

Revisitando um Jogo Educacional para desenvolver o Pensamento Computacional com Gramática de Grafos *

Braz Araujo da Silva Junior¹, Simone André da Costa Cavalheiro¹, Luciana Foss¹

¹Programa de Pós-Graduação em Computação - Universidade Federal de Pelotas
CEP 96.010-610 - Pelotas - RS - Brazil

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

Abstract. *This paper presents a digital version of an educational game, to make feasible the data gathering and recording, which shall fuel more detailed evaluations. For this, we reexplore a game that uses Graph Grammars to develop the Computational Thinking. We run 6 validating plays, from which data is gathered to exemplify the automation and its subsequent evaluation, which is based on this data. We conclude that the digital version, besides granting the data for a better evaluation, also reinforces fundamental concepts of Graph Grammars.*

Resumo. *Este artigo apresenta a versão digital de um jogo educacional, com o intuito de viabilizar a coleta e o registro de dados para fomentar avaliações mais detalhadas. Para isto, reexplora-se um jogo que utiliza Gramáticas de Grafos para desenvolver o Pensamento Computacional. É realizada uma aplicação de 6 jogos para validação, de onde são retirados dados para exemplificar a automatização e a avaliação subsequente, que é baseada nestes dados. Conclui-se que a versão digital, além de fornecer os dados para uma melhor avaliação, ainda reforça conceitos fundamentais das Gramáticas de Grafos.*

1. Introdução

O conjunto de habilidades ligadas à área da ciência da computação que auxiliam no processo de resolução de problemas tem sido extensamente explorado no meio acadêmico. Tais habilidades, ditas parte do Pensamento Computacional (PC), começaram a se popularizar com um influente artigo de Jeannete Wing [Wing 2006], onde foi ressaltada a importância de promover estas habilidades em todos, não somente em cientistas da computação. Para tal, é recomendado que o desenvolvimento do PC seja iniciado já nos primeiros anos de aprendizado [Barr and Stephenson 2011].

Apesar das iniciativas práticas para desenvolver o PC serem recentes, o Brasil vem avançando na área com uma expressiva quantidade de publicações [Ortiz and Pereira 2018]. Estas iniciativas variam significativamente em suas abordagens, por exemplo, estimulando o PC por meio de: robótica [Silva and Javaroni 2018]; jogos digitais [Melo et al. 2018]; e atividades desplugadas [Guarda and Goulart 2018].

Entretanto, todos estes esforços enfrentam uma mesma questão: a avaliação. Existe um número de obstáculos para avaliar o PC, desde a falta de um consenso sobre

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Projeto realizado com o apoio da PREC e PRPPG / UFPel.

a definição operacional, até a complexidade de avaliar construtos, habilidades cognitivas que não podem ser mensuradas diretamente [Araujo et al. 2018]. Grande parte dos trabalhos que se propõem a fazer tal avaliação se apoiam em instrumentos de avaliação qualitativa por meio de observações, e quantitativa por pré e pós-testes [Avila et al. 2017].

“A Última Árvore” [Silva Junior et al. 2017] é um destes trabalhos, um jogo educacional que se apoia nas relações entre o PC e as Gramáticas de Grafos (GGs), um formalismo matemático de caráter visual. Exatamente por meio de observações, pré- e pós-testes, Silva Junior et al. [2018] expõe detalhadamente relações entre aspectos do jogo e habilidades do PC, além de tecer uma avaliação de desempenho dos alunos. Estes trabalhos utilizam uma definição do PC que inclui as habilidades de: coleta de dados, análise de dados, representação de dados, decomposição de problemas, abstração, pensamento algorítmico, simulação e paralelismo [CSTA et al. 2009].

Considerando o caráter dessas habilidades cognitivas envolvidas no PC e o desafio de mensurá-las diretamente, este trabalho visa fornecer ferramentas para robustecer a avaliação. Procura-se destacar a avaliação de competências relacionadas às GGs, uma vez que estas podem ser mensuradas de forma mais direta e possuem, por sua vez, expressivas relações com o PC, descritas em [Silva Junior et al. 2017]. Para isto, apresenta-se uma versão digital do jogo “A Última Árvore”, justificada pela possibilidade de automatizar a aquisição e o registro dados críticos para uma avaliação robusta de eventuais aplicações.

