A User Experience and Usability Test on Playing Games Specified as Graph Grammars in GrameStation

Júlia Veiga da Silva[1], **Braz Araujo da Silva Junior**[1], **Luciana Foss**[1], **Simone André da Costa Cavalheiro**[1]

[1]Fundamentals of Computing Laboratory – Federal University of Pelotas (UFPel)
CEP 96.010-610 – Pelotas – RS – Brazil

{jvsilva, badsjunior, lfoss, simone.costa}@inf.ufpel.edu.br

Abstract. *GrameStation is a game engine based on Graph Grammar, a formal language used to describe systems and verify properties. Despite being intuitive, understanding a Graph Grammar may not be trivial for those who do not have previous experiences with this formalism. Therefore, we propose an experiment to analyze the support given by GrameStation during the running of games modeled as Graph Grammars in order to facilitate this understanding. We analyzed three groups of people with different levels of knowledge about Graph Grammar. They played two games in the platform and answered a questionnaire. It was found that the greatest difficulty was in understanding how to make the mappings to progress in the game and, as a consequence, the addition of tutorials in the platform was mentioned by several participants.*

1. Introduction

Computational **T**hinking (**CT**) is a problem-solving process based on **C**omputer **S**cience (**CS**) [Wing 2006] that has been intensively explored in the scientific community. Wing (2006) advocates that CT's general-purpose skills should be learned by everyone, not just CS professionals. In recent works, it is observed the application of different practices allied to CT in Basic Education, such as unplugged CT [Souza and Nunes 2019] and block-based programming [Sousa et al. 2020]. In this context, having problem-solving based on CS as focus and proposing a new approach for CS education, a game engine that uses **G**raph **G**rammar (**GG**) to create games was proposed: the GrameStation [Silva Junior et al. 2021].

A GG is a visual and formal language used to describe systems and verify properties, while GramesStation is a tool for making and running games using GGs. GrameStation was developed on the Unity platform [Technologies 2022] using C# codes. It allows to make, edit, and run **Gr**aph Ga**mes** (**Grames**). That is, creating a Grame corresponds to specifying a GG in the tool. GrameStation is meant to allow all publics to design and/or run GGs in a ludic environment while fostering CT skills, such as pattern recognition, data representation, and abstraction. Thus, it is a tool aiming to support education in multiple ways: encouraging active methodologies and creative learning by turning students into "Grame makers"; enabling teachers to create educational games without coding; and developing CT in activities that are not restricted to CS subjects.

However, GrameStation can only successfully achieve these goals if it is able to provide the proper support for presenting GG (a formal language) to the general public in a friendly and engaging way – since although intuitive, specifying and even understanding a specification in GG may not be simple for people who do not have previous experience in this subject. In order to solve this issue, pedagogical agents (called "Gramers") were proposed to help and guide users in the platform [Silva et al. 2021]. The authors' proposal was guided by difficulties identified during creating games modeled as GGs in their first experiences with GrameStation. These difficulties were related to terms and actions exclusive to the platform, GG's terms and concepts, the definition of relations between elements, and restrictions imposed by the graphs.

In this context, an experiment was organized to analyze more deeply technical aspects relevant for the future implementation of those agents in GrameStation – such as the recurrent challenges the students face during the creation and execution of games – aiming to evaluate the suitability of the initial proposal. Therefore, this paper presents a user experience and usability experiment report of GrameStation, in order to analyze the support given by the tool during the running of games modeled as GG. We adopted the empirical strategy suggested by Nielsen (1994) as a way to inspect usability. Following this strategy, we develop and report an interface test with real users. A total of 17 people with CS background was selected. Each of them played two games and answered a questionnaire about their experience in the platform. The profile of participants was chosen, and the tasks performed by them were defined specifically for this experiment. The questionnaire was an adaptation of MEEGA+ proposed by Petri et al. (2016).

