A Functional Model of a Designer of Educational Maze Game

Yavor, Y.I.D., Dankov*
Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski" Sofia, Bulgaria
yavor.dankov@fmi.uni-sofia.bg

ABSTRACT

Game-based learning and serious games are popular topics among researchers and practitioners, resulting in multiple studies, demonstrating the various benefits to the users. The paper is based on the application of the TIMED-VGE taxonomy (or *Taxonomy of Instruments for Management and Evaluation of the Design of Video Games for Education*) in the APOGEE (smArt adaPtive videO GamEs for Education) software platform. The platform provides an automatic generation of educational video maze games (based on formal maze game descriptions as XML documents), personalization capabilities and personalized content, and adaptive gameplay. According to the application of the TIMED-VGE taxonomy in the APOGEE platform, the software tools are divided into two main categories (Assistive instruments and Analytics Instruments), which represent the whole set of tools facilitating the processes of the design and creation of educational video games in the APOGEE platform. The taxonomy represents the hierarchical structure of the software tools and in each category, there is a certain set of tools. The focus of this paper is on the development of a functional model of the instrument *Maze Game Designer*, based on the application of the TIMED-VGE taxonomy. The Maze Game Designer is in the category "Design management" instruments, according to the taxonomy. The tool is key to starting the process of designing and creating educational games in the APOGEE platform. Therefore, the paper presents further development of the Maze Game Designer instrument and its functionalities, based on the core user functionalities of the tool, analyzed and developed in previous publications of the author. As a result, the paper presents an extended version of the user functionalities of the Maze Game Designer instrument, based on the application of the TIMED-VGE taxonomy in the APOGEE software platform. Based on these functionalities, the paper presents the developed two functional models of the tool and the main workflow of the Maze Game Designer instrument. This includes a Functional model of the tool and a Detailed functional model of the Edit Maze Game use case of the Maze Game Designer instrument. All these results will serve for the future development of the tool. The development of a key instrument of the APOGEE software platform, according to the application of the TIMED-VGE taxonomy, will contribute to increasing the designer's capabilities to use the platform's instruments for the design and creation of improved video games for education.

CCS CONCEPTS

• **Applied computing** → E-learning; Learning management systems; • **Information systems** → Data analytics; • **Software and its engineering** → Software design engineering.

KEYWORDS

Educational games, Maze Game Designer, TIMED-VGE taxonomy, Software instruments, APOGEE, Game design, Serious games

ACM Reference Format:
Yavor, Y.I.D., Dankov. 2021. A Functional Model of a Designer of Educational Maze Game. In *2021 4th International Conference on Computer Science and Software Engineering (CSSE 2021), October 22–24, 2021, Singapore, Singapore.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3494885.3494952

1 INTRODUCTION

Game-based learning and serious games are popular topics among researchers and practitioners [1, 2], resulting in multiple studies and an overall increase of interest in the topic [3, 4]. The benefits to users of educational video games have been proven in multiple research studies [5–7]. The APOGEE (smArt adaPtive videO GamEs for Education) [8] is a platform for the construction and generation of educational video maze games. The platform provides an automatic generation of educational video maze games (based on formal maze game descriptions as XML documents [9]), personalization capabilities and personalized content [10], and adaptive gameplay.

The development and integration of new capabilities to the platform is a task, requiring constant research, analysis, and development to meet the changing user requirements. The APOGEE software platform provides the users with educational video games reaching high levels of satisfaction from users of the platform. One of the main contributors to such a high level of user satisfaction and increase in the quality of the user experience within the game is the appropriately designed software instruments of the APOGEE platform.

One of the main goals of the *TIMED-VGE taxonomy* (or *Taxonomy of Instruments for Management and Evaluation of the Design of Video Games for Education*) is that it „can serve as a foundation starting point for developing better educational video games with personalized and high-value learning content combined with the adaptive video gameplay process and improved user-experience "[11]. The taxonomy and its application in the APOGEE software platform are described in detail in [11] and this taxonomy has been successfully applied to the design and development of the platform's tools. According to the application of the TIMED-VGE taxonomy in the APOGEE platform, the software tools are divided

*Assistant Professor at the Faculty of Mathematics and Informatics at Sofia University "St. Kliment Ohridski", Sofia, Bulgaria. Email: yavor.dankov@fmi.uni-sofia.bg / ORCID: https://orcid.org/0000-0003-3670-8599 / Scopus Author ID: 57202891597. ResearchGate Profile: https://www.researchgate.net/profile/Yavor-Dankov

