A Math Educacional Computer Game Using Procedural Content Generation

Luiz A. L. Rodrigues[1], **Robson P. Bonidia**[2], **Jacques D. Brancher**[1]

[1] Department of Computer Science – Londrina State University
10.011 – 86.057-970 Londrina, PR – Brazil

[2]IT Department – Federal Technological University of Paraná
Cornélio Procópio, Paraná, Brazil

Abstract. *This paper presents an educational game, MentalMath, with the goal of motivating its users to practice math by solving problems. It uses, besides gamification, procedural content generation, which aims to increase the game's scenarios and problems diversity, tackling some gamification limitations. A preliminary study was conducted with 40 students from a Brazilian school with pre and post written tests, and gameplays data. The results demonstrated that our tool aroused the volunteers' interest in math, had a higher impact on the ones which played for longer and imposed hard scenarios. We conclude that MentalMath is a reliable tool to motivate children, that our study gave interest insights on its usability and our approach innovates using procedural content generation.*

1. Introduction

According to the National Curricular Parameters of Brazil [Nacionais 1998], math education is an important element in the citizens' development. However, math is not only focused on formulas, static and unquestionable concepts. Its focuses on the improvement of social and cultural relationship, entrance to the labor market and social context. Furthermore, we might state that brazilian elementary school faces difficulties, supported by its 66º place in 70 countries analyzed on Programme for International Student Assessment [Gurria 2016].

Despite the technology advance, it is notable that there are yet several aspects to be investigated on the elementary school. One of them, is gamification, the use of game elements in daily context [da Silva Marinho et al. 2016]. Its goal is to provide opportunities that aid to minimize motivation and engagement problems of the students. Note that gamification does not seek to teach through games, but to use game mechanics as a way to develop the student stimulation and involvement [Huang and Soman 2013]. It might be argued that it offers a collaborative and playful dynamics, stimulate coordination skills, concentration and logical reasoning. These are essential skills to the learning success, primarily in exact sciences [da Rocha Seixas et al. 2014]. However, gamified tools yet have the limitation of being repetitive and boring [Boulet 2016], which might be dealt using Procedural Content Generation (PCG).

PCG aims to automatically generate an output [Linden et al. 2013], such as trees (SpeedTree[1]), buildings [Rodrigues et al. 2015] or scenarios [Khalifa et al. 2016], where

[1] www.speedtree.com

DOI: 10.5753/cbie.sbie.2017.756

games has been one of its largest field of application and testbed [Hendrikx et al. 2013, Carli et al. 2011]. One of the main characteristics of a PCG technique is to create diverse outputs, leading to a certain level of innovability at each time. Furthermore, it might allow the user to influence its outputs through parameters or configurations [Doran and Parberry 2010], biasing the outcomes towards the desired result. Examples would be to define a difficulty level, the type of a tree or the number of scenarios to be generated.

An implementation of a PCG technique into an application might be defined as PCG-System (PCG-S) [Shaker et al. 2016]. Therefore, a PCG-S can be seen as a resource, which is a black box to the other systems that use it. For instance, the system employed in MentalMath is a PCG-S. MentalMath is a gamified tool to motivate children to practice math, through an adventure game which requires the player to solve a math problem, in order to advance to the next level. This approach differs by its use of PCG in an educational context, improving the diversity presented by the game, which automatically creates its scenarios and math problems.

Thus, the main goal of this paper is to introduce MentalMath, its PCG-S and to conduct a preliminary usability test, with the aim to identify its practical issues and insights of its benefits. The remaining of this paper is arranged as follows. Sections 2.1 and 2.2 presents a background for Educational Games and PCG, respectively. Section 3 introduces the developed tool, followed by the materials and methods used to evaluate it, described in Section 4. Finally, we demonstrate the achieved results and discuss them in Section 5, and our conclusions in Section 6.