Several works use MEEGA+ to evaluate educational games in diverse areas, from computing to health [Karakasis and Xinogalos 2020, Tsopra et al. 2020, Venigalla and Chimalakonda 2020]. Moreover, it is common to notice adaptations of the questionnaire to fit it with the games to be evaluated. Karakasis and Xinogalos (2020) for example, deleted all MEEGA+ questions about social interaction with other players since this feature is not supported by BlockScript, an educational game that aims to help students develop their CT by learning basic programming concepts, designing algorithms, and correcting mistakes. Venigalla and Chimalakonda (2020) also omitted some dimensions of the questionnaire to evaluate their proposal: a game for novice programmers to learn debugging. However, despite being very popular to evaluate educational games, no studies were found about the application of this model to evaluate game engines.

The rest of this paper is organized as follows. Section 2 provides basic notions of GG. Section 3 presents the GG-based game engine GrameStation and how a user can play games using the tool. Section 4 details the activity applied in this experiment. Section 5 gives an overview of obtained results and highlights some of them. Section 6 discusses the main results of this experiment. Section 7 concludes the paper, discusses research directions, and indicates the future works.

2. Graph Grammar

Due to the increasing complexity of software systems over the years, it became necessary to elaborate more precise specifications to verify properties, correct errors, and describe the behavior of these systems [Ribeiro 2000]. In this sense, formal methods – based on mathematical methods – such as GG have been in the computing area to design and analyze systems. GG is a formal language [Ehrig et al. 1997] that can be seen as a generalization of the Chomsky grammars, replacing strings by graphs, or Petri Nets, with dynamic changes over the system topology and references between tokens [Ribeiro 2000].

A GG describes a system modeling its states as graphs (structures composed of vertices and edges usually represented by points and arrows, respectively) and simulating its events (transitions between states) with graph transformation rules. Formally, a GG is defined by a **type graph**, where the system elements (vertices and edges) are differentiated and restricted; a **start graph** (composed of the elements defined by the type graph) that defines in which state graph the system starts; and a set of **rules** that defines how the system progresses. A rule can be represented by two graphs: the **L**eft **H**and **S**ide (**LHS**) and the **R**ight **H**and **S**ide (**RHS**). The LHS expresses a condition for applying a rule and the RHS expresses a consequence of applying this rule. The condition of a rule is satisfied when it is possible to relate each element of the LHS graph to an element of the state graph – this mapping is called a **match**.

Figure 1 illustrates the specification of the Pac-Man game as a GG. The type graph (depicted at the top of the figure) declares the existence of Pac-Man, ghosts, fruits, places (grey dots), a counter (pink triangle), and the relations between these elements. The start graph (depicted at the bottom of the figure) shows one Pac-Man, one ghost, and three fruits in a 3x4 map of places, while the counter shows that no fruit has been eaten yet.

