A Study of Adaptations and Support in Serious Games Dedicated to Learning Programming

Chaker Abid[1], Hedia Mhiri Sellami[2] and Lamjed Ben Said[3]
abidchaker@yahoo.fr[1], hedia.mhiri@isg.rnu.tn[2], bensaid_lamjed@yahoo.fr[3]
Laboratory of Strategies for Modeling and ARtificial intelligence
Institut Supérieur de Gestion de Tunis, Université de Tunis
Cité Bouchoucha 2000 Le Bardo,Tunis, Tunisia

***Abstract*— Learning programming presents many difficulties for learners, especially beginners, and even for teachers, who must find pedagogical alternatives to facilitate understanding of certain difficult concepts. To overcome these difficulties, some researchers have suggested focusing on teaching methods such as the use of serious games. The use of serious games in education can improve motivation, engagement and learning. Learners have different skills, learning styles or rhythms, and they don't learn in the same way. Games therefore need to take account of their profiles and offer them tasks adapted to their skills, as well as providing support when needed. In this article, we present an overview of the use of adaptive serious games for teaching programming over the last ten years. These studies have highlighted the need to adapt games to learners' profiles, and the need to offer them support to overcome their difficulties. This adaptation is achieved by machine learning algorithms. The study showed that fuzzy logic and the Bayesian network are the most widely algorithms used, and that text-based support is the most widely support used. Video simulation is the most appropriate support for reducing cognitive load.**

I. INTRODUCTION

Learning to program has become an important skill in the 21st century [1]. In many countries, it is already taught in elementary school and even nursery schools. Several studies have shown that learning programming presents many difficulties for learners, especially beginners, and even for teachers, who need to find teaching methods that make it easier to learn certain difficult concepts. Many studies [2], [3], [4] have shown that the failure rate of learners in this subject is very high. Also, [5] have shown that learners' levels of engagement in learning programming courses are very low. To overcome these difficulties, [6] proposed focusing on teaching methods such as the use of serious games. [7] have shown that serious games have attracted the interest of researchers. According to [8], [9] the use of serious games improves learner engagement, motivation and learning. [10] assert that learners have different skills, rhythms and learning styles. So, some high-skilled learners will be demotivated if they have to play all the levels, even those who have mastered the related concepts. Similarly, low-skilled learners need to repeat a game level as many times as necessary to assimilate the learning objective, otherwise they will abandon the game. The adaptation of a game to learners' profiles is achieved using artificial intelligence technique. The use of artificial intelligence techniques in learning, and game-based learning in particular, goes back a long way. With the evolution of this technique and the increased use of games in education, this use has reappeared.

[11] have shown that the use of artificial intelligence techniques in serious games takes two forms. On the one hand, it enables the monitoring of learners and the memorizing of their activities through the use of tools such as machine learning algorithms in order to provide feedback to learners to find out their weaknesses, to teachers to take certain actions or to designers and developers for re-engineering purposes. On the other hand, it also enables games to be adapted to learners' profiles and needs, and to offer them support in any difficulties. According to [1], adaptive educational games are a useful pedagogical alternative because of the intelligent tutoring systems they offer compared to non-adaptive games. They are based on differentiated pedagogy to provide personalized support for each learner. [12], [13] have shown that adaptation in serious games improves learner engagement and learning. [14] have shown that this adaptation takes three forms:

-Static: at the start of the game, using the data collected at the beginning.

-Dynamic: the game adapts during play by collecting data about the game. This adaptation is continuously updated.

-Hybrid: combining the two above.

[1] mentioned that the use of an adaptive serious game for learning programming is still in its infancy and has not taken full advantage of advances in artificial intelligence. That's why we've decided to address this topic. This paper presents a survey of adaptative serious games dedicated for learning programming during the last ten years.

In the remainder of this paper, the second section presents related work. In Section 3, we give an overview of the types of support used in the adaptation of serious games. In Section 4, we discuss the results of the research presented. Finally, we end with a conclusion.

II. RELATED WORK

This section reviews research on the use of adaptive serious games in learning programming. [1] have shown that the adaptation of serious games dedicated to learning programming is still in its early stages. We have therefore chosen to address this subject in this study. We performed a

literature search using the following keywords: programming learning, adaptive serious games and adaptive supports. This search generated a first list of games, from which we examined their abstracts, titles and keywords. We then selected a shortlist of 10 games relevant to our study, which we will examine in the remainder of this article.