CSSE 2021, October 22–24, 2021, Singapore, Singapore

ACM ISBN 978-1-4503-9067-5/21/10…$15.00
https://doi.org/10.1145/3494885.3494952

into two main categories, which represent the whole set of tools that would be needed to facilitate the processes of the design and creation of educational video games in the APOGEE platform [11]. The taxonomy represents the hierarchical structure of the software tools and in each category, there is a certain set of tools. Assistive tools are divided into two subcategories, and Analytics instruments are divided into three main subcategories. In this paper, the focus will be on the category of Assistive instruments, or more specifically on Design Management instruments [11].

The software instruments in the APOGEE software platform must meet a certain set of requirements: functional, non-functional (or quality characteristics), domain requirements, and other specific requirements. These requirements are diverse and are defined, depending on the criteria and needs of stakeholders, such as defined goals, scale, and purpose of the educational games to be designed, a different set of resources, target educational groups of users, and so on.

Applying the TIMED-VGE taxonomy in the APOGEE platform and guided by the purpose of the platform for automated creation of educational games by various specialists (including non-IT people who have no experience in programming) the basic user functionalities of all instruments are designed and presented in detail in [12, 13]. These results provide the opportunity to enter the next stages of the design and development of the APOGEE software tools and their further development and improvement.

This paper is based on the application of the TIMED-VGE taxonomy in the APOGEE software platform. The focus of this paper is on the development of a functional model of the instrument *"Maze Game Designer"*, based on the application of the TIMED-VGE taxonomy. The paper presents an extended version of the user functionalities of the Maze Game Designer software instrument. Therefore, based on the extended, defined user functionalities of the instrument, a functional model of the Maze Game Designer tool is developed and presented in the paper. Following the proposed functional model, the paper presents the workflow of the Edit Maze Game functional model of the Maze Game Designer instrument. The presented functional models and the workflow will support future processes of the development of the tool and its improvements.

The paper continues with the following parts. The paper presents the development of an extended version of the Maze Game Designer user functionalities, and on this basis, a conceptual model of the tool is developed. Then the paper presents the workflow of the design and creation of educational video games using assistive software instruments. The section, also presents the workflow of the Maze Game Designer instrument, as an assistive instrument for managing the design and creation of educational video games. The paper ends with a conclusion.

2 CONCEPTUAL MODEL OF MAZE GAME DESIGNER

2.1 Developing an Extended Version of Maze Game Designer User Functionalities

According to the application of the TIMED-VGE taxonomy in the APOGEE software platform, the Maze Game Designer is purposefully placed under number 1 in the category "Design management" instruments. This is because the tool is key to starting the process of designing and creating educational games in the APOGEE platform. For this reason, all requirements of that instrument must be analysed. Therefore, it is necessary to design functionalities that provide a good basis for the further development of appropriate tools that will benefit the designer and facilitate him to create and design video games for education in the APOGEE software platform, according to certain subjective criteria and goals.

The paper presents further development of the Maze Game Designer instrument and its functionalities, based on the core user functionalities of the instrument, analyzed and developed in [12]. As a result, the paper presents the extended version of these user functionalities, defined as follows:

- Load an XML Schema
- Create a maze game
- Save a maze game
- Delete a maze game
- Load an existing maze game
- Configure global settings
- Export a maze game
- Edit a maze game
- Choice of number of maze halls
- Choice of maze connectivity (number of doors in each maze hall and their direction)
- Choice of the overall visual arrangement of each maze hall (images and maps for walls, floor, ceiling, and so on)
- Choice of the overall audio arrangement in each maze hall (playback and sound effects)
- Choice of the overall audio arrangement in each mini-game (playback and sound effects)
- Updating of existing learning boards
- Selection of existing types of mini-games (uses an existing XSD schema)
- Selection of properties of existing types of mini-games (uses an existing XSD schema)
- Selection of distribution of existing types of mini-games through the maze rooms
- Saving the new designed maze

The extended version of Maze Game Designer user functionalities comprises 18 defined user functionalities, required for inclusion into the software instrument. This contributes to the further development of the functional model of the Maze Game Designer tool.