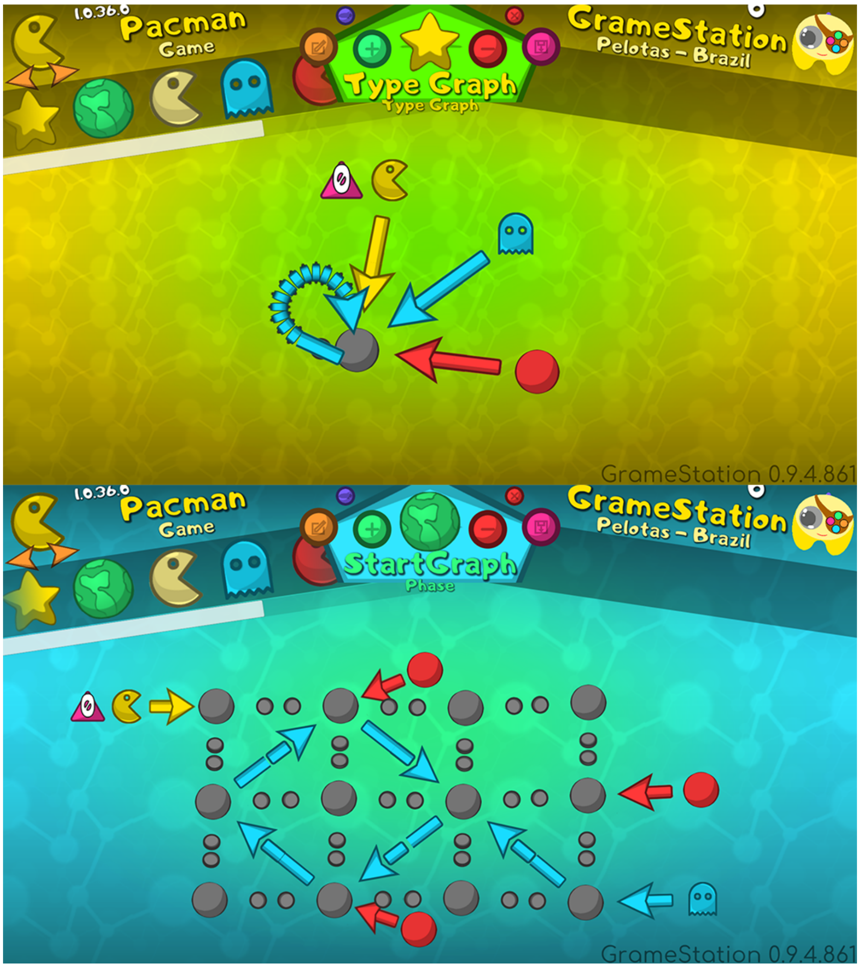

Figure 1. Type graph (above) and start graph (below) for the Pac-Man game

The set of rules (Figure 2) includes: *Pac Move*, *Ghost Move*, *Pac Eat*, and *Ghost*

Eat. The rules are represented by a pair of graphs linked by an arrow. In *Pac Move* rule, for instance, the LHS defines the condition to apply the rule: having a Pac-Man in a place that has a way to another. The RHS, on the other hand, defines the consequence of this rule: the Pac-Man is removed from its initial place and is moved to the next one. This mapping between graphs, respecting the original sources and targets of the edges, is called **morphism**. In this sense, the elements maintained during the application of the rule (i.e., related by the morphisms) are **preserved**, the elements lost (i.e, LHS' elements that are not related by morphisms) are **deleted**, and the new elements generated during the application (i.e., RHS' elements that are not related by the morphisms) are **created**.

Figure 2. Rules *Pac Move* (left, above), *Ghost Move* (right, above), *Pac Eat* (left, below), and *Ghost Eat* (right, below)

3. GrameStation

GrameStation is a GG-based tool used for creating and running games modeled according to this formal language. Due to the games' representation as GG, GrameStation also promotes the development of skills related to CT. These skills are developed both by the person who creates a game (specifies a GG) and by the person who runs a game (simulates a GG). Silva Junior (2019) detailed how CT concepts such as data representation, problem decomposition, abstraction, algorithms and processes, and parallelism are developed through the design and simulation of a game (GG). For instance, the whole gameplay in GrameStation revolves around **pattern matching** and **reevaluating** the current state of the game in a certain **data structure** (graph).

For playing a game in the tool the user can select, map, and apply the specified rules during the execution. The LHS and RHS graphs are shown when a rule is selected, then the user should find a match by clicking LHS elements and, in sequence, their corresponding elements in the state graph. Besides, GrameStation also signalizes when a match is correct or incorrect.

4. Methods

We invited 17 people (with different levels of knowledge about GG) to participate in an activity where they played two games in GrameStation and then completed a questionnaire about their experience. The questionnaire was an adaptation of the MEEGA+ model proposed by Petri et al. (2016). The activity was remote, asynchronous, and individually followed. All participants were people that have CS background: 1 doctoral student, 1 master, and 15 undergraduate students. The students were enrolled in three different courses (Systems Analysis and Development, Computer Science, and Computer Engineering) from four different higher education institutions of Rio Grande do Sul state, Brazil – Federal University of Pelotas (UFPel), Federal University of Rio Grande do Sul (UFRGS), Federal University of Pampa (Unipampa), and Federal Institute of Education, Science, and Technology from South of the Rio Grande do Sul (IFSul). GrameStation is not restricted to a specific target audience, however, for this first experiment, we selected this specific group because we assumed they could focus on the investigated issues at the current state of GrameStation – since people with computing backgrounds tend to be familiar with digital tools/platforms, graphs, formal languages, and understand actions through automated-feedback. Before the activity, all participants read and signed the Informed Consent Form (ICF), indicating they decided to take part in the research of their own free will and are aware of the usage of the data.