O artigo está organizado como segue. A seção 2 introduz o modelo de GGs e revisita suas relações com o PC. A seção 3 apresenta o jogo “A Última Árvore” em sua versão física e descreve as similaridades e diferenças em relação à sua versão digital. Na seção 4 são discutidas as informações coletadas automaticamente durante a execução do jogo e como elas auxiliam no processo de avaliação. Na seção 5 é exposta uma aplicação inicial para a validação da implementação e os resultados provenientes dela. A seção 6 relata as impressões e percepções sobre a aplicação, por parte dos alunos e tutores. A seção 7 conclui o artigo discutindo os próximos passos deste projeto.

2. Gramáticas de Grafos e o Pensamento Computacional

Gramática de Grafos (GG) é um formalismo matemático intuitivo que permite representar um sistema e seu comportamento utilizando grafos. Seguindo a dinâmica das gramáticas de Chomsky, tem-se um estado corrente (representado por um grafo-estado), e um conjunto de eventos que podem alterar este estado (representado por regras de transformação de grafos) [Ehrig et al. 1997]. Estes eventos são representados por dois grafos: uma condição (representada pelo grafo de lado esquerdo da regra, ou LHS); e uma consequência (representada pelo grafo de lado direito da regra, ou RHS). Superficialmente, esta condição é satisfeita quando é possível mapeá-la para o estado corrente, isto é, relacionar cada elemento do LHS a um elemento do grafo-estado. Este mapeamento é chamado de *Match*. Os eventos ocorrem sobre os *matches*, transformando os elementos mapeados em uma nova configuração, a descrita pela consequência do evento (o RHS).

O funcionamento das GGs pode ser mais profundamente entendido em [Ribeiro 2000] e a especificação do jogo em GGs é dada em [Silva Junior et al. 2017]. Este artigo limita-se a esta explicação por considerá-la suficiente para ressaltar pontos da GG que podem ser relacionados ao PC. A Tabela 1 a seguir traz as relações das habilidades do PC com a GG, aprofundando a análise exposta em [Silva Junior et al. 2017].

Tabela 1. Revisão das relações do PC com GG

Habilidade do PC	Relação com as Gramáticas de Grafos
Coleta de Dados	A disposição dos elementos como vértices e relações entre eles confere um caráter de simultaneidade, o que permite coletar a informação de maneira emaranhada. Em contraste com linguagens textuais, que possuem caráter sequencial.
Análise de Dados	Os mapeamentos entre elementos, intrínsecos ao funcionamento de GG, estimulam a projeção e ponderamento sobre a informação e como ela se transforma.
Representação de Dados	As GG sintetizam sistemas de variadas complexidades a um modelo universal de representação muito simples: um estado e um conjunto de eventos para alterá-lo.
Decomposição de Problemas	A simplicidade da dinâmica das regras de transformação de grafos (uma condição e uma consequência) direciona a modelagem de processos complexos para sua decomposição em tarefas mais simples.
Abstração	O aspecto local das aplicações de regras sobre os matches, que especificam como alvo um subgrafo do grafo-estado, infere constantemente a abstração do restante do grafo-estado.
Algoritmos e Processos	Cada regra de transformação de grafos representa uma alteração atômica no grafo-estado. Portanto, podem ser apresentadas como passos bem definidos para a construção de algoritmos.
Simulação	A base de uma simulação é a descrição do estado inicial de um sistema e o comportamento de seus agentes. Há portanto, uma relação 1:1 com a base das GGs: grafo-estado e conjunto de regras.
Paralelismo	Não há relação de ordem entre regras de transformação de grafos a menos que esta seja criada indiretamente por meio da descrição de, respectivamente, criação e consumo de algum recurso nas regras antecessora e sucessora.