In Table 1, that describe these games, [15] have adapted an existing game called Gidget [16] for learning programming to generate the adapted game GidgetML. The adaptation is performed using machine learning algorithms based on the learner's data. Players start practicing different tasks to generate enough data for adaptation. The game uses the data collected to create a model of the learner's profile, which is then used to adapt the game's tasks. The k-means algorithm uses the number of failures and the energy expended on each task to classify the player into one of three groups: low, medium and high, and then updates the player's model. The player receives tasks adapted to his skills based on his group. The adaptation strategy consists of identifying the required quantity of energy to solve a task, and reducing the limit of this quantity of energy for those with difficulties. In addition, the game accepts a solution containing one or two errors as a valid solution for struggling learners, whereas for highly skilled learners, the modification of a line of code is made only once. The k-means algorithm used in GidgetML has a few drawbacks, especially when extreme normal values are introduced into the data set, which can be a problem for small amounts of student data.

Another adaptive serious game called Nanodoc proposed by [1] equipped with a support to help teach sequences and iterations in programming to beginners. In this game, non-players help the players by giving them information about enemies, the procedure and mechanics of the game, using text messages. The adaptation is done during the programming phase. The adaptive support used is the example worked which is the resolution of an example similar to the problem posed. The game creates a model of the player's level of knowledge, which is updated with each solution presented. When this level of knowledge is insufficient to solve a puzzle, the game presents the player a worked example. By manipulating this example, the player acquires the knowledge necessary to solve the problem. The game uses fuzzy logic to identify the player's level of knowledge. It also uses a dynamic difficulty adjustment system (DDA) [17] to dynamically adjust the difficulty of the game based on the player's skills. The use of worked examples reduces the cognitive load on learners, which improves their learning. [17] have proposed an intelligent game-based learning environment (ENGAGE) with adaptive scaffolding to teach computer programming to high school students. The adaptive strategy consists of adaptively selecting the next task to be solved in the game and, also, automatically generating hints adaptively to guide learners step-by-step to solve a problem. Learners' knowledge is modeled using dynamic Bayesian networks based on their proposed solutions. Once a learner proposed a solution, the game environment identifies the concepts not yet mastered and proposes a new intermediate task between this current task and the task supposed to be next. When the student proposes a new solution, his model is dynamically updated. The game uses an automatic hint generation system to provide the student with hints to help him correct his solution. This system uses a data-driven solution space approach by building a solution graph. When the player encounters difficulties in a problem-solving task involving programming concepts, he can communicate with non-players to help him through hints, tips and animated images. The drawback of this game is that it has not been tested in teaching rooms to evaluate its effectiveness.

AutoThinking [12], [18] is an adaptive serious game for developing computational thinking. The adaptation concerns the game (story) and the content to be learned. To achieve this adaptation, the Bayesian network algorithm uses game data to assess the player's skills and, consequently, control the movement of enemies and the type of feedback to be sent to the player. The game provides the player with support in the form of text, images or video to help him improve his solution. There are two ways of adapting a solution before or after it has been completed. In the first case, the player executes his solution without debugging, receiving feedback or advice on errors and how to resolve them to get a correct solution. In the second case, the player executes the solution with the help of the debug button, in order to use the decision-making technique. He receives an estimate of the correctness of his solution in relation to the optimal solution for improvement.

[19] has developed an adaptive serious game called BOTS to teach programming to beginners. The player develops a code formed by a list of commands to move a robot around a 3D area. They can create their own puzzles and share them with their peers to solve, or they can solve their peers' puzzles. Those who propose puzzles offer hints for difficult puzzles to help solve them. Hints are generated automatically using the "hint factory" algorithm. The game's adaptation strategy is based on the creation of a network or graph of interactions using program output as the definition of states to provide clues and feedback to future students.

[20] has proposed an adaptive serious game entitled HTML Escape Game designed for learning the HTML language. The adaptation concerns the game scenario based on the players' knowledge levels. Four fuzzy sets have been created and the player will be classified according to his level of knowledge in one or two sets. These fuzzy sets are transformed into fuzzy states in a state diagram. The transitions between states describe the dynamic adaptation of the game and the learner's progress, enabling the next step to be chosen. They determine whether the scenario should be dynamically extended according to the player's progress. The game dynamically adapts its scenario by adding objects, new rooms, new non-player characters and, consequently, new exercises for the players.