Among the 17 invited people, 7 had some previous experience with GG, so we divided the participants into three groups: one group composed of those who had previous experiences in GG (group 1) and two groups (each with 5 people) composed of those who did not have any knowledge in GG (groups 2 and 3). Group 2 was introduced to a video about GG's basic notions (not presented to group 3), so we could compare the impact of this little experience on using GrameStation. In order to define which groups the participants would fit into, an initial talk was carried out. In this meeting, an overview of the activity was given and asked if any of them had previous experiences with GG. All three groups did the same activity (they played the same games and answered the same questionnaire) and had 5 days to complete the activity remotely and asynchronously. The decision on the number of participants was based on Nielsen and Landauer (1993), who claim the best results come from testing no more than 5 users. According to the authors, as more and more users are added to a test, fewer and fewer new things are discovered, and after the fifth user, the same findings are observed repeatedly. Moreover, according to the authors, if the test is composed of three or more groups, at least 3 users are needed in each group – this way, it is possible to ensure at least 3 users cover the behavior diversity within the group [Nielsen and Landauer 1993].

The games played by the participants were Pac-Man and The Last Tree. The first is a simplified version of the classic 80's game, whose main goal is to eat all fruits in a map, by moving the main character through the paths while avoiding the ghost. The second is a puzzle game whose goal is to restore a destroyed forest from a single remaining tree. This game is a single-player version of the original one, which is a multiplayer turn-based strategy game [Silva Junior et al. 2017]. Pac-Man was chosen as one of the games due to its popularity. Therefore, most participants would already know the mechanics of the original game and could analyze and compare the way the game is shown in GrameStation. On the other hand, The Last Tree is far less known but is a game originally created as a GG, so the participants could play something that was properly designed to be a GG from the beginning.

Figure 3. Start graph and *Know Down* rule of The Last Tree game on GrameStation

A questionnaire was applied to evaluate the participants' experience with GrameStation and the games. We used an adapted version[1] of a questionnaire for evaluating educational games for computing: MEEGA+ (**M**odel **E**valuation of **E**ducational **Ga**mes) [Petri et al. 2016]. MEEGA+[2] is composed of three categories: Demographic Information, Usability, and Player Experience. The Usability and Player Experience categories are divided into areas. In the Usability category, the statements cover *aesthetics*, *learnability*, *operability*, and *accessibility* aspects, while in the Player Experience category, the statements cover *confidence*, *challenge*, *satisfaction*, *social interaction*, *fun*, *focused attention*, *relevance*, and *perceived learning* aspects. The response format for each item in the Usability and Player Experience categories is based on a 5-point Likert scale [DeVellis 2003] with alternative answers ranging from -2 ("strongly disagree") to 2 ("strongly agree").

We adapted MEEGA+ to meet our particularities, modifying, adding, and deleting some statements and questions. In the Demographic Information category, we only replaced the "Subject" field with a "Semester" field, which indicates the students' current semester. In the Usability category, we made minor changes in all statements, preserving their initial meaning while turning them explicitly related to specific GrameStation features (Q1-Q19). Some statements (Q1-Q3 and Q7-Q19) were replicated to be separately related to each resource: GrameStation, Pac-Man, and The Last Tree. Finally, in the User Experience category, we excluded some statements that did not fit our context (Q10, Q12, Q18-Q20, and Q24-Q31 of the original questionnaire), such as "I was able to interact with other people during the game", as GrameStation does not support users to play online yet. We also added specific statements (Q37-Q41) in the *perceived learning*

[1] Adapted version (in Portuguese) available on: https://bityli.com/CTVNan.
[2] Original version (in Portuguese) available on: https://bityli.com/ZSnsR.

area to analyze whether participants understood the GG concepts covered in GrameStation, as well as created a new categorization ("Graph Grammar") composed exclusively of statements related to GG (Q42 and Q43). We also added 10 questions (Q44-Q53) to collect specific data about participants' opinions on playing games modeled as GG and if they understood the role of each of the game rules.