2.2 The Proposed Functional Model

Thanks to the extended version of the user functionalities, the paper presents the developed functional model of the Maze Game Designer instrument, according to the application of the TIMED-VGE taxonomy. The diagram (Figure 1) presents the *Functional Model of the Maze Game Designer* tool as a UML Use Case diagram, describing the main users of the Maze Game Designer, the core functionalities of the instrument, and the main associations between the interconnected elements in the diagram. The Unified Modelling Language or UML, provides an understandable manner to visually describe a representation of the tool. This functional model will provide a basis for further discussion for the development and improvement of the Maze Game Designer tool.

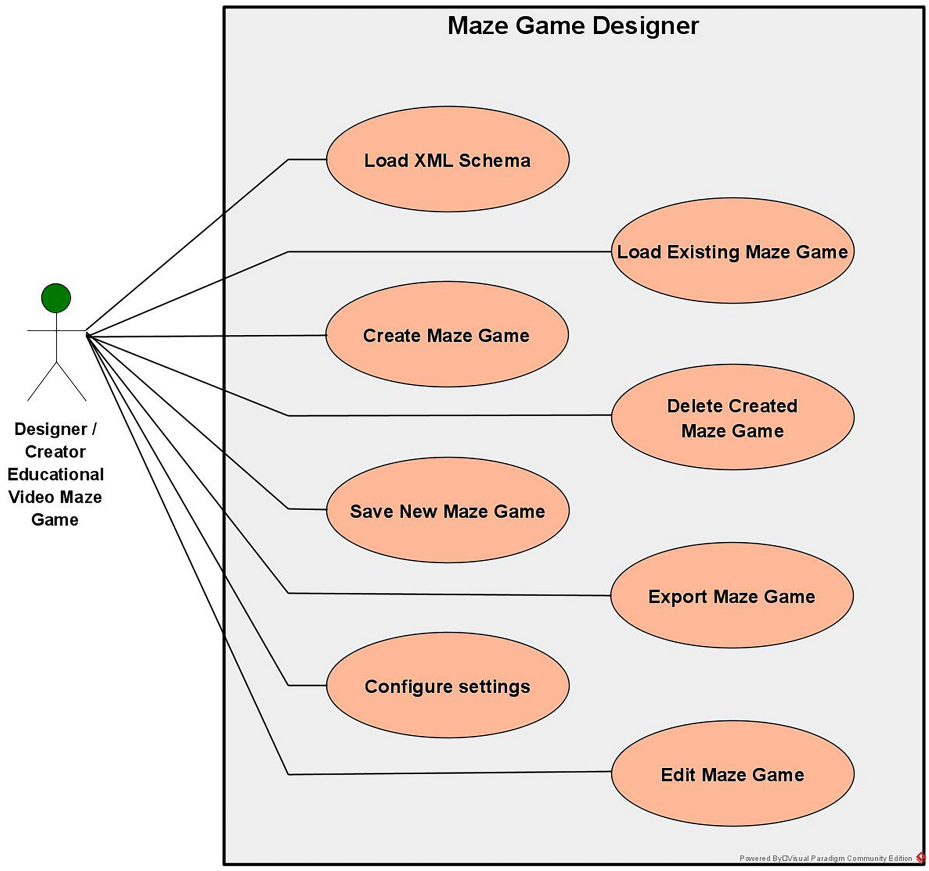

Figure 1: Functional Model of the Maze Game Designer

The UML Use Case diagram, illustrated in Figure 1, utilized for this purpose presents the extended version of the user functionalities of the Maze Game Designer instrument and the visual representations of these functional requirements are presented as oval shapes in the diagram. All diagrams in this paper are generated using the Visual Paradigm tool [14]. Every oval shape represents a defined function that has to be designed, developed, and integrated into the tool. All these oval shapes are positioned in the boundaries of the tool, meaning that all the functionalities belong to that instrument, and are developed internally. Everything outside the boundaries of the system is defined as different users of the tool. In this case, the Maze Game Designer Instrument has defined one main user as *Designer / Creator of Educational Video Maze Game*. If the tool in the future requires special software maintenance from a specialist such as a programmer or a developer, a second user can be defined as an Administrator. Every user of the platform has a certain set of defined user functionalities that they can use. Therefore, certain restrictions apply to different users of the instrument. If additional users of the tool are added in the future, then these restrictions will be applied to the respective defined users and the allowed functionalities will be formalized. The Designer / Creator of the educational video maze game can use all the functionalities, included in the extended version of the user functionalities of the Maze Game Designer software instrument.

