adjustment of level content is Hamlet [10]. Hamlet is based on Valve's Half-Life game engine. It monitors games statistics based on predefined metrics and use them to adjust the game settings. Hamlet firstly assesses when the adjustment is necessary by defining, collecting and assessing data like the power of the player or the damage a player takes over time. Then, it will execute adjustment policies that combine adjustment actions with cost calculation to ensure effective adjustment. Calculating the cost of each action and modifying it to the player's performance ensure an effective adjustment. Another type of DDA is automatic level generation using PCG (Procedural Content Generation) technique. PCG is the process of producing algorithmically the game content (rules, items quests, characters, stories, weapons, maps, levels and so on). PCG can be used to generate the content while the player is playing. Thus it can be used to maximize the learning effects of an educational game or to create player-adaptive games that match the player's tastes and needs [11]. Among online video games that use PCG we cite Infinite Mario Bros where levels are created automatically for the player while s/he is playing the game offering a different experience for each player. In [12] authors took as a test bed a modified version of Markus Persson's Infinite Mario Bros, where the content generated is automatically adapted to the player. To fulfill their research, they collected three kinds of data: Controllable features of the game that are used for level generation, gameplay metrics and finally player experience which was through a questionnaire where the player was asked to report the preferred game for three emotional dimensions: Fun, challenge and frustration. Using this data, the gameplay features that predict the player's preferences were determined using single layer perceptrons (SLPs). Then, multi-layered perceptrons (MLPs) were trained to learn, given a set of data as an input (controllable features and gameplay characteristics); what a subjective experience would a player have. The MLPS decide using the collected data, the right level that matches the player's preferences; then, the level is generated and the process is repeated. Genetic algorithms technique is another approach used for dynamic AI adjustment. Intelligent agents or NPCs are usually built using machine learning and genetic algorithms. In [13] authors used online coevolution to speed up the learning process. In fact, they used pre-defined agents with good genetic features constructed either by offline training or manually as parents in the genetic operation, so that they bias the evolution.

4.2 Games Metrics to Adapt the Learning Content

AMEG uses three kinds of metrics. Gameplay and generic metrics are the game metrics we talked about in section three. They are defined by the game designers/developers and are used to determine the playing/learning style of the player/learner. They are also used to adapt the gameplay. This kind of metrics as well as the IMSLD properties can be used to adapt the learning content. Since metrics are measures of events that happen in the game, they can also be used to monitor the learner's knowledge, objectives and competences. Pedagogues will define the adequate metrics that will be used to adapt the learning content and they will communicate them to the game designers/developers who, then, will express the attribute data that match each metric. In [14] authors have identified three categories of educationally relevant assessment variables: General Trait Variables, General State Variables, and Situational-Specific Variables. General trait variables are about the learner abilities and capabilities. General state variables target variables like goal orientation, self-regulation, attitudes, interests and prior knowledge of subject matter. And finally Situational-Specific Variables are the result of the interaction of the player with the game such as the emotional state of the player, cognitive load and engagement. All these metrics help to model the learner and to adapt the learning content. In the adaptive mechanism, the IMSLD brick will use these metrics as input properties to select the adequate learning activities. As an e-learning specification [15], IMSLD provides a common notation to represent personalized Units of Learning (UoL). The latter constitute the smallest units to refer to any delimited pieces of education. IMSLD is constituted of three levels: level A touches upon all the vocabulary needed to build any unit of learning. Level B includes elements such as properties, conditions, calculations, global elements and monitoring services. It is in charge of learning content adaptation based on the preferences, the portfolio, the pre-knowledge, the educational needs, and the situational circumstances of learners [16]. Level C is associated with Notifications, and enables a means to trigger some new activities.

In Fig. 1 there is an example of a unit of learning that concerns mathematical addition operation. There are four properties: *Initial-Knowledge* that stores the initial knowledge of the learner, *Answer-L1* and *Answer-L2* properties store the answers of the given learning activities and *Problem-Solved* contains the total number of problems solved. Those properties are filled while the player is playing the game. The unit of learning proposes two different learning activities. Let's assume that the first learning activity is easier than the second one. If the initial knowledge of the learner is less than a specific 'VAR1' (a global property which value has been determined beforehand by the learning content designer) then the learning activity 1 will be displayed, since it is easier than the second one. If the learner's answer for the first learning activity is correct, then the initial knowledge of the learner will increase by the value of the variable 'VAR2' and the number of problem solved activities will increment. The same applies to the second learning activity. The *Initial-Knowledge* and the *Problem-Solved* properties are global metrics; which means that they will persist until the end of the game and their values will change through the game. The *Answer-L1* and *Answer-L2* properties are local, they will be used for assessment purposes and they will no longer be used outside the UoL. By combining IMSLD adaptive capacities with DDA, we presume we can ensure an adaptive learning experience. In fact, the adaptive mechanism will use the collected metrics to model the learner and to select the adequate learning activities then it will generate dynamically the right content (gameplay, learning content) that matches the player.