[21] have developed an adaptive educational game to teach the basics of structured query language (SQL). Adaptation is performed at the start of the game and then dynamically during the game based on learners' interactions and according to their learning styles. Before starting the game, participants are asked to complete a questionnaire based on the Felder-Silverman learning style to identify their individual learning style. Players must complete certain missions by writing SQL queries. They receive support in the form of text, images or diagrams, depending on their learning style. During the game, participants' interactions with the game are automatically registered to identify their learning style, which may change from mission to mission depending on the solutions they

propose. At the end of the game, participants are asked to answer an SQL post-test to study the contribution of adaptation in educational games. The consistency of the test questions was measured using Cronbach's alpha test.

In order to study the effectiveness of adaptation, they compared learning without games to learning through games, and learning through non-adaptive games to adaptive games. The disadvantage of game-based learning is that the learner progresses through the game even if he or she fails to write the SQL query correctly after six attempts. In this case, the learner continues to play even without understanding the SQL queries. Furthermore, this approach does not take into account the problem-solving process, the pedagogical aspect or how learners use the game to solve the problem. It is only interested in the learners' tracks (the ends).

[22] proposed a game in which, when the system detects a player in difficulty when solving a puzzle, the player receives a phone call from a non-player person to give advice or instructions on how to solve the problem. The diagnosis of the player's weakness is based on the number of failed attempts or the time spent solving a puzzle. There are some limitations to this game, particularly in terms of the adaptive elements of the game, which are limited, and the adaptation aspect is not well explained.

[14] have proposed a game called Minerva of multiple genres (adventure, action, puzzle) to teach basic programming concepts to elementary school students. It deals with loops, inputs, outputs, the mathematical library as well as executing a piece of code. The adaptation is based on the game's style and the player's learning style. These two styles are identified at the start of the game using a questionnaire. Game adaptation consists of modifying the number of obstacles and non-player characters (NPCs) according to the player's style, as well as the choice of the next step. This adaptation changes throughout the game, depending on the player's actions. Adapting the learning style leads to modifying the learning content.

TABLE I. PROGRAMMING CONCEPTS STUDIED IN ADAPTIVE SERIOUS GAME.LE TYPE STYLES

References	**Game**	**Type of adaptation**	**Programming concepts**
[15]	GidgetML	Adapting an existing game (Gidget)	-debugging, -conditional and iterative structures,
[23]	Nanodoc	Creation of a new adapted game	-sequences -iterations
[14]	Minerva	Creation of a new adapted game	Inputs, Outputs iterative structures, sequential execution, math library
[24]	AutoThinking	Creation of a new adapted game	-conditional control structures quences, -le débogage
[17]	ENGAGE	Creation of a new adapted game	--conditional and iterative control structures
[19]	BOTS	Creation of a new adapted game	-Iterative structures (for, while,...) -Functions, The problem-solving approach
[21]	Not mentioned	Creation of a new adapted game	SQL language
[20]	HTML Escape Game	Creation of a new adapted game	Le langage HTML
[24]	Autothing	Creation of a new adapted game	-iterative and conditional structures, -sequences, debugging,
[22]	siret security Game	Creation of a new adapted game	

TABLE II. CRITERIA AND MACHINE LEARNING ALGORITHM USED IN ADAPTATION

References	**Machine learning algorithms**	**Adaptation criteria**
[15]	k-means	-the number of failures, -energy expended to solve a problem e.

[23]	Fuzzy logic	-player's level of knowledge
[14]	Questionnaire based on likert scale.	Learners' learning and playing styles
[24]	Bayesian networks	The player's skills (the proposed solution),
[17]	-Use of dynamic Bayesian networks. -graph construction: solution space	Student solution
[19]	-generation of a network of interactions (graphs or solution spaces)	-Student solution,
[21]	-Quiz, -tests	-player learning style
[20]	-fuzzy logic, -creation of a state graph	-Players' level of knowledge,
[24]	the Bayesian network algorithm,	
[22]		-the number of failed attempts, -the time spent solving a puzzle

III. ADAPTATION SUPPORTS

In educational games, research has shown that learners who have difficulties quickly abandon the game. This is why games need to offer them support to help them solve difficult tasks and motivate them. According to [1], support methods have been effective in teaching programming structures. Table 3 summarizes the different forms of support. We cite:

- Texts: these are instructions designed to provide students with explanations or diagnoses. They appear at specific points in the game or between levels. Another version is the support manual, which contains more detailed information.