2. Related Work

This section presents relevant research related to the main topics of this paper. First, we approach educational games used in different education subjects, and then some techniques used to procedurally generate different types of content.

2.1. Educational Games

Educational Games (EG) might be applied in distinct levels of teaching, such as preschool, graduation and specialization [Savi and Ulbricht 2008]. It demonstrated to have the ability to entertain people and encourage them to learn through interactive and dynamic environment [Hsiao 2007]. Furthermore, it provide interest and urge students with challenges, curiosity and lucidity [Balasubramanian and Wilson 2007]. Among the EG benefits, there are the motivating effect, facilitator of learning, development of cognitive abilities and learning by discovery, which resulted in different advantages as a didactic resource on the teaching practice [Savi and Ulbricht 2008]. Moreover, the main methodologies used to analyze these applications are based on questionnaires and evaluation pre and post to the use [Almeida Ferreira et al. 2016].

In [Menezes and da Roza 2016] was developed an app with the goal of supporting the teaching and learning of mathematics in child education, testing it on the classroom. The famous Angry Bird was used in [Moita et al. 2013] to aid with math concepts, specifically of the ninth year of elementary school. Another solution is the Mathematics [Araújo et al. 2016], which assists on the teach of basic math knowledgment. It was applied to students of a public school and was argued that it promoted efficiency and ensured a good learning.

Other researchs [Farias et al. 2016, Toda et al. 2014, da Rocha Seixas et al. 2014, Andrade and Canese 2013] introducing similar proposes, indicated the EG's effectivity to stimulate users engagement, presenting positive results. Although the advantages established on the literature, [Boulet 2016] points out that tools which use gamification concepts might suffer from the lack of interest or become annoying, after it is used for a determined period of time. Yet, the author proposes that this could be solved creating contexts closer to the tool's target users. Given this appointment, we developed MentalMath as an adventure game to attract our target users, approximately 9-years-old children, which uses PCG to tackle the disinterest increasing its diversity.

2.2. Procedural Content Generation

Rogue in 1980 (*Troy and Wichman*) was one of the first games to use PCG, where for each play a new adventure was generated. Furthermore, nowaday games also used these techniques, e.g. Grand Theft Auto IV (*RockStar Games, 2008*) and No Man's Sky (*Hello Games, 2016*), where different algorithms might be applied to procedurally generate a content.

Grammars are used to obtain a restricted output, achieving this through explicit production rules [Font et al. 2016, Karavolos et al. 2015]. Another approach is the Answer Set Programming (ASP) [Neufeld et al. 2015, Smith and Mateas 2011], which allows the design to declare expected output's features, and finds an output respecting them. Search Based Algorithms (SBA) are commonly used to generate scenarios [Togelius et al. 2011, Ashlock and McGuinness 2013], optimizing the generated content towards the desired output.

There are other approaches to generate different types of contents, as presented in [Hendrikx et al. 2013], where the authors survey algorithms for procedurally generate contents specific to the game. However, both problems and scenarios generation presented in this paper uses template based and constructive algorithms, respectively. The latter is based on the one introduced in [Khalifa et al. 2016]. The first follows the notion of a grammar, replacing elements.

