Towards a taxonomy of support systems in serious games about programming

Pavlos Toukiloglou[1], Stelios Xinogalos[2]
toukiloglou@uom.edu.gr, stelios@uom.edu.gr
[1] PhD candidate,
[2] Professor, Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece

Abstract

Serious games (SG), designed for purposes beyond entertainment, have gained popularity in engaging students in programming education. One crucial aspect of SG is a support system, which plays a pivotal role in achieving their educational objectives. However, there is a lack of consensus in the literature regarding the terminology and classification of support systems, leading to inconsistencies and varying interpretations among researchers. To address this gap, this ongoing research presents the Taxonomy of Support Methods for Serious Games for Programming (TSM-SGP). This taxonomy provides a systematic approach to identifying and categorizing support systems, specifically in the domain of programming. It offers clarity in categorizing support systems, enabling researchers, educators, and game designers to gain a deeper understanding of their roles and effectiveness. Additionally, the taxonomy reveals insightful relationships between support methods and their properties, contributing to the development of effective SG.

Keywords: Taxonomy, Serious games, support systems, programming

Introduction

Serious games (SG), also known as educational games, serve a primary purpose beyond mere entertainment. The concept of SG was first mentioned by Clark C Abt in the 1970s (Elder, 1971) and defined as games that have an "explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement". The term "serious" encompasses video games utilized across various sectors, including defense, education, scientific exploration, healthcare, and more (Susi et al., 2007). These games have the potential to facilitate a deep understanding of specific topics and be an effective learning tool (De Freitas, 2018). SG are becoming increasingly popular as a method of delivering educational content to students (Mordor Intelligence, n.d.). Their ability to enhance engagement and provide immersive learning experiences makes them a compelling choice in educational settings. Computational thinking (CT), on the other hand, has become a fundamental skill of analytical thinking for everyone in the 21st century (Wing, 2006). It encompasses the utilization of fundamental computer science principles, including abstraction, debugging, and problem-solving. Programming fosters CT since it serves as a scaffold for its development (Belmar, 2022). Regardless, programming is a complex task, posing significant challenges in the teaching and learning process. To address this problem and enhance students' chances of success, SG emerged as a promising option. By leveraging the engaging nature of video games, SG can serve as interactive platforms that facilitate active learning and skill improvement for aspiring programmers.

Nonetheless, the effectiveness of SG as a learning tool relies heavily on the presence of robust support systems and methods that aid learners in achieving their educational objectives. These systems play a crucial role in facilitating the acquisition of knowledge and skills (Toukiloglou & Xinogalos, 2023). They encompass a range of features and mechanisms to ensure that learners receive the appropriate level of guidance and feedback throughout the gameplay experience. Without well-designed support, SG may fail to deliver educational goals, hindering progress and skill acquisition. Therefore, a comprehensive understanding of the diverse types of support systems and methods available in SG is critical for maximizing their educational potential. This paper proposes a new taxonomical scheme called Taxonomy of Support Methods for Serious Games for Programming (TSM-SGP) that aims to establish a terminology for describing support systems in programming SG. Our goal is to develop a framework that fills in the gap that currently exists in the literature and contributes to the field's advancement by: identifying the various types of support systems within the context of SG for programming; facilitating the clear and precise expression of support systems within scientific papers, enabling readers to readily discern their key elements and distinctive characteristics; providing clarity and a common understanding among SG designers, developers, academics, and students regarding the available support options within these games.

The paper is structured as follows: Section 2 provides an overview of the existing literature in the o support systems in SG. Section 3 presents an in-depth classification of the diverse support methods, highlighting their distinct characteristics and applications. Lastly, in Section 4, the results are discussed, and use cases of the taxonomy are illustrated, offering insights into potential future research and development in the field of SG support systems.