-Hints or support tips: these are brief notifications designed to help the player find the right answer.

-Images: these are visual aids or diagrams that illustrate the learning content.

Videos: these are short videos offering additional explanations or demonstrations of the content to be learned.

TABLE III. ADAPTIVE SUPPORT IN ADAPTIVE SERIOUS GAME

References	**Adaptation support**	**element to adapt**	**Type of adaptation on**
[15]	-text	content adaptation	hybrid
[23]	worked examples	content adaptation	dynamic
[14]	image, text, video, tips	content and game adaptation	hybrid
[24]	Texts, Image, Video, indices	Adapting learning content and game play	dynamic
[17]	-text -animated images, - hints	-content adaptation. Adaptive -selection of next task -adaptive hints generation	dynamic
[19]	hints	Content adaptation	dynamic
[21]	Image, Text, Audio	content adaptation	hybrid
[20]	Audio (talking to NPCs)	Adapting the game scenario	dynamic

[24]	text, image, video	content adaptation	dynamic
[22]	-Tips, -texts	content adaptation	dynamic

IV. DISCUSSION

This study has highlighted the interest of adapting serious games. It keeps learners' attention and enables them to take full advantage of games as a powerful pedagogical tool. Adaptation is performed before or during the game, and concerns the pedagogical content, the interface and the non-player characters. Content adaptation is the most used. It is achieved using machine learning algorithms such as the Bayesian network and the fuzzy logic which are the most used. These algorithms uses several characteristics of palyer in order to modeling them. The study has shown that the time spent, number of failures and knowledge level are the most used characteristics. Many studies use a dynamic difficulty adjustment system (DDA) to dynamically adjust the difficulty of the game based on the player's skills.

The study also underlined the value of assisting the player. Textual support is the most widely used, as it is easy to implement and enables direct dialogue with the user. [1] showed that the use of video simulations containing an example presented solutions to situations similar to the task to be solved, thus reducing learners' cognitive load.

Some studies have shown that there are two ways of adapting a game: either by adapting an existing non-adaptive game, or by developing a new adaptive game. In the case of adapting an existing adaptive serious game, the researchers used a control group using the non-adaptive version to compare the two results.

V. CONCLUSION

In this paper, we presented a study on adaptation in serious games used to teach programming. Adaptation consists in making game intelligent using artificial intelligence techniques such as machine learning algorithms. Bayesian networks and fuzzy logic are the most commonly used algorithms for representing and predicting learners' knowledge. Adaptive games often provide players with support to help them overcome difficulties. This study has shown the interest of adapting educational games to learners' profiles, as it increases learners' motivation, engagement and learning compared to non-adaptive serious games.

REFERENCES

[1] P. Toukiloglou et S. Xinogalos, « Adaptive Support With Working Examples in Serious Games About Programming », *Journal of Educational Computing Research*, vol. 61, nº 4, p. 766-789, juill. 2023, doi: 10.1177/07356331231151393.

[2] J. Bennedsen et M. E. Caspersen, « Revealing the programming process », in *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, St. Louis Missouri USA: ACM, févr. 2005, p. 186-190. doi: 10.1145/1047344.1047413.

[3] Simon *et al.*, « Pass Rates in Introductory Programming and in other STEM Disciplines », in *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, Aberdeen Scotland Uk: ACM, déc. 2019, p. 53-71. doi: 10.1145/3344429.3372502.

[4] C. Watson et F. W. B. Li, « Failure rates in introductory programming revisited », in *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*, Uppsala, Sweden: ACM Press, 2014, p. 39-44. doi: 10.1145/2591708.2591749.

[5] M. Butler, J. Sinclair, M. Morgan, et S. Kalvala, « Comparing international indicators of student engagement for computer science », in *Proceedings of the Australasian Computer Science Week Multiconference*, 2016, p. 1-10.

[6] A. Luxton-Reilly *et al.*, « Introductory programming: a systematic literature review », in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, p. 55-106.