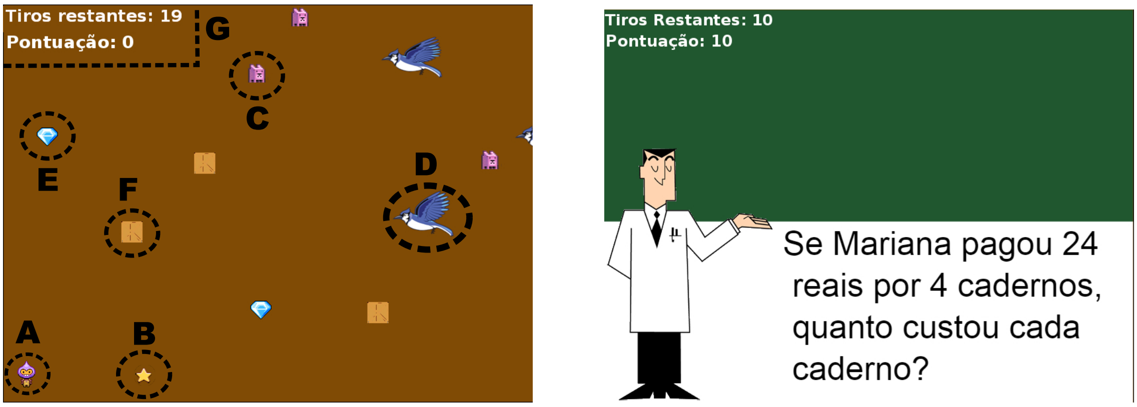
3. MentalMath

This section presents MentalMath[2], a game which aims to aid children to practice its math skills. It was developed using Phaser[3], a desktop and mobile HTML5 framework. Here, we explain MentalMath functionalities and how it generates both its problems and scenarios. Figure 1 presents an example of scenario and problem, which is further explained.

Figure 1(a) demonstrates a randomly selected scenario highlighting each one of the elements which might be present in it with a different letter. *A* is the player's avatar, which is able to shoot stars (*B*). Initially, the player has 20 shots and receives 3 more for each enemy (*C*) killed or diamond (*E*) collected. In case a star collides with a enemy, bird (*D*) or box (*F*) they are instantly killed. Boxes have the goal to hinder the avatar's movement through the scenario, since they do not kill him. At the same time, the player might use these boxes to hide from enemies, but no from birds. Birds and diamonds do not collide on boxes, allowing the birds to fly over it and kill the player and a box to hide

[2] game.rpbtecnologia.com.br
[3] phaser.io

a diamond. Finally, at the screen's top left (*G*) there is the game's status, presenting how many shots the player has (*Tiros restantes*) and its score (*Pontuação*).



Se Mariana pagou 24 reais por 4 cadernos, quanto custou cada caderno?

(a) Scenario's elements

(b) Final challenge

Figure 1. MentalMath examples with increased letter's size to favor the read.

In order to complete each level, the player has to collect every diamond in the scenario, kill all enemies and correctly answer a challenge, presented after the previous requirements are achieved. Furthermore, the avatar can not be touched by an enemy or bird, which would instantly kill it and restart the game. Note that running out of shots before killing every enemies also makes impossible to complete the level.

The player's score starts at 0. If he misses an answer, we choose to use a small decrease (-5), in order to do not discourage the player. Firing a correct shot on an enemy positively rewards (5) the player, since it is a requirement to finish the level. However, this reward is small compared to correctly answer the level's final challenge, which is 100. Furthermore, a negative reward of -20 is received when the player is killed.

The challenge presented at the end of each level is a math problem. These problems encompass basic operations (additions, subtractions, multiplications and divisions) and number comparisons. Every problem used in MentalMath was extracted from Assessment of Process Learning (APL - *Avaliação da Aprendizagem em Processo*)[4], an elementary education test of the state government of São Paulo, Brazil. The APL was also de database which we extracted the problems used as template for the game, and to evaluate our approach (see Section 4).

3.1. Problems Generation

A template based generation was adopted to automatically create math problems at the end of each level. Eight different problems were selected as templates, also from the APL, having problems that address a specific type of operation and general ones. The following list presents two examples translated from Portuguese:

- Help Rafaela to calculate the operation 154 + 49.
- If Mariana paid 20 dollars for 4 notebooks, how much she will pay for 8 notebooks?

[4] npediadema.wordpress.com/documentos-orientadores-aap-1o-bimestre-2016/

Here our goal was not to generate the interpretation problem, but the calculation problem. Following that, every problem was adapted in a way that its numbers can be generated by an algorithm. Furthermore, in the case of our first example, previously presented, the operation to be done might also be algorithmically created. However, as in our second example, some problems text requires that a specific operation is used. Moreover, besides the problem text, each template also carries other properties: an identifier; the amount of numbers in that problem; an indicator if an operation must be generated; a optional fixed operator for that problem; and the biggest number's value allowed. Figure 1(b) demonstrates a generated problem, in Portuguese, as example.