Related work

The existing literature on taxonomies of support systems in SG is notably limited in scope. Despite scholarly attention, consensus on characterizing support systems and relevant categories in SG is lacking. This paper addresses this gap by presenting an ongoing effort to develop a comprehensive and practical categorization of support systems. To the best of our knowledge, this is the only dedicated taxonomy specifically focused on the subject. However, it is worth noting that support systems have been acknowledged as important factors within other existing taxonomies. This section explores these taxonomies and their implications within the broader context of understanding support systems in SGs.

In a previous taxonomy study conducted by Laine and Lindberg (2020), feedback was identified as a significant factor influencing motivated engagement in SGs. According to the study, instructions, tutorials, prompts, and relevant feedback through different channels enhanced player motivation. It concluded that support is a key component of achieving a state of flow in players and a critical mechanism that facilitates optimal engagement within their taxonomy model. Rato and Prada (2021) explored the concept of support systems within a taxonomy of social roles for agents in games. They proposed a wide range of social interactions that agents can engage in during gameplay. One role discussed in the taxonomy is the Tutor/Learner NPC (Non-Player Character). This agent is responsible for providing game state information to players and offering guidance on gameplay actions and strategies. The provided information can be delivered proactively or explicitly upon player request. In an empirical study by Bedwell et al. (2012), a taxonomy was introduced that links game attributes to learning outcomes. The taxonomy emphasizes the importance of assessment as a key component in SG. Assessment in this context refers to the feedback provided to players throughout the game, guiding player actions and offering both subtle hints and instructions.

Classification criteria

This section provides a comprehensive description of the constituent elements of each category and their application within the taxonomy. To ensure that all key criteria for a formal and precise support mechanism description were included, the following methodology has been commenced. Initially, the objectives were defined to establish a clear scope and goal for the proposed taxonomy. Subsequently, a literature review was conducted on support systems for SG examining various aspects of their implementation. Building upon the findings, a draft of the taxonomy including potential categories and subcategories was identified. Then, based on the draft taxonomy, data collection of existing SG has been commenced to analyze how support systems are structured and utilized. Next, according to the results, the taxonomy was refined identifying gaps, redundancies, or overlaps in the categories. Future research will validate, adjust, and finalize the taxonomy through peer review, feedback from experts, and insights gained from empirical analysis.

Integration

The integration of support content in SG can be categorized as internal or external. Internal support refers to assistance provided within the game environment, seamlessly integrated into the gameplay experience. It includes features such as in-game tutorials, pop-up tips, and contextual help systems. In contrast, external support systems are accessed outside the game environment, through websites, applications, or physical materials. They can be initiated by the SG, instructor, or student. OGITS (Hooshyar, 2018) is an example that combines both types of integration as it provides internal support with learning material accessible within the game and external resources from webpages through a separate window. The floating window is a special type of internal integration that is a part of the game application but does not interact with the game world. Both internal and external support systems have their advantages and limitations. Internal support systems are frequently characterized by greater immersion and seamlessness as they are more compatible with the overall style and design of the SG. Contrarily, external support systems can be maintained and updated independently, allowing for greater flexibility and customization.

Media

The media category is comprised of five main formats, each offering a unique means of delivering information and engaging learners. *Text* support is a common and versatile form found in most SGs. It utilizes written language to effectively communicate information to players with precision and clarity. The text offers flexibility in formatting and style, allowing game designers to customize it according to their specific requirements. Furthermore, text is efficient in terms of memory usage and speed compared to other media such as video or audio. It is also easily searchable, editable, and updatable as needed. Nonetheless, there are drawbacks associated with text-based support. It may not be as engaging as other media, as it lacks visual and auditory appeal. Also, it can pose challenges for learners with weak reading skills or limited language proficiency. Finally, text may struggle to convey complex emotions, tones, or subtle meanings that can be expressed through other media forms. *Images* are also commonly used in SG to convey visual cues and information to players. They are effective in presenting complex concepts and ideas in a compact and clear manner. In addition to enhancing aesthetics, images provide immersion and serve as visual relief in text-heavy environments, making the learning experience engaging and less overwhelming. *Video* offers a multisensory experience by combining visuals and audio to express complex ideas. Its use in SG enhances immersion and engagement, enabling learners to interact meaningfully with the content. However, producing video content can be time-consuming and costly. Examples of SGs utilizing video support to demonstrate programming commands include Minerva (Lindberg & Laine, 2018), Hour of Code (Hour of Code, n.d.), and Rodocodo (Rodocodo, n.d.).