5. Results

All 17 participants finished the activity, and we considered all the answers for the analysis. Figure 4 demonstrates the demographic information about participants: 41.2% (7) were female and 58.8% (10) were male. Regarding the age, 94.1% of people were between 18-28 years old, while 5.9% were between 29-39 years old. The Figure also illustrates the percentage of students in the different institutions, courses, and their current semesters. Additionally, are presented information about participants' familiarity with digital and non-digital games.

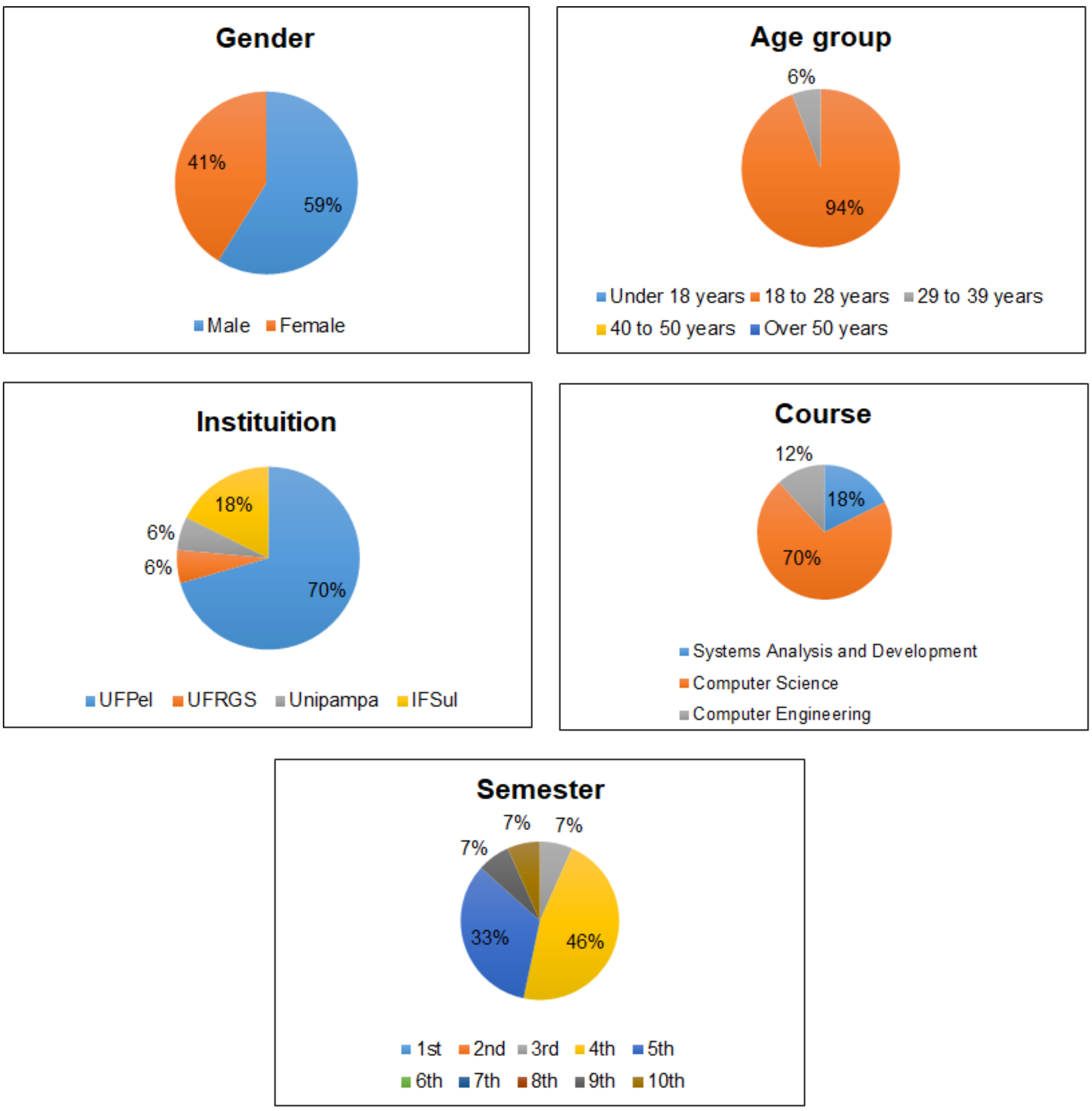

Figure 4. Demographic information charts

Figures 5 and 6 illustrate the results obtained in the Usability and Player Experience categories considering the three groups[3]. Column M in the figures represents the answers' median. The results can range from -2 ("strongly disagree") to 2 ("strongly agree"). Regarding the Usability category (Figure 5), the obtained results about GrameStation, Pac-Man, and The Last Tree were mostly positive ($M \geq 1$). We can highlight the statement about fonts (considering size and style) used in GrameStation (Q5), and the statement about the presentation of the Pac-Man game rules (Q16). Most participants strongly agreed with both statements. However, it is possible to notice that in *learnability*, Q8, Q10, Q11, Q13, and Q19 received less than 50% of positive responses. All these statements referred to The Last Tree game. Except for Q13, which did not receive negative responses, all others were answered with "strongly disagree" or "disagree" by participants.