3. Versão Física x Versão Digital

Originalmente um jogo de tabuleiro, de estratégia baseada em turnos, “A Última Árvore” faz seus jogadores manipularem uma GG para interpretar animais tentando restaurar sua floresta, que fora desmatada, restando uma única árvore [Silva Junior et al. 2017]. A base do jogo é apresentar um grafo-estado com animais, blocos de terra e recursos (frutas, sementes, plantas e árvores) que podem ser criados ou consumidos ao longo do jogo. Cada jogador tenta alcançar um objetivo diferente, aleatório e descrito por um grafo. Por exemplo, a carta ilustrada na esquerda da Figura 1 mostra o objetivo de conseguir um animal com 5 frutas. Após terem seus objetivos em mãos, cada jogador realiza uma ação e passa a vez para o próximo, até que alguém atinja seu objetivo e vença o jogo. As ações possíveis são regras de transformação de grafos que descrevem:

- r1** como movimentar um animal de um bloco de terra para outro;
- r2** como gerar uma fruta a partir de uma árvore;
- r3** como usar uma fruta para conseguir uma semente;
- r4** como usar uma semente para conseguir uma planta;
- r5** como usar uma fruta e uma planta para conseguir uma árvore; e
- r6** como destruir 2 frutas;

Maiores detalhamentos sobre o funcionamento do jogo podem ser lidos em [Silva Junior et al. 2017, Silva Junior et al. 2018]. Este artigo foca-se em descrever as diferenças e similaridades entre as versões física e digital.



Figura 1. Versões física e digital de “A Última Árvore”.

O desenvolvimento, fundamentação, metodologia e componentes da versão física do jogo são apresentados em [Silva Junior et al. 2017] (protótipo) e [Silva Junior et al. 2018] (versão final). Já a versão digital, foi desenvolvida utilizando o motor de jogos Unity [Technologies 2019] tentando preservar a identidade visual da versão física. Não só a identidade visual foi preservada, a versão digital apresenta os mesmos elementos, mesma jogabilidade, ações possíveis, quantidade de jogadores e objetivos (com algumas exceções que tiveram elementos adicionados ou removidos para fins de balanceamento). Adicionalmente, algumas regras tiveram seus nomes alterados: **r3** era chamada de **plantar**, no entanto não envolvia plantas, o que confundia os alunos, portanto foi trocada para **comer**, se referindo ao animal comer a fruta e deixar cair suas sementes; **r4** era chamada de **regar** e, instantaneamente os alunos questionavam sobre água, que nem é um elemento existente no jogo, para evitar a confusão seu nome foi trocado para **plantar**, se referindo a plantar as sementes para então nascer uma planta; e **r6** foi trocada para **devorar** por que **comer** agora estava sendo usado para **r3**.

Na versão física, para realizar uma ação os jogadores devem manipular as cartas do jogo de maneira que represente corretamente as consequências das regras que estão aplicando. Embora desejável e estimulado, não é obrigatório que os alunos explicitem ou mesmo passem pelo processo do *match* (encontrar os elementos do LHS no grafo-estado), basta que eles compreendam a semântica da regra e não a desobedeça. Na versão digital, para realizar uma ação os jogadores devem escolher uma regra, e então mapear elemento a elemento o LHS dela para poder aplicá-la. Por exemplo, na próxima página, a Figura 2 ilustra os passos de um jogador para mover um urso para o lado, onde o jogador:

1. clicou no animal no LHS;
2. clicou no urso, mapeando então animal→urso (o contorno verde indica que já estão mapeados);
3. clicou no bloco de terra abaixo do animal no LHS (o contorno azul indica que ele está selecionado para ser mapeado);
4. precisa clicar no bloco de terra abaixo do urso (qualquer outro bloco de terra violaria a preservação de origem e destino das arestas, resultando num *match* inválido);
5. depois no outro bloco de terra do LHS; e
6. por fim, num bloco de terra adjacente ao do urso.



Figura 2. Encontro de um *match* durante o jogo.

Assim, a versão digital trabalha muito mais explicitamente com a questão dos mapeamentos nas GGs que a versão física, na qual isto pode ser eventualmente negligenciado. Com este novo foco nos *matches*, foram adicionadas duas funcionalidades: limite de tempo e limite de erros por turno. Realizar qualquer mapeamento que invalide um *match* cancela todos os mapeamentos correntes e conta como um erro para o jogador. Nesta aplicação foram definidos como limite de tempo e erros, respectivamente, 60 segundos e 5 erros. Ao atingí-los, perde-se a vez, tendo que passá-la para o próximo jogador.