Audio support in SG can be delivered through various means such as background music, voice-overs, sound effects, and narration. It enhances immersion and emotional connection, particularly when visual cues are insufficient. Nonetheless, audio support may pose challenges for learners with hearing impairments or limited language proficiency. Furthermore, the use of audio support can be costly and time-consuming, as high-quality recordings may require professional equipment, musicians, and actors. Finally, in a shared environment such as a computer lab without individual headphones, audio can cause disturbances among players. *Animation* encompasses a dynamic form of visual content that shares similarities with videos. However, what sets animation apart is its utilization of the game's inherent elements, such as sprites, 3D models, and other graphical assets, seamlessly integrated within the game environment. Unlike pre-recorded videos, SG animations are rendered in real-time, allowing for interactive and responsive experiences. This facilitates engaging and captivating learning experiences for players making animation a highly effective medium of support. Nevertheless, incorporating game elements into the support design introduces an additional layer of complexity. This fact can impact the cost and completion time of the SG, as it requires careful coordination and integration of the animation with the existing game mechanics and assets. An SG example can be found in Kodable (Kodable, n.d.) where animations utilizing game graphics are shown before each puzzle to explain game mechanics and new learning concepts.

Technique

The technique element plays a crucial role in identifying the specific approach employed by the SG to deliver its learning content. While it is possible for a SG to utilize multiple ones, there is typically one dominant technique that characterizes the core concept of the support system. *Tips* offer learners helpful suggestions or recommendations related to gameplay or learning content. They appear as short text messages or pop-up windows during the game, providing targeted support and feedback on navigation, task completion, or problem-solving. However, it is important to balance the use of tips as excessive use can disrupt gameplay, reducing challenge and engagement. Tips are common techniques utilized in articles about SG such as BOTS (Hicks et al., 2014), and employed in numerous SG like most of the Hour of Code games (Code.org, n.d.). *Manual/Instructions* can be used as a support method to provide learners with in-depth information about a specific topic. They are usually presented in a digital format providing a comprehensive and structured approach to complex topics and concepts. Manuals are also often written by subject matter experts, ensuring that the content is accurate and up-to-date. Though they can be lengthy and dense, potentially overwhelming learners with new information, they may not be suitable for learners who prefer more interactive and visual forms of support.

Working examples provide demonstrations of how a specific task or concept can be performed or applied. They allow learners to see the learning concepts in action and observe how different strategies and actions can be employed to tackle challenges within the game environment. Players analyze and reflect upon the demonstrated tasks by applying the acquired knowledge in similar situations. Working examples are relatively new in the field of SG for programming. However, implementations like DungeonClass (Toukiloglou & Xinogalos, 2022) and NanoDoc (Toukiloglou & Xinogalos, 2023) show promising results in learning efficiency. *Chat / NPC Dialog* offers players the opportunity to engage in communication with NPCs or other human players to seek assistance and guidance. In the case of NPCs, the requests and responses can be scripted or dynamic. Dialog trees provide players with a structured framework to navigate and explore the available support options. Chat usually consists of focused short messages that address specific subjects or topics of interest about the game. Nonetheless, it is important to note that when engaging in chat with another human player, there is a potential challenge of introducing educational noise or deviating from the intended learning path. While human interactions can offer valuable insights and foster collaboration, there is a need for moderation and guidance. This will ensure that the conversation remains focused on educational objectives and does not steer players off the course. Human chat communication is exemplified in SG such as CMX (Malliarakis et al., 2016) and Quiz Time! (Troussas, 2020). An example of NPC dialog support can be found in a Neverwinter Nights based game (Soflano et al., 2015). *Narration* serves as a form of support that leverages explicit audio to deliver the learning content. It is employed through an NPC engaging in dialogue with the player or a dedicated narrator providing guidance. Narration support can assume different roles based on the game's design and learning objectives such as reinforcing specific learning points or providing direction and contextual information to players.