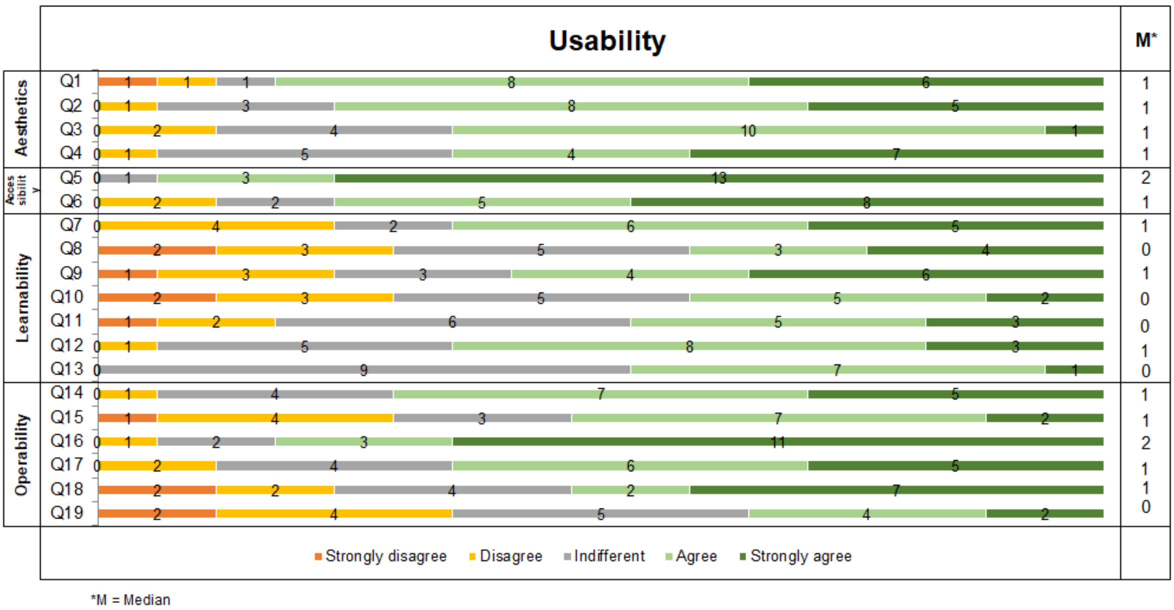

Figure 5. GrameStation usability chart

Overall, the result of the Player Experience category was also positive (Figure 6). We can highlight the Q25 statement, in which most participants demonstrate satisfaction

[3]Individual evaluations per group are available on: https://bityli.com/xuWfJ.

with the things learned in GrameStation; Q30 in which most participants claimed had fun using the platform; Q34 and Q35, in which participants agreed that The Last Tree game could be used for educational purposes; and Q36, in which participants demonstrate agree that GrameStation helps exercise pattern recognition. Additionally, the Q42 statement received only positive responses as it was restricted to people with previous GG experience. On the other hand, the Q43 statement received a lot of negative responses, since positive answers to "I have a good knowledge of Graph Grammar" were only expected from group 1. Finally, analyzing the groups separately, it is possible to observe that group 3 answered more statements with "totally disagree" or "indifferent", with emphasis to Q33: "I was able to understand my mistakes with the platform feedback".