The generation algorithm respect two specific rules, given the game's main target audience. One of them is that, in basic operations, the generated numbers aways appear in decreasing order. This avoids single subtractions to have negative results and divisions which lead to fractional numbers. The other one is that the dividend always must be divisible by the divider, ensuring that all results will be integers. Furthermore, the applied algorithm does not handle problems with different operations (e.g. $2 + 2 * 3$), it only handles multiples operations of the same type (e.g. $2 + 2 + 3$).

3.2. Scenarios Generation

A constructive approach was utilized to generate MentalMath levels, which uses information received from the problem to be solved at the end of the level as parameters. Furthermore, the generation system persists other information about the gameplay to use as generation parameters. For example, how many levels were already played is used to determine the number of harmful elements will be created. In the scope of this paper, we consider each time a level is played a gameplay. Moreover, each scenario is created according to the steps described below.

1. **Pre-processing:** This step selects which problem will be this level's final challenge and updates other generation parameters, such as the number of each harmful elements to be created.
2. **Add Avatar:** This step places the player's avatar at the scenario's bottom left. This is the only element which has a fixed position, in order to make easier for the player to find it.
3. **Add Diamonds:** This step randomly places *n* diamonds. Here, *n* is determined by $an + ops$, where *an* is the amount of numbers in the selected problem. *ops* is 0 if that problem has a fixed operation and 1 otherwise.
4. **Add Enemies and Boxes:** This step randomly place enemies and boxes. Each one of them respects a distance of at least 100 pixels, in both x and y coordinate, of the avatar.

Despite the problem's selection uses a Pseudo-Random Number Generator (PRNG), after one of them is selected, it is stored in a *used list*, avoiding that the same problem is selected during the levels' progression. After this list contains all templates or the game is restarted, it is restored. Moreover, the number of harmful elements is determined by an initial parameter, set to 3 each time the game restarts, which we set to increment after each level is finished. Nevertheless, birds are generated during the game execution, aways being spawned on the right and moving to the left, with an x position also determined by the PRNG. This approach avoids the absence of them in case a player takes too long to complete the level. Also, it maintains a certain level of challenge even if the player kills every enemy before collecting all diamonds.

4. Materials and Methods

This section begins presenting how the process of data collection was conducted, and then it describes the experiments performed. All data reported in this paper was collected by a total of 40 students of the school *Bom Pastor*, in *Quatigua*, *Paraná*, Brazil. These volunteers are from the fourth year of elementary education, therefore, every step described here had the school's principal authorization and was supervised by the class's teacher. Moreover, no personal information about any student was exposed.

4.1. Data Collection

This process was accomplished through three steps in two different ways. The first and last step gathered information through regular written tests. The second one used MentalMath gameplays to collect the following information: which question was presented at the end of the level; how many levels was played until that point; whether the player died or not in that gameplay; how much time the player took to finish the level; how much time the player took to correctly answer the level's challenge; how many times the player tried an answer to the challenge; and how many shots the player fired.

4.2. Experiments

Two main experiments were conducted to evaluate MentalMath: a gameplays observation and a written tests comparison based on [Almeida Ferreira et al. 2016]. These written tests are similar to a regular school test, where the student answers questions, writing on a piece of paper.

The gameplays observation had the participation of the entire sample. It was conducted presenting MentalMath to small groups of volunteers and providing approximately 30 minutes for them to play, supervised by a research member, besides its teacher. This research member was responsible for explaining how the game works, present the commands used to play it and to answer any questions the students had during the experiment. At each gameplay, the following features were collected: time spent playing the scenarios and solving the challenges; the number of attempts to answer each problem; how many shots were fired; and their score at the end of each one. Moreover, the data gathered here were analyzed using the Turkey's five number summary from [R Core Team 2014], which presents the minimum, lower-hinge, median, upper hinge and maximum for a numeric set. In addition, the mean was also analyzed.