Source

Despite the current practice of *human-authored* learning content in support systems for SG, recent developments in computer science have the potential to profoundly transform its creation process. With the emergence of sophisticated technologies and techniques, such as natural language processing and machine learning algorithms, there is a growing possibility of automating or augmenting the creation of support content. The availability of *Artificial Intelligence* (AI) Application Programming Interfaces (APIs) made it possible to create support content in SG without the need for human experts. Language models such as GPT-3.5 (OpenAI, n.d.) and LaMDA (Cheng & Thoppilan, 2022) have the capability to generate responses that closely resemble human-like interactions. These models can understand player questions and generate coherent and contextually relevant responses. However, their functionality primarily revolves around text-based interactions, limiting the inclusion of other forms of support. Furthermore, it is important to acknowledge that content generated solely by AI may be susceptible to errors since human fact-checking is absent from the process. Although the transition to computer-generated support content is still an evolving area, as technology progresses, such methods will likely be integrated in the near future.

Individualization

Individualization categorizes the SG support as adaptive or non-adaptive. *Adaptive support* refers to personalized and customized support that is tailored to the individual needs of the learner. This type is designed to adapt to the player's progress and performance, adjusting the level and learning content accordingly. SG use data-driven approaches to create player knowledge models or artificial neural networks to identify the learning gaps and deliver focused support. On the other hand, *non-adaptive support* refers to support that is provided in a more generic or standardized way, without taking into account the individual knowledge level. This type of support is often designed to be more broadly applicable regardless of individual progress or performance. Although it is considered less effective it is faster and easier to implement. The number of adaptive programming SG has increased in recent years as they show positive learning results (Toukiloglou & Xinogalos, 2023).

Engagement

Engagement classifies player involvement and interaction with the support method as interactive or non-interactive. *Interactive* support systems allow players to actively engage with the content by taking action or making decisions as it unfolds. These systems are more difficult to implement since they need to track and respond to user actions effectively. However, they offer a higher level of engagement. An example of interactive support is shown in Nanodoc (Toukiloglou & Xinogalos, 2023) where players can interact with the presented working examples. On the other hand, *non-interactive* support systems first present the relevant content to the player and then allow them to make choices or take actions within the game. *Non-interactive* support systems are more commonly employed as they are easier to design and relatively safer in terms of unpredicted player actions.

Image 1 summarizes the proposed taxonomy factors. By following a typical syntax outlined by the taxonomy, a support system can be precisely identified by incorporating the technique and media terms and subsequently appending any relevant prefixes. Examples of support systems based on this syntax include "external manual", "adaptive text tips", "AI audio narration", "interactive animated working example", and "in-game video chat".