Regarding discursive questions[4], most participants (13) left positive comments related to the design of GrameStation. They described the platform as ludic, intuitive, organized, and pleasant, and praised the colors and arts. Participants also left implementation suggestions to the platform, such as sound control and screen responsiveness. Considering the main difficulties in playing the games in GrameStation, 6 participants reported problems about do not understand how to execute game actions (group 1) and what exactly should be done to play and progress in the game (group 2 and 3), that is difficulties related to mappings. Especially about The Last Tree, 2 participants (groups 2 and 3) reported difficulty in understanding how to play and the lack of a clear objective. Three participants considered games with a lot of actions as a limitation of GrameStation. They considered the platform most suitable for strategic games and games that do not require the execution of actions quickly. Lastly, 8 participants (from the three groups) suggested creating tutorials for the GrameStation or features to turn actions more intuitive, such as informative texts and sound signals.

6. Discussion

At the process beginning, we obtained information about the previous experiences of the participants with GG – 7 claimed to have previous experience and 10 did not. However, we observed that some answers were given erroneously in the questionnaire, due to the participants' misinterpretation. For example, one of them reported having experience with GG in the questionnaire but later related that he confused GG with Graph Theory. Moreover, we divided the participants into three groups according to their knowledge about GG because we had the hypothesis that it would impact playing a Grame in GrameStation. So considering this separation, the introductory video about GG seemed to make difference to group 2, as the responses of group 3 were more negative in both categories (Usability and Player Experience).

Regarding the Usability category, 73% of the statements were answered positively by groups 1 and 2. However, statements Q8 and Q10 received only negative responses from group 2. Similar behavior could be seen in the Player Experience category – with emphasis on the *perceived learning* dimension (group 1) and *fun* and *focused attention* dimensions (group 2), which also presented a good result. Additionally, 50% of the statements in this category were answered only with "agree" and "strongly agree" by group 2. Finally, the charts of group 3 (participants without previous experience with GG that did not receive support material) were the most visually negative. In the Usability category, 5

[4] Available on: https://bityli.com/xuWfJ.

Figure 6. GrameStation player experience evaluation chart

statements (Q11, Q12, Q16, Q18, and Q19) received less than 50% approval and only 3 (Q2, Q3, and Q5) did not receive any negative response. Similar behavior was observed in the Player Experience category, in which only 3 of the 24 statements did not receive a negative answer. With these results, it is possible to assume that group 3 was the one who felt the most difficulty during the activity.

The answers to Q8, Q10, Q13, and Q19 revealed the difficulty to play The Last Tree. It contrasts with how easier Q7, Q9, Q12, and Q18 revealed that playing Pac-Man was. We assumed that people did not struggle to play Pac-Man because they already knew the core mechanics and goals of the game. Thus, they did not need much support to play. The Last Three otherwise, they did not know and the support GrameStation gave seemed insufficient. For instance, Q45 and Q49 revealed the participants felt the lack of a clear objective for this game. Regardless of the platform, each game has its own objectives, thus, explaining what the players are expected to do is always a task of the game designer. However, GrameStation does not support this communication between designer and player yet. Access to an "instructions panel", containing text written by the game designer to guide players, could solve this. Since a GG game can only finish through the application of a certain rule, an additional solution for this would be granting the player access to a "game goals panel", showing the LHS of rules that bring the game to an end. Additionally, possible limitations reported by participants about playing fast-paced games in GrameStation can also be circumvented. A possible adaptation is to create a kind of "pre-mapping" for the rules based on the players who apply them. Thus, the matching process can become a lot faster, following the game pace. GrameStation is still under development, with several improvements planned to be implemented. Therefore, suggestions related to the functional part of the platform were already expected. Finally, tutorials in GrameStation to facilitate the comprehension of how to play the games were suggested both by participants with previous experiences in GG and by participants with no knowledge.