A total of 29 volunteers from our sample were present in both applications of the written tests, where the first was previously to the experiment described above and the other after that. Both written tests are available for consultation[5]. Here was evaluated if MentalMath application would enhance a class' short-term performance, that played for a longer time in comparison to another, on written tests. Thereby, regardless of the student's skills, we analyze how their performance changed, and not who achieved the highest scores. It used this sample divided into two groups named *X* and *Y*, with a size of 14 and 15, respectively. These groups follow the division of the school, which has

[5] https://drive.google.com/open?id=0B5D9MH0sMguKTmpNWVhRNkVNdEk

two third year class. Group *X* played for 20 minutes on average, while the other played for 40 minutes on average. This experiment was achieved using the Walch Two Sample t-test, also from [R Core Team 2014]. Its parameters were the resultant vector from the differences between scores of the second and first written tests for groups *X* and *Y*. Note that here we are testing how much their scores, on the written tests, changed between the pre and post test. Also, we used the default confidence level of 95%.

5. Results

The gameplays observation resulted in 1044 gameplays, achieving a total of 286 final challenges answered. Therefore, only 27% of the gameplays reached the final challenge. This result aroused two main insights. One is that the configuration we used on the scenarios generation created levels too hard for our sample. The second one is that the volunteers had a short time between the moment they were presented to the tool and starting to play. Therefore, this experiment allowed us to identify a flaw on our tool's playability according to our sample.

Although the majority of losses, over 16 hours were played on the scenarios and 5.8 hours spent answering the challenges. An average of 24,86 scenarios, during approximately 55 seconds, was played by each player. Also, solving the final challenge took approximately 73 seconds on average, and the number of shots fired in each gameplay had an average of 7.67, as can be seen in Table 1. It presents the Tukey's five number summary and mean for the time volunteers spend playing the scenarios and solving the challenges (both in seconds), the amount of shot they fired and their score at the end of each gameplay. Nevertheless, since the number of elements in each scenario is aways higher than 4, excluding the birds which might spawn unexpectedly, we argue that the possibility of firing shots did not influence the students, once that this average is near to the minimum required to complete the scenarios.

Table 1. This table presents the Tukey's five number summary and means for four gameplay attributes. Time playing and answering are both measured in seconds.

Attribute	minimum	lower-hinge	median	upper-hinge	maximum	mean
Time playing	1,46	14,49	27,53	66,82	1979,27	55,24
Time solving	9,55	34,41	53,55	90,34	549,18	73,37
Shots	0	3	6	11	44	7,67
Sum Score	-700	-110	90	437,5	2160	212,18

However, during the written tests was notable that the volunteers did not demonstrate enthusiasm solving the theoretical activity. In addiction, most of them presented mastery of math operations, however, could not apply this knowledge when they had to interpret math problems. Counterwise, on the gameplays observation the volunteer's behavior showed their interest to participate. Also, the classe's teachers stated that MentalMath aroused interest in students which did not have it when the subject was math. Moreover, the gameplays collection allowed to verify the volunteer's engagement solving the problems, since the game enables them to explore, experiment and collaborate, presenting realtime feedbacks and providing an environment free os risks. Nevertheless, this promoted experimentation and exploration, stimulated the curiosity, learning by discovery and perseverance to the users [Balasubramanian and Wilson 2007, Savi and Ulbricht 2008]. Furthermore, the PCG-S differs MentalMath from others gamified tools increasing its diversity, automatically creating a new scenario and problem at each gameplay, tackling the repetitive/boring characteristic of some gamified tools [Boulet 2016]. Hence, we argue that our game demonstrated to be reliable motivating the practice of math.