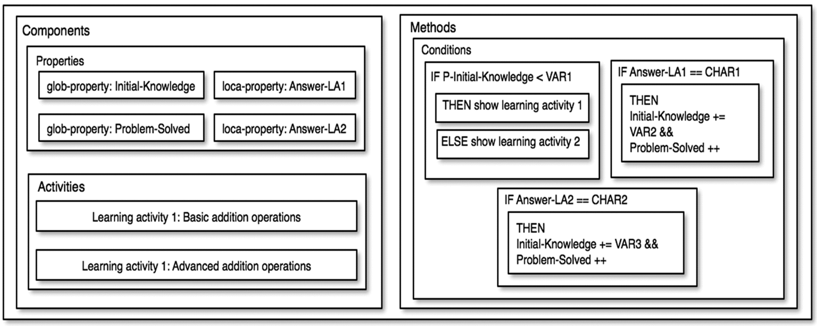

Fig. 1. Unit of learning: initiation to addition operations

5 Conclusion and Future Work

Good educational games offer an engaging immersive and an adaptive learning experience. While there are many approaches to adapt the gameplay in video games, a unified approach to adapt educational games to the learner abilities is still missing. In our previous work [1] we proposed a mechanism (AMEG) that collects data about the player and his/her interaction with the game objects and uses them to model him/her and adapt the game. In this paper, we have detailed the mechanism and the kinds of metrics that it uses. To adapt the game to the player's preferences, we need to know his/her learning/playing style. We have compared two taxonomies Bartle's playing styles with Kolb's learning styles and we have witnessed a great correlation between the two of them. Thus, determining the playing style of the learner will presumably help to determine his/her learning style. AMEG collects many metrics through the game; those metrics are used to determine the playing style of the learner and to select the correct gameplay and learning activities. In fact, to dynamically adapt the gameplay the mechanism needs to measure the difficulty that the learner/player is facing before performing the right DDA algorithm. The same applies to learning content adaptivity; the mechanism needs to assess the learner's abilities and knowledge before displaying the right learning activities. As a future work, we intend to conduct a survey to prove the correlation between the playing style and the learning style and make a prototype to apply this mechanism to an educational game.

References

1. Hamdaoui, N., Idriss, M.K., Bennani, S.: AMEG: adaptive mechanism for educational games based on IMSLD and artificial intelligence. In: 10th International Conference on Intelligent Systems: Theories and Applications (2015)
2. Gee, J.P.: What Video Games Have to Teach Us About Learning and Literacy. Palgrave/Macmillan, New York (2003)
3. The Kolb Learning Style Inventory—Version 3.1 Technical Specifications 2005
4. Bartle, R.A.: Hearts, clubs, diamonds, spades: players who suit MUDs. J. MUD Res. **1**, 19 (1996)
5. Drachen, A., El-Nasr, M.S., Canossa, A.: Game analytics—the basics. In: Game Analytics, pp. 13–40 (2013)
6. Hunicke, R., Chapman, V.: AI for Dynamic difficulty adjustment in games. In: Challenges in Game Artificial Intelligence AAAI Workshop (2004)
7. Gilleade, K., Dix, A.: Using frustration in the design of adaptive videogames. In: Advances in Computer Entertainment Technology, pp. 228–232. ACM Press (2004)
8. Tychsen, A., Canossa, A.: Defining personas in games using metrics. In: Conference on Future Play: Research, Play, Share, pp. 73–80 (2008)
9. Dynamic game difficulty balancing. https://en.wikipedia.org/wiki/Dynamic_game_difficulty_balancing
10. Hunicke, R., Chapman, V.: AI for Dynamic Difficulty Adjustment in Games
11. Shaker, N., Togelius J., Nelson, M.J.: Procedural Content Generation in Games: A Textbook and an Overview of Current Research, p. 3 (2015)

12. Shaker, N., Yannakakis, G., Togelius, J.: Towards automatic personalized content generation for platform games. In: Artificial Intelligence and Interactive Digital Entertainment (2010)
13. Demasi, P., Adriano, J.: On-line coevolution for action games. Int. J. Intell. Games Simul. **2**, 80–88 (2003)
14. Plass, J.L.: Metrics in simulations and games for learning. In: Game Analytics, Maximizing the Value of Player Data, pp. 697–729 (2013)
15. IMS Global Learning Consortium: Learning Design Specification (2003)
16. Mavroudi, A., Hadzilacos, T.: Implementation of Adaptive Learning Designs. Open University of Cypru