In the process of developing the functional model of the tool, the need for the addition of more functionalities has occurred. This happened because, after the initial analysis and design of the tool and developing the basic functionalities of the Maze Game Designer, the author of this paper, has concluded, that the system must necessarily require a conception of validation and maintenance of the tool and such functionalities have to be developed to sustain such requirements. For this reason, two more user functionalities have been added to the tool. The first one is *Load XML Schema*, concerning the requirements for validations of the XML files against predefined XML Schemas. The second functionality that has to be developed and included in the functional model of the tool, is the *Configure Instrument* functionality. This will provide a basis for the maintenance of the tool and integration of future upgrades and developments. Both functionalities have been updated into the extended version of the user functionalities of the tool and can be utilized by the Designer / Creator of Educational Video Maze Game user.

After presenting the functional model of the Maze Game Designer tool, the paper presents the developed detailed *Functional Model of the Edit Maze Game Use Case* of the instrument, illustrated

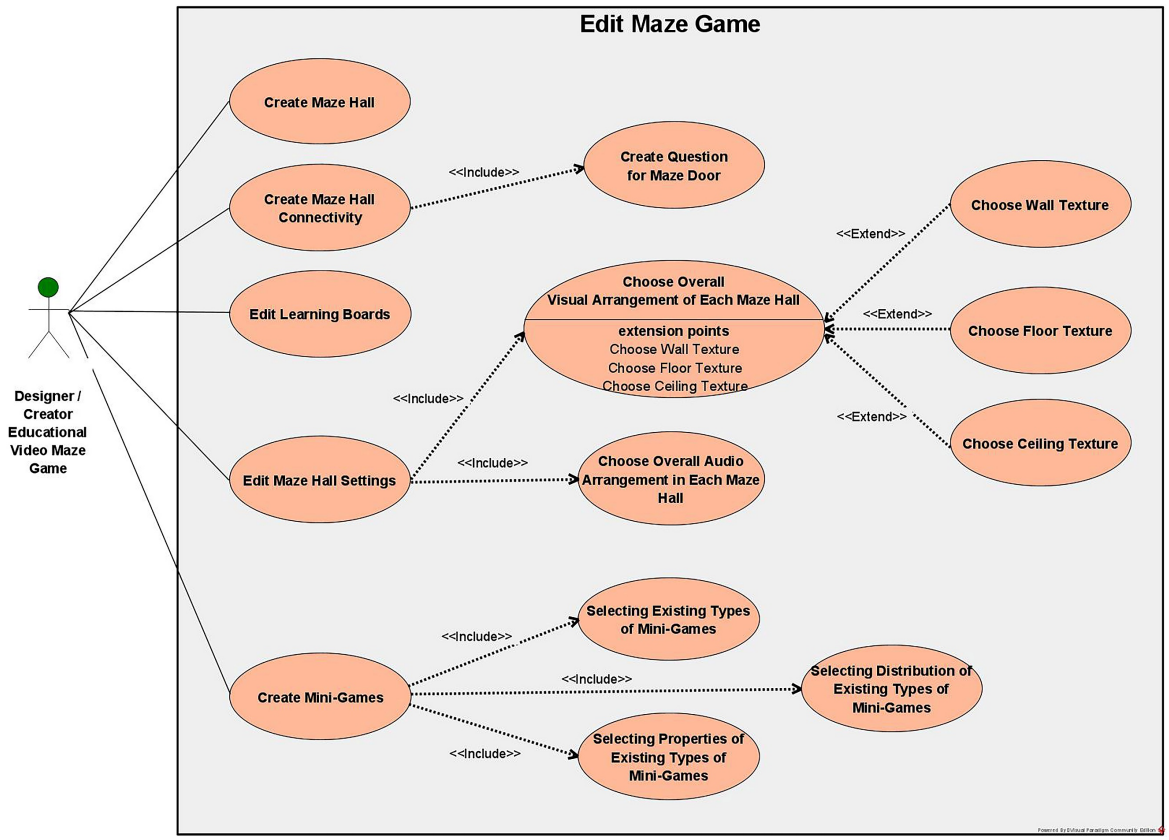

Figure 2: Detailed Functional Model of the Edit Maze Game Use Case of the Maze Game Designer instrument

Figure 2. The model describes in great detail the functionalities in the Edit Maze Game use case, the interaction of the user, and the main associations between the interconnected elements in the diagram.