Essa explicitação dos mapeamentos foi uma consequência positiva inesperada da versão digital, cuja principal justificativa para o desenvolvimento é a automatização da coleta de dados para uma avaliação mais robusta. Apesar de trazerem análises e discussões significativas e de serem extensamente utilizadas, é razoável considerar que meras observações informais dos tutores durante as aplicações possuem sérias limitações. Portanto, com a finalidade de fornecer recursos para robustecer a avaliação, apresenta-se o esforço de automatizar este processo de observação por meio de coleta e registro de dados durante o próprio jogo em sua versão digital.

4. Avaliação GG-PC

Cada jogada na versão digital gera um log que detalha todos os eventos ocorridos durante a execução, desde requisições de interface, como alterações de *zoom*, até as seleções de elementos para o mapeamento dos *matches*. Os logs podem ser analisados diretamente caso seja necessário verificar algum detalhe, mas cada partida gera um relatório próprio, resumindo as informações mais importantes por jogador: **número de acertos**, elementos corretamente mapeados; **número de erros**, mapeamentos que invalidam o *match*; **relatório de erros**, relação do número de erros cometidos por classe de erro, conforme a classificação da Tabela 2; **algoritmo**, sequência de aplicações de regras realizadas pelo jogador; **relatório de turnos**, relação do número de aplicações de cada regra, e de turnos perdidos por limite de tempo, por limite de erros ou passados voluntariamente; e **relatório de tempo**, melhor e pior tempo gasto para um turno, tempo médio gasto em um turno e tempo total jogado por este jogador.

Tabela 2. Classificação de erros de mapeamento.

Classe	Definição	Por quê é um erro?
Remapping	Iniciar o mapeamento de um elemento já mapeado	As definições de GG que o jogo utiliza estabelecem que os <i>matches</i> são compostos por funções injetoras, sendo assim, cada elemento de um grafo só pode ser associado a um único elemento de outro grafo e vice-versa.
Overmapping	Concluir o mapeamento em um elemento já mapeado	
Meta-Mapping	Mapear um elemento em outro elemento do mesmo grafo	O <i>match</i> deve ser um mapeamento do LHS no grafo-estado, portanto domínio e imagem são conjuntos compulsoriamente diferentes, ainda que eventualmente equivalentes.
Invalid Value	Mapear um atributo cujo valor não satisfaz a expressão do outro, ou cuja expressão não é satisfeita pelo valor do outro	Atributos em regras são definidos por expressões matemáticas no lugar de valores. Elas só podem ser mapeadas para valores que as satisfaçam. Podem ser irrestritas como uma variável, ou podem aparecer em condições descritas por equações ou inequações.
Attribute Link-A	Mapear um atributo que pertence a um vértice já mapeado, em um atributo que pertence a outro vértice, cujo mapeamento não existe ou não corresponde ao primeiro.	A definição de atributo inclui uma função que mapeia a qual vértice o atributo pertence. Em um mapeamento, estas funções devem ser preservadas, isto é, devem comutar de tal forma que: o atributo A que pertence a um vértice V seja mapeado em um atributo B que pertence a um vértice W, e V é mapeado em W. O mesmo acontece com as arestas, elas possuem funções que mapeiam origem e destino e devem ser preservadas da mesma forma.
Attribute Link-V	Mapear um vértice que possui um atributo já mapeado, em um outro vértice que não possui nenhum atributo cujo mapeamento corresponda ao primeiro.	
Edge Link	Mapear um vértice que possui uma aresta com outro vértice já mapeado, em um vértice que não possui nenhuma aresta com um vértice cujo mapeamento corresponda ao primeiro.	
Attribute Type	Mapear um atributo de um tipo em um atributo de outro tipo.	Em uma GG tipada, como é o caso do jogo, existe um grafo-tipo, para onde são mapeados todos os elementos existentes nos demais grafos da gramática. Mapeamentos entre elementos tipados devem preservar suas tipagens.
Vertex Type	Mapear um vértice de um tipo em um vértice de outro tipo.	
Attribute-Vertex	Mapear um vértice em um atributo, ou o contrário.	Vértices só podem ser mapeados em vértices, o mesmo para atributos e arestas.