Finally, the comparison of samples X and Y using the *Two-Sided* alternative of t-test rejected the null hypothesis with a *p-value* < 0.05, confirming a difference in the means of the volunteer's performance. Also, we compared them with the *Less* alternative, which led to a similar result, confirming a difference less than 0 between their means. Finally, using the alternative *Greater*, the test accepted the null hypothesis with a p-value of 0.98, confirming that X means was not greater than Y. These results allow us to argue that the group Y, which were exposed to MentalMath for a longer time, achieved a higher increase on the written tests. Thus, although the short period the students used MentalMath, the group which played more achieved better results than the others.

6. Conclusion

This paper presented MentalMath, a gamified tool to motivate children to practice math. It provides an adventure game based on scenarios, where at the end of each one the player must solve a math problem to advance to the next scenario. Both game's scenarios and problems are procedurally generated in real time, using a constructive and template based approach, respectively. We evaluated our approach with a preliminary usability test at an elementary school from Brazil. The experiments were conducted with 40 volunteer students of fourth year during a two-week period. These analysis was based on pre and post use written tests, and data collected on gameplays.

The results demonstrated that the group which played the most had a higher increase in its written test's score. This suggest that using our tool for longer might improve the user's hability to solve the problems. Also, although the majority of losses while playing, the experiments demonstrated that its use promoted enthusiasm for them to pratice math, confirmed by the children's teacher, which supervised the experiment. We also argue that MentalMath differs from others gamified tools by its use o PCG. This increased its scenarios diversity, with a new level at each time it is played. Furthermore, automatically generating math problems also required higher efforts from the players in their math skills, despite our game only used 8 problem templates. Therefore, we conclude that MentalMath achieved its main goal, demonstrating to be a reliable tool to motivate the study of math.

As the main direction of future works we pretend to adapt our scenarios generation approach, so that MentalMath provide easily gameplays, and thus, induce the player to solve more math problems. Also, we suggest that the use of a system which dynamically updates the game progression's difficulty, rather than the pre-defined schema used in ours, could tackle this problem in game time. Moreover, since we presented a preliminary evaluation over our approach, we pretend to perform further researchs with more volunteers from different places, and a specific analyze over the effects of PCG on gamification. This should give more concrete insights about which positive and negative effects both of them might provide.

Acknowledgements

We are very thankful to the Education Department of the Municipality of Quatiguá, and from Bom Pastor Municipal School, for allowing us to conduct the experiments.

References

Almeida Ferreira, A., Werneck, V., and Santos, N. (2016). Avaliação da aprendizagem em ambientes educacionais: Uma revisão sistemática. In *SBIE 2016*, volume 27, page 179.

Andrade, J. and Canese, M. (2013). Elementar: Aplicando gamificação ao processo de ensino-aprendizagem da lógica formal. *Conferencias LACLO*, 4(1).

Araújo, J. P. P., Costa, G., and Júnior, J. G. R. (2016). Matematech: Plataforma de apoio à aprendizagem de matemática nos anos iniciais do ensino fundamental. In *SBIE 2016*, volume 27, page 110.

Ashlock, D. and McGuinness, C. (2013). Landscape automata for search based procedural content generation. In *2013 CIG*, pages 1–8.

Balasubramanian, N. and Wilson, B. G. (2007). Learning by design: Teachers and students as co-creators of knowledge. *Educational Technology: Opportunities and Challenges, Oulu, Finland: University of Oulu*, 30:51.

Boulet, G. (2016). Gamification is simply bells and whistles. *eLearn*, 2016(11).