7. Conclusions

This paper presented a user experience and usability test of GrameStation: a game engine based on GG. In this first moment, considering only the running of games, the participants did not report difficulty in using GrameStation, but in understanding how the games should be played (applying rules). The feedback provided by participants – such as to save the game progress, go back to the last action executed in the game, and allocate character movement to keyboard controls – will be considered to improve the platform. The inclusion of tutorials, the complement to the visual feedback already provided by the tool, and a hint menu for those who have difficulty with GG, also suggested by the participants, will be presented by the already proposed pedagogical agents [Silva et al. 2021]. Finally, for future work, we want to organize a second activity to evaluate the support given by GrameStation in creating/specifying a game as GG. Thus, the pedagogical agents can be implemented, and later we can also test their effectiveness. Additionally, we aim to test the tool with other groups, such as adults with no CS background and children.

References

DeVellis, R. F. (2003). Scale development: Theory and applications. *Sage London*, 13:0–176.

Ehrig, H., Heckel, R., Korff, M., Löwe, M., Ribeiro, L., Wagner, A., and Corradini, A. (1997). Algebraic Approaches to Graph Transformation. Part II: Single Pushout Approach and Comparison with Double Pushout Approach. In *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1: Foundations*. World Scientific Publishing Co., Inc.

Karakasis, C. and Xinogalos, S. (2020). BlocklyScript: Design and Pilot Evaluation of an RPG Platform Game for Cultivating Computational Thinking Skills to Young Students. *Informatics in Education*, 19(4):641–668.

Nielsen, J. (1994). Usability Inspection Methods. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, page 413–414, New York, NY, USA. Association for Computing Machinery.

Nielsen, J. and Landauer, T. K. (1993). A Mathematical Model of the Finding of Usability Problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 206–213, New York, NY, USA. Association for Computing Machinery.

Petri, G., von Wangenheim, C. G., and Borgatto, A. F. (2016). MEEGA+: An Evolution of a Model for the Evaluation of Educational Games. *INCoD/GQS*, 3:1–40.

Ribeiro, L. (2000). Métodos Formais de Especificação: Gramática de Grafos. *Escola de Informática da SBC-Sul*, 8:1–33.

Silva, J. V., Silva Junior, B. A., Cavalheiro, S. A. C., and Foss, L. (2021). Gramers: Agentes Pedagógicos para uma plataforma de jogos baseada em Gramática de Grafos. In *Anais do VI Workshop-Escola de Informática Teórica*.

Silva Junior, B. A., Cavalheiro, S. A. C., and Foss, L. (2017). A Última Árvore: exercitando o Pensamento Computacional por meio de um jogo educacional baseado em Gramática de grafos. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, pages 735–744.

Silva Junior, B. A., Cavalheiro, S. A. C., and Foss, L. (2021). GrameStation: Specifying Games with Graphs. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, pages 499–511, Porto Alegre, RS, Brasil. SBC.

Sousa, L. L., Farias, E. J. P., and Carvalho, W. V. (2020). Programação em Blocos Aplicada no Ensino do Pensamento Computacional: Um Mapeamento Sistemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, pages 1513–1522.

Souza, F. F. and Nunes, M. A. S. N. (2019). Práticas e resultados obtidos na aplicação do Pensamento Computacional Desplugado no ensino básico: Um Mapeamento Sistemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, volume 30, pages 289–298.

Technologies, U. (2022). Unity real-time development pltaform – 3D, 2D, VR, and AR engine. Available at: `http://unity.com/`. Accessed: 2022-03-08.

Tsopra, R., Courtine, M., Sedki, K., Eap, D., Cabal, M., Cohen, S., Bouchaud, O., Mechaï, F., and Lamy, J.-B. (2020). Antibiogame®: A serious game for teaching medical students about antibiotic use. *International Journal of Medical Informatics*, 136:104074.

Venigalla, A. and Chimalakonda, S. (2020). G4d - a treasure hunt game for novice programmers to learn debugging. *Smart Learn. Environ.*, 7:104074.

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