5. Aplicação

Para validar a versão digital do jogo, foram realizadas 6 partidas com diferentes grupos de alunos: 3 deles formados por 4 pessoas, 2 com 3 jogadores e um com somente 2 alunos. A aplicação se deu no contexto de um evento de aprendizagem criativa. Participaram 20 alunos, com idades entre 8 e 17 anos, de 6 escolas de Pelotas e Jaguarão: E.E.E.M. Coronel Pedro Osório; Colégio Estadual Getúlio Vargas; E.E.E.F. Joaquim Caetano da Silva; E.M.E.F. Cecília Meireles; Colégio Gonzaga; e I.E.E. Assis Brasil.

Devido a limitações de algumas das aplicações, uma série de adaptações foram estabelecidas. Com o grupo dos 2 alunos mais jovens (menos de 10 anos), o jogo seguiu a abordagem cooperativa, com ambos tentando alcançar o mesmo objetivo, ao contrário da proposta padrão do jogo, de cada um tentar atingir o seu. Nos grupos com 3 alunos, o turno do quarto jogador era passado sem fazer nada. Por limitações de tempo, 2 das 6 partidas não chegaram a ser finalizadas, deixando o jogo sem vencedores.

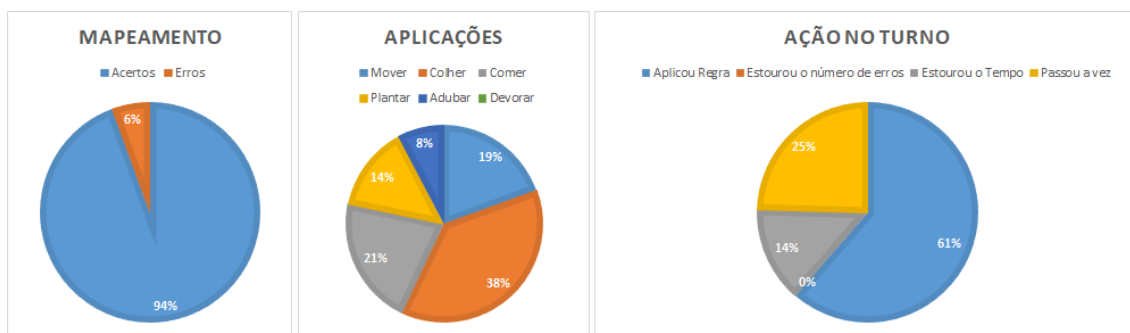


Figura 3. Gráficos sobre as ações realizadas nas 6 partidas.

Os gráficos da Figura 3 resumem as ações dos jogadores, considerando todas as 6 aplicações. Destaca-se a regra devorar (r6) nunca ter sido usada pelos jogadores. Uma possível explicação para isso é ser a única regra de natureza exclusivamente disruptora (para atrapalhar adversários), o que envolve uma estratégia mais profunda que buscar o próprio objetivo isoladamente. Em contra partida, a regra colher (r2) foi disparadamente a regra mais utilizada, mas isso era esperado, já que a fruta é o recurso mais consumido por regras. É importante ressaltar também que o limite de tempo (60s) mostrou-se justo, sendo atingido em 14% dos turnos, mas o limite de erros (5) nunca foi atingido. Considera-se então que esse limite deva ser abaixado em futuras aplicações. É importante ressaltar que o número relativamente alto de turnos passados voluntariamente dá-se ao fato de 3 das partidas estarem com jogadores a menos, cujas vezes eram repetidamente passadas.