Carli, D., Bevilacqua, F., Pozzer, C., and d'Ornellas, M. (2011). A survey of procedural content generation techniques suitable to game development. In *SBGAMES 2011*, pages 26–35.

da Rocha Seixas, L., Gomes, A. S., Melo Filho, I. J., and Rodrigues, R. L. (2014). Gamificação como estratégia no engajamento de estudantes do ensino fundamental. In *SBIE 2014*, volume 25, page 559.

da Silva Marinho, A., Melo, A. V. C., Poggi, G. H., Kosiur, M. B., Marrane, W. R., and Boghi, C. (2016). Mathematics of mobile application in basic education for teaching children of fundamental i 1st to 3rd year. *Research, Society and Development*, 3(1):69–90.

Doran, J. and Parberry, I. (2010). Controlled procedural terrain generation using software agents. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(2):111–119.

Farias, M., Gasparini, I., and Hounsell, M. (2016). Jogos digitais educacionais para alfabetização matemática: Levantamento de habilidades e level design. In *SBIE 2016*, volume 27, page 430.

Font, J. M., Izquierdo, R., Manrique, D., and Togelius, J. (2016). *Constrained Level Generation Through Grammar-Based Evolutionary Algorithms*, pages 558–573. Springer International Publishing, Cham.

Gurria, A. (2016). Pisa 2015 results in focus. *PISA in Focus*, (67):1.

Hendrikx, M., Meijer, S., Van Der Velden, J., and Iosup, A. (2013). Procedural content generation for games: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1):1:1–1:22.

Hsiao, H.-C. (2007). A brief review of digital games and learning. In *DIGITEL'07*, pages 124–129. IEEE.

Huang, W. H.-Y. and Soman, D. (2013). Gamification of education. *Research Report Series: Behavioural Economics in Action, Rotman School of Management, University of Toronto*.

Karavolos, D., Bouwer, A., and Bidarra, R. (2015). Mixed-initiative design of game levels: integrating mission and space into level generation. In *FDG 2015*.

Khalifa, A., Liebana, D. P., Lucas, S. M., and Togelius, J. (2016). General video game level generation. In *2016 GECCO*, pages 253–259.

Linden, R. v. d., Lopes, R., and Bidarra, R. (2013). Designing procedurally generated levels. In *IDPv2 2013 - Workshop on Artificial Intelligence in the Game Design Process*, pages 41–47, AAAI Press, Palo Alto, CA. AAAI, AAAI Press. ISBN 978-1-57735-635-6.

Menezes, S. and da Roza, J. (2016). Genius math: uma aplicação mobile para auxiliar a aprendizagem da matemática na pré-escola. In *SBIE 2016*, volume 27, page 250.

Moita, F., Luciano, A., Costa, A., and Barboza, W. (2013). Angry birds como contexto digital educativo para ensino e aprendizagem de conceitos matemáticos: relato de um projeto. *SBC–Proceedings of SBGames*.

Nacionais, I. a. P. c. (1998). terceiro e quarto ciclos do ensino fundamental. *Brasília: MEC-Secretaria de Educação Fundamental*.

Neufeld, X., Mostaghim, S., and Perez-Liebana, D. (2015). Procedural level generation with answer set programming for general video game playing. In *2015 CEEC*, pages 207–212.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rodrigues, F. C. M., Neto, J. B. C., and Vidal, C. A. (2015). Split grammar evolution for procedural modeling of virtual buildings. In *SVR 2015*, pages 75–83.

Savi, R. and Ulbricht, V. R. (2008). Jogos digitais educacionais: benefícios e desafios. *Revista Novas Tecnologias na Educação*, 6(2):10.

Shaker, N., Togelius, J., and Nelson, M. J. (2016). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.

Smith, A. and Mateas, M. (2011). Answer set programming for procedural content generation: A design space approach. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):187–200.

Toda, A. M., do Carmo, R. S., Neto, J. C., Silva, A. L., and Brancher, J. D. (2014). Desenvolvimento de uma aplicação web para auxiliar no ensino da matemática para alunos do ensino fundamental. In *SBIE 2014*, volume 25, page 392.

Togelius, J., Yannakakis, G., Stanley, K., and Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186.