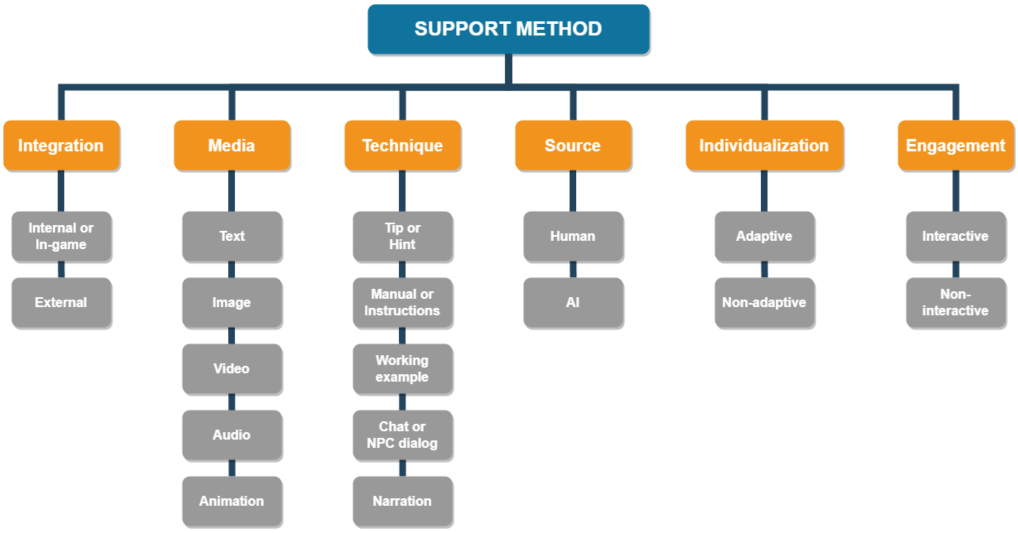

Image 1: The Taxonomy factors

Discussion

The TSM-SGP framework allows the identification of support systems in SG. By classifying and organizing the different types of support, TSM-SGP provides a systematic approach to examine and categorize the support systems. The taxonomy serves as a valuable tool for researchers, educators, and game designers providing a deeper understanding of support methods, their roles, and effectiveness. The TSM-SGP taxonomy is designed to be expandable, allowing for the inclusion of new support methods as they emerge. For instance, if new media forms such as augmented reality are introduced as support methods, they can be easily incorporated into the taxonomy. Similarly, if semi-AI systems become available as sources of support, they can be included in the source category. Future research in this field may also introduce a more robust system using acronyms to identify support methods in scientific papers. Additionally, new categories can be introduced to provide information on aspects such as collaborative support from other players.

The following examples of SG about programming, highlight the advantages of employing the TSM-SGP taxonomy. In the case of DungeonClass (Toukiloglou and Xinogalos, 2022), a study investigated the impact of working examples compared to textual support. According to the taxonomy syntax, the term "text tips" could be used instead, allowing a more precise characterization. Moreover, since according to the article the working examples are not interactive and displayed with animated NPCs, the prefixes non-interactive and/or animated would make the term self-explanatory. Similarly, in a study on adaptive support using an SG based on Neverwinter Nights (Soflano et al., 2015), the support types were labeled as non-adaptive text and adaptive text. However, considering that the support involved communication through NPCs, a more precise naming convention would be NPC text dialog and adaptive NPC text dialog. This terminology aligns with the nature of the support and facilitates better comprehension. In the case of Minerva (Lindberg & Laine, 2018), the support is initially identified as text, images, and video. However, upon further examination of the support mechanisms described in the article, a proposed identification based on the taxonomy would be instructional text, image hints, and working examples video. This refined categorization provides a more explicit understanding of the intended purpose of each media type mentioned, enhancing the clarity of the support system.

Limitations

Although the taxonomy is developed based on existing literature and expert insights, it lacks empirical validation. Variations in researchers' perspectives and interpretations may arise when categorizing certain support methods, leading to inconsistencies or disagreements in their implementation. Furthermore, the taxonomy's generalizability across different domains and specific contexts may be limited. The efficacy of specific support methods can vary depending on factors such as subject matter, target audience, and learning objectives. Further research is necessary to validate its effectiveness in practical applications and evaluate its usefulness in diverse SG settings and educational contexts.

Conclusion

The support method categories discussed in this article have significant implications that extend beyond the study of support systems as they contribute to a deeper understanding of effective SG design. The taxonomy derived from this study serves as a valuable tool for organizing support methods, while the identified relationships between support and properties offer assistance to the scientific community in developing an integrated model of learning within SG. This, in turn, advances the field of game-based learning and promotes the adoption of more sophisticated approaches to improve the educational potential of games. Although this work is focused on the context of programming in SGs, the taxonomy can serve as a foundation for exploring support systems in other domains. Future research will involve empirical validation and refinement of the taxonomy to ensure its effectiveness in practical applications and diverse educational contexts.