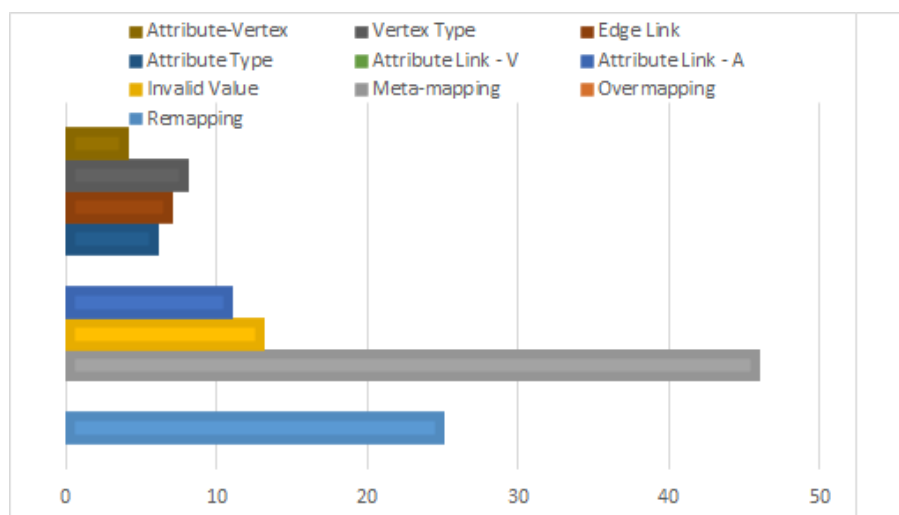


Figura 4. Erros de mapeamento cometidos durante a aplicação.

A Figura 4 traz a relação de todos os erros cometidos durante as 6 partidas, classificados conforme a Tabela 2. Particularmente, os erros de *meta-mapping* e *remapping* se mostraram muito comuns, mas pode-se atribuir parte de sua popularidade a razões não intencionais como cliques acidentais. Também é interessante a ausência de erros *Attribute Link-V*, uma vez que um erro muito similar, *Attribute Link-A* ocorreu expressivamente. A inoportunidade de *Overmappings* pode chamar a atenção da área formal, pois ele poderia nem ser considerado um erro (outras definições de GGs podem ter *matches* não injetores).

A Figura 5 traz a relação de tempos dos turnos de todos os jogadores. Seu eixo vertical representa o tempo em segundos e é duplamente escalado, sendo a escala da esquerda para o tempo médio, e a da direita para o tempo total. Seu eixo horizontal representa os jogadores, divididos em partidas. A versão digital tem 4 jogadores fixos em cada partida, portanto, as partidas P3, P5 e P6 apresentam uma discrepância nos tempos de seus últimos jogadores porque elas foram jogadas por menos de 4 jogadores reais. Pode-se perceber pelo gráfico perfis consideravelmente diferentes de jogadores, onde o mais rápido (média em torno de 25s) leva, em média, metade do tempo dos mais lentos.

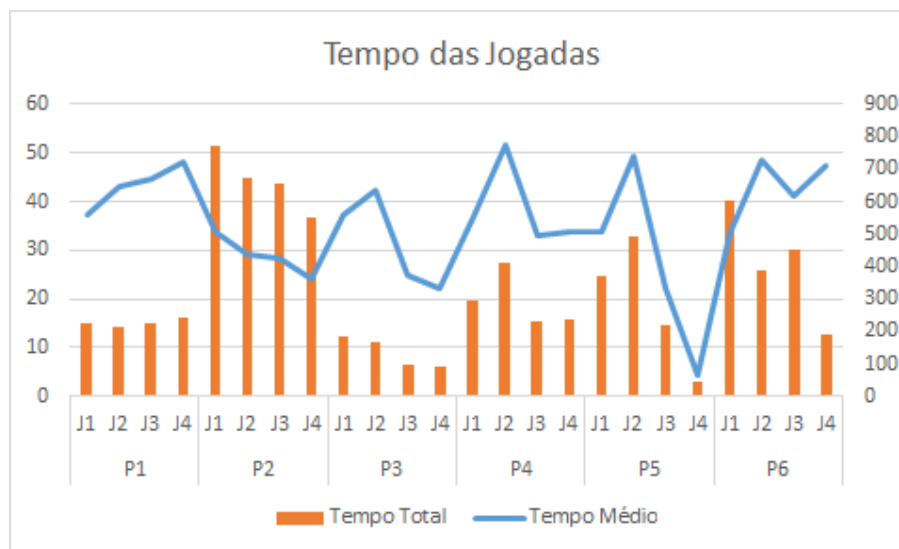


Figura 5. Tempos dos turnos durante a aplicação.