The core functionalities in the Edit Maze Game use case, depicted in the detailed functional model are as follows: 1) Create maze hall; 2) Create maze hall connectivity; 3) Edit learning boards; 4) Edit maze hall settings, and 5) Create mini-games. The model presents different associations between the use cases, such as inclusion and extension associations. This comprises the associations between the Create maze hall connectivity use case and the Create Question for maze door use case, interconnected with the *include* association. This means that the process of creating a maze hall door always involves the definition of a special question to be answered to open the door to the next maze hall in the APOGEE software platform. In a similar way for the other user cases, which are presented in the model – the Edit maze hall settings use case includes the use case Choose overall visual arrangement of each maze hall and the use case Choose an overall audio arrangement in each maze hall.

The Designer / Creator of the Educational Video Maze Game user utilizes the functionality of the instrument for editing the maze hall settings. Therefore, the use case includes the choice of overall visual and audio arrangements of each maze hall of the video maze game. The uses cases Choose Wall Texture, Choose Floor Texture, and Choose Ceiling Texture to extend the base use case Choose an overall visual arrangement of each maze hall. The Designer / Creator of the Educational Video Maze Game user can also utilize the Create mini-games functionality, presented in the Create Mini-Games use case. The use case includes the use cases of Selecting Existing types of mini-games, Selecting Distribution of existing types of mini-games, and selecting Properties of existing types of mini-games. The detailed functional model can be further developed for every use case, but the focus of this paper is on illustrating one of the most important uses cases in the model

3 WORKFLOW OF THE MAZE GAME DESIGNER INSTRUMENT

The Maze Game Designer instrument represents one of the main assistive instruments, and the tool is key to starting the process of designing and creating educational games in the APOGEE platform. The workflow described in detail in [15], represents the overview of the main process in the platform, using the assistive instruments.

Based on the results in previous sections, regarding the extended version of the user functionalities of the Maze Game Designer tool, the proposed two functional models, and concerning the main process in the APOGEE software platform, the paper presents (Figure 3) the Workflow of the Maze Game Designer instrument.

This process describes the sequence of steps and opportunities that a designer can go through using one of the most important