References

Bedwell, W. L., Pavlas, D., Heyne, K., Lazzara, E. H., & Salas, E. (2012). Toward a taxonomy linking game attributes to learning: An empirical study. *Simulation & Gaming*, *43*(6), 729-760.

Belmar, H. (2022). Review on the Teaching of Programming and Computational Thinking in the World. Frontiers in Computer Science, 4, 997222.

Cheng, H.-T., & Thoppilan, R. (2022, January 21). LaMDA: Towards Safe, Grounded, and High-Quality Dialog Models for Everything. Google AI Blog. https://ai.googleblog.com/2022/01/lamda-towards-safe-grounded-and-high.html

Code.org. (n.d.). Hour of Code. Code.org. Retrieved May 19, 2023, from https://hourofcode.com/us

De Freitas, S. (2018). Are games effective learning tools? A review of educational games. *Journal of Educational Technology & Society*, *21*(2), 74-84.

Elder, C. D. (1971). Serious Games. By Clark C. Abt.(New York: The Viking Press, Inc., 1970. Pp. 176. 4.95 paper.). *American Political Science Review*, *65*(4), 1158-1159.

Hicks, A., Peddycord, B., & Barnes, T. (2014). Building games to learn from their players: Generating hints in a serious game. *In Intelligent Tutoring Systems: 12th International Conference*, ITS 2014, Honolulu, HI, USA, June 5-9, 2014. Proceedings 12 (pp. 312-317). Springer International Publishing.

Hooshyar, D., Binti Ahmad, R., Wang, M., Yousefi, M., Fathi, M., & Lim, H. (2018). Development and evaluation of a game-based bayesian intelligent tutoring system for teaching programming. *Journal of Educational Computing Research*, *56*(6), 775-801.

Kodable. (n.d.). Programming for Kids. Kodable. Retrieved May 19, 2023, from https://www.kodable.com/

Laine, T. H., & Lindberg, R. S. (2020). Designing engaging games for education: A systematic literature review on game motivators and design principles. *IEEE Transactions on Learning Technologies*, *13*(4), 804-821.

Lindberg, R. S., & Laine, T. H. (2018). Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12. *International Journal of Serious Games*, *5*(2), 3-24.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2016). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, *10*(2), 219-235.

Mordor Intelligence. (n.d.). Serious Games Market Analysis - Industry Report - Trends, Size & Share. https://www.mordorintelligence.com/industry-reports/serious-games-market

OpenAI. (n.d.). Models - OpenAI API. Retrieved May 19, 2023, from https://platform.openai.com/docs/models/gpt-3-5

Rato, D., & Prada, R. (2021). A taxonomy of social roles for agents in games. In *Entertainment Computing–ICEC 2021: 20th IFIP TC 14 International Conference, ICEC 2021*, Coimbra, Portugal, November 2–5, 2021, Proceedings 20 (pp. 75-87). Springer International Publishing.

Rodocodo. (n.d.). Coding game for primary school children ages 4-11. Rodocodo. Retrieved May 23, 2023 from https://www.rodocodo.com/

Soflano, M., Connolly, T. M., & Hainey, T. (2015). An application of adaptive games-based learning based on learning style to teach SQL. *Computers & Education*, *86*, 192-211.

Susi, T., Johannesson, M., & Backlund, P. (2007). Serious Games—An Overview. 28. 2007. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2:2416 (accessed on 5 December 2022).

Toukiloglou, P., & Xinogalos, S. (2022). Ingame worked examples support as an alternative to textual instructions in serious games about programming. *Journal of Educational Computing Research*, *60*(7), 1615-1636.

Toukiloglou, P., & Xinogalos, S. (2023). Adaptive Support With Working Examples in Serious Games About Programming. *Journal of Educational Computing Research*, 07356331231151393.

Toukiloglou, P., & Xinogalos, S. (2023). A Systematic Literature Review on Adaptive Supports in Serious Games for Programming. *Information*, *14*(5), 277.

Troussas, C., Krouska, A., & Sgouropoulou, C. (2020). Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. *Computers & Education*, *144*, 103698.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.