[7] M. A. Miljanovic et J. S. Bradbury, « Robobug: a serious game for learning debugging techniques », in *Proceedings of the 2017 acm conference on international computing education research*, 2017, p. 93-100.

[8] D. Hooshyar, M. Yousefi, M. Wang, et H. Lim, « A data-driven procedural-content-generation approach for educational games », *Journal of Computer Assisted Learning*, vol. 34, nº 6, p. 731-739, 2018.

[9] T. H. Laine et R. S. Lindberg, « Designing engaging games for education: A systematic literature review on game motivators and design principles », *IEEE Transactions on Learning Technologies*, vol. 13, nº 4, p. 804-821, 2020.

[10] R. S. Lindberg et T. H. Laine, « Detecting Play and Learning Styles for Adaptive Educational Games. », in *CSEDU (1)*, 2016, p. 181-189.

[11] W. S. Ravyse, A. Seugnet Blignaut, V. Leendertz, et A. Woolner, « Success factors for serious games to enhance learning: a systematic review », *Virtual Reality*, vol. 21, p. 31-58, 2017.

[12] D. Hooshyar, R. Binti Ahmad, M. Wang, M. Yousefi, M. Fathi, et H. Lim, « Development and evaluation of a game-based bayesian intelligent tutoring system for teaching programming », *Journal of Educational Computing Research*, vol. 56, nº 6, p. 775-801, 2018.

[13] S. Xinogalos, « Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy », *Educ Inf Technol*, vol. 21, nº 3, p. 559-588, mai 2016, doi: 10.1007/s10639-014-9341-9.

[14] R. S. Lindberg et T. H. Laine, « Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12 », *International Journal of Serious Games*, vol. 5, nº 2, p. 3-24, 2018.

[15] M. A. Miljanovic et J. S. Bradbury, « GidgetML: An adaptive serious game for enhancing first year programming labs », in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*, 2020, p. 184-192.

[16] M. J. Lee, « Gidget: An online debugging game for learning and engagement in computing education », in *2014 ieee symposium on visual languages and human-centric computing (vl/hcc)*, IEEE, 2014, p. 193-194.

[17] W. Min, B. Mott, et J. Lester, « Adaptive scaffolding in an intelligent game-based learning environment for computer science », in *Proceedings of the Workshop on AI-supported Education for Computer Science (AIEDCS) at the 12th International Conference on Intelligent Tutoring Systems*, 2014, p. 41-50.

[18] D. Hooshyar, H. Lim, M. Pedaste, K. Yang, M. Fathi, et Y. Yang, « AutoThinking: An Adaptive Computational Thinking Game », in *Innovative Technologies and Learning*, vol. 11937, L. Rønningsbakk, T.-T. Wu, F. E. Sandnes, et Y.-M. Huang, Éd., in Lecture Notes in Computer Science, vol. 11937. , Cham: Springer International Publishing, 2019, p. 381-391. doi: 10.1007/978-3-030-35343-8_41.

[19] A. Hicks, Y. Dong, R. Zhi, V. Cateté, et T. Barnes, « BOTS: Selecting Next-Steps from Player Traces in a Puzzle Game. », in *EDM (Workshops)*, 2015.

[20] K. Chrysafiadi, S. Papadimitriou, et M. Virvou, « Fuzzy states for dynamic adaptation of the plot of an educational game in relation to the learner's progress », in *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*, IEEE, 2020, p. 1-7.

[21] M. Soflano, T. M. Connolly, et T. Hainey, « An application of adaptive games-based learning based on learning style to teach SQL », *Computers & Education*, vol. 86, p. 192-211, 2015.

[22] C. D'Apice, C. Grieco, R. Piscopo, et L. Liscio, « Dms2015short-2: Advanced learning technologies for elearning in the enterprise: Design of an educational adventure game to teach computer security », *Journal of Visual Languages & Computing*, vol. 31, p. 260-266, 2015.

[23] P. Toukiloglou et S. Xinogalos, « NanoDoc: Designing an Adaptive Serious Game for Programming with Working Examples Support », in *European Conference on Games Based Learning*, 2022, p. 628-636.

[24] D. Hooshyar, L. Malva, Y. Yang, M. Pedaste, M. Wang, et H. Lim, « An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking », *Computers in Human Behavior*, vol. 114, p. 106575, 2021.