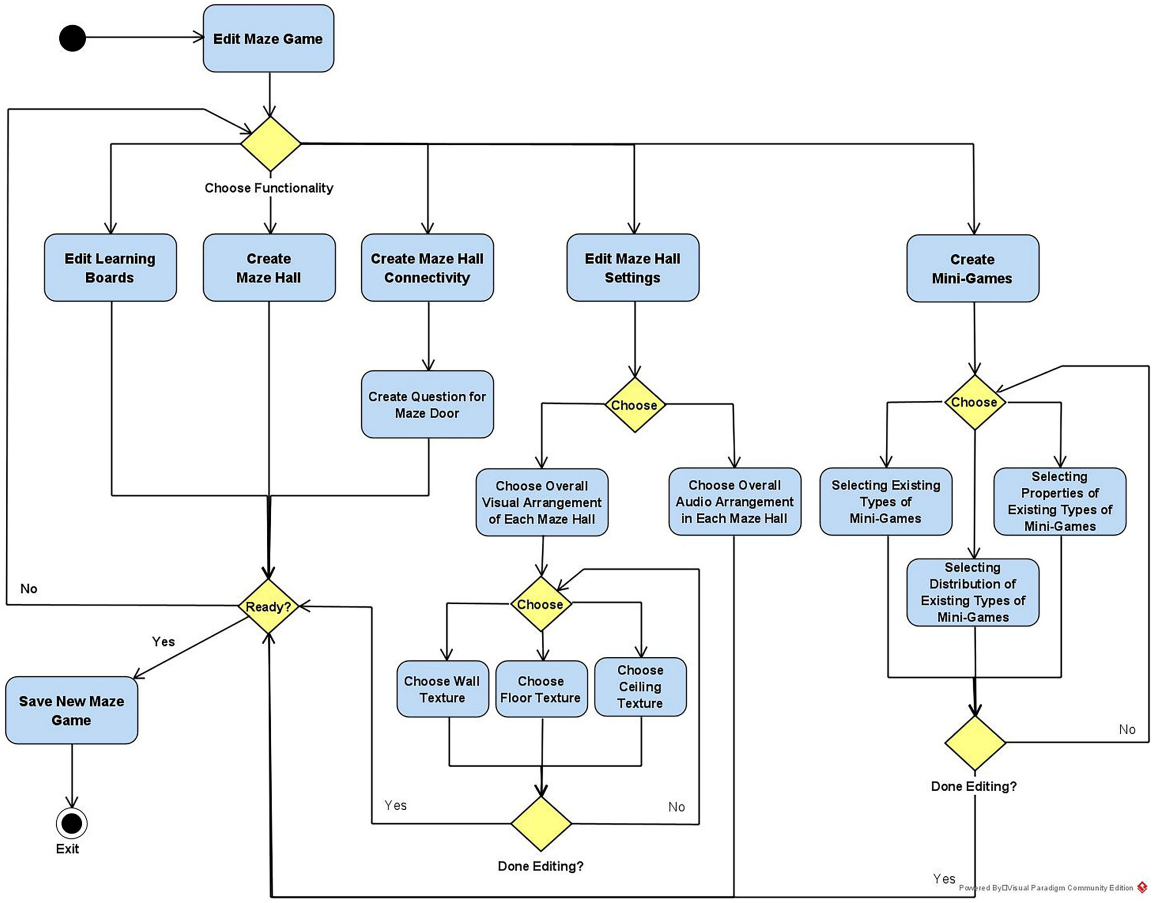

Figure 3: Workflow of the Maze Game Designer instrument

functionalities of the Maze Game Designer tool, part of the Assistive Tools in the APOGEE platform. The paper describes the workflow of this functionality of the tool because it shows the wide range of opportunities that will provide the integration of these functionalities in such a platform for designing and creating educational video games. The workflow is consistent and allows video game designers/creators to use the Maze Game Designer tool to start designing and editing maze game design. Therefore, the tool will provide opportunities not only for the design and creation of an educational video game, but also opportunities for its improvement based on the games played by the learners and the game results obtained. Based on these results and the data that will be processed, the designers will have the opportunity to improve the design of the educational video games already created in the APOGEE platform and to achieve even higher and satisfactory results. This in turn will improve the user experience in the platform, will contribute to the efficiency of achieving the desired educational and personalized goals and also will contribute to the overall improvement and refinement of the APOGEE platform and the software tools included in it.

The Designer / Creator of the Educational Video Maze Game user decides to edit a maze game. For this reason, the Edit Maze Game functionality is used. The process begins with choosing the available functionalities. The user has the ability to 1) Edit the learning boards; 2) Create maze hall; 3) Edit maze hall settings, and 4) Create mini-games. When choosing the Edit learning boards functionality the user edits the learning boards of the maze game and if the job is done, and the user decides that there's no more else to update into the learning boards, the user can decide to 1) Save the new maze game and exit the Edit Maze Game functionality, or to 2) return to the starting position of choosing the available functionalities. The Create Maze Hall function has a similar process logic. When using the Edit maze hall settings, the user can choose from the two available functionalities – choosing between the audio and visual arrangements of each maze hall. Editing the visual arrangement of each hall requires the choice of three available editing options, including the choice of wall texture, floor texture, and ceiling texture. After editing the visual arrangements of each maze hall, the user can choose between returning to edit the audio arrangements of each hall, to return to the choice of all the available functionalities, or to save the new maze game, and exit the Edit Maze Game functionality.

4 CONCLUSION

The paper presented an extended version of the user functionalities of the Maze Game Designer instrument, based on the application of the TIMED-VGE taxonomy in the APOGEE software platform. Based on these functionalities, two functional models of the instrument were developed and presented. This includes a Functional model of the tool and a detailed functional model of the Edit Maze Game use case of the instrument. The paper also presented the main workflow of the Maze Game Designer instrument. All these results will serve for the future development of the tool. The development of a key instrument of the APOGEE software platform, according to the application of the TIMED-VGE taxonomy, will contribute to increasing the designer's capabilities to use the platform's instruments for the design and creation of improved video games for education.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the APOGEE project, funded by the Bulgarian National Science Fund, Grant Agreement No. DN12/7/2017, and from the National Scientific Program "Infor-mation and Communication Technologies in Science, Education and Security" (ICT in SES) financed by the Ministry of Education and Science of Bulgaria.

REFERENCES

[1] Maxwell Hartt, Hadi Hosseini and Mehrnaz Mostafapour. 2020. Game On: Exploring the Effectiveness of Game-based Learning, Planning Practice & Research, 35:5, 589-604, DOI: 10.1080/02697459.2020.1778859

[2] Azam Abdelhakeem Khalid, Adel M. Sarea, Azzam Hannoon, Abdalmuttaleb M. A. Musleh Al-Sartawi. 2021. Game-Based Learning: Recommendations Driven from Literature. In: Musleh Al-Sartawi A.M.A. (eds) The Big Data-Driven Digital Economy: Artificial and Computational Intelligence. Studies in Computational Intelligence, vol 974. Springer, Cham. https://doi.org/10.1007/978-3-030-73057-4_10

[3] Jacqueline Schuldt and Helmut Niegemann. 2021. Instructional Design for Digital Game-Based Learning. In: Aprea C., Ifenthaler D. (eds) Game-based Learning Across the Disciplines. Advances in Game-Based Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-75142-5_13

[4] Boyan Bontchev, Valentina Terzieva and Elena Paunova-Hubenova, 2020 Personalization of Serious Games for Learning, Interactive Technology and Smart Education, Emerald, ISSN: 1741-5659, 18 (1), DOI: https://doi.org/10.1108/ITSE-05-2020-0069

[5] Christian Loh, Yanyan Sheng, and Dirk Ifenthaler, 2015. Book of Advances in Game-Based Learning, Serious Games Analytics: Methodologies for Performance Measurement, Assessment, and Improvement, Springer, https://doi.org/10.1007/978-3-319-05834-4

[6] Albena Antonova, Boyan Bontchev. 2019. Exploring puzzle-based learning for building effective and motivational maze video games for education, Proc. of 11th annual Int. Conf. on Education and New Learning Techn. (EDULEARN19), Palma de Mallorca, Spain, 1-3 July, 2019, ISBN: 978-84-09-12031-4, pp.2425-2434

[7] Plass, J., Mayer, R., Homer, B.: Handbook of game-based learning. In: Plass, J., Mayer, R., Homer, B., Cambridge, M.A. (eds.) MIT Press, Cambridge (2020). ISBN 9780262043380

[8] APOGEE Project Homepage, http://apogee.online/index-en.html , last accessed 2021/10/17.

[9] Boyan Bontchev. 2019. Rich Educational Video Mazes as a Visual Environment for Game-Based Learning. In CBU International Conference Proceedings, 7, https://doi.org/10.12955/cbup.v7.1388

[10] Valentina Terzieva,, Elena Paunova-Hubenova and Boyan Bontchev. 2019. Personalization of Educational Video Games in APOGEE, Chapter in: Brooks A., Brooks E. (eds) Interactivity, Game Creation, Design, Learning, and Innovation. ArtsIT 2019, DLI 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 328, Springer, Cham., pp 477-487. https://doi.org/10.1007/978-3-030-53294-9_34

[11] Yavor Dankov and Boyan Bontchev. 2020. Towards a taxonomy of instruments for facilitated design and evaluation of video games for education. In: Proceedings of the 21st International Conference on Computer Systems and Technologies 2020 (CompSysTech 2020), ACM, pp. 285–292 (2020). https://doi.org/10.1145/3407982.3408010

[12] Yavor Dankov and Boyan Bontchev. 2021. Software Instruments for Management of the Design of Educational Video Games. In: Ahram T., Taiar R., Groff F. (eds) Human Interaction, Emerging Technologies and Future Applications IV. IHIET-AI 2021. Advances in Intelligent Systems and Computing, vol 1378. Springer, Cham. https://doi.org/10.1007/978-3-030-74009-2_53

[13] Yavor Dankov and Boyan Bontchev.2021. Designing Software Instruments for Analysis and Visualization of Data Relevant to Playing Educational Video Games. In: Ahram T., Taiar R., Groff F. (eds) Human Interaction, Emerging Technologies and Future Applications IV. IHIET-AI 2021. Ad-vances in Intelligent Systems and Computing, vol 1378. Springer, Cham. https://doi.org/10.1007/978-3-030-74009-2_54

[14] Visual Paradigm Tool Official Website, https://www.visual-paradigm.com/, Last Accessed: 15 October 2021

[15] Yavor Dankov, Boyan Bontchev and Valentina Terzieva. 2021. Design and Creation of Educational Video Games Using Assistive Software Instruments. In: Ahram T.Z., Karwowski W., Kalra J. (eds) Advances in Artificial Intelligence, Software and Systems Engineering. AHFE 2021. Lecture Notes in Networks and Systems, vol 271. Springer, Cham. https://doi.org/10.1007/978-3-030-80624-8_42