6. Avaliação do Jogo

Para avaliar a versão digital, a recepção e reação dos alunos ao software, utilizou-se questionários padrão do método para a avaliação de jogos educacionais para a computação, **MEEGA+** [Petri et al. 2016]. A Figura 6 no final do artigo apresenta os resultados dos questionários de avaliação. Como um dos alunos era portador de necessidades especiais, o responsável por ele, que o acompanhou e conduziu durante a atividade, também respondeu o questionário em seu lugar. O questionário revela que os alunos possuem um contato maior com jogos digitais e que, dentre as nuances do MEEGA+, as mais carentes foram percepção de aprendizagem e atenção focada, essa última, já esperada dado o contexto da aplicação (um evento com diversas outras atividades). Entende-se que essa outra carência deve-se ao caráter do que está sendo ensinado, habilidades cognitivas, portanto é difícil enxergar, até para os próprios alunos, o desenvolvimento delas (o que foi aprendido).

7. Conclusão

Este trabalho apresenta uma versão digital jogo educacional “A Última Árvore”, com o intuito de fornecer ferramentas para auxiliar uma melhor avaliação do desempenho dos alunos durante a atividade. As relações de GG e PC são revisitadas neste artigo, e a dinâmica das GGs é fortalecida na versão digital do jogo. São discriminados tipos de erros possíveis ao se jogar e apresentados gráficos exemplificando os dados coletados

automaticamente durante a aplicação e as respostas dos alunos nos questionários para a avaliação de qualidade do jogo. Os questionários mostram uma reação positiva no geral, mas um tanto carente nos aspectos de atenção focada, relevância e percepção de aprendizagem. Além da visão geral apresentada pelos gráficos neste artigo, ressalta-se que os logs e dados coletados são particularmente interessantes para avaliações individuais. Todas as versões do jogo citadas neste artigo estão disponíveis no site do projeto¹.

Referências

- Araujo, A. L. S. O., Andrade, W., Guerrero, D., Melo, M., and de Souza, I. M. L. (2018). Explorando teoria de resposta ao item na avaliação de pensamento computacional: um estudo em questões da competição bebras. In *SBIE*, volume 29, page 665.
- Avila, C. et al. (2017). Metodologias de avaliação do pensamento computacional: uma revisão sistemática. In *SBIE*, volume 28, page 113.
- Barr, V. and Stephenson, C. (2011). Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acad Inroads*, pages 48–54.
- CSTA, ISTE, and NSF (2009). *Computational Thinking Leadership Toolkit*. CSTA, Albany, NY, USA.
- Ehrig, H. et al. (1997). *Handbook of graph grammars and computing by graph transformation: volume I. Foundations*. World Scientific Publishing, Singapore.
- Guarda, G. and Goulart, I. (2018). Jogos lúdicos sob a ótica do pensamento computacional: Experiências do projeto logicamente. In *SBIE*, volume 29, page 486.
- Melo, D. et al. (2018). Robô euroi: Game de estratégia matemática para exercitar o pensamento computacional. In *SBIE*, volume 29, page 685.
- Ortiz, J. S. B. and Pereira, R. (2018). Um mapeamento sistemático sobre as iniciativas para promover o pensamento computacional. In *SBIE*, volume 29, page 1093.
- 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.
- Ribeiro, L. (2000). Métodos formais de especificação: gramáticas de grafos. *VIII Escola de Informática da SBC-Sul*, pages 1–33.
- Silva, E. and Javaroni, S. L. (2018). Pensamento computacional e atividades com robótica para a promoção da aprendizagem sobre o significado do resto da divisão euclidiana. In *SBIE*, volume 29, page 815.
- 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 *SBIE*, volume 28, pages 735–744.
- Silva Junior, B. A., Cavalheiro, S. A. C., and Foss, L. (2018). Uma análise de um jogo educacional sob a ótica do pensamento computacional. In *SBIE*, volume 29, page 595.
- Technologies, U. (2019). Unity. <https://unity.com/>. Acesso em: 07/07/2019.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

¹<https://wp.ufpel.edu.br/pensamentocomputacional/a-ultima-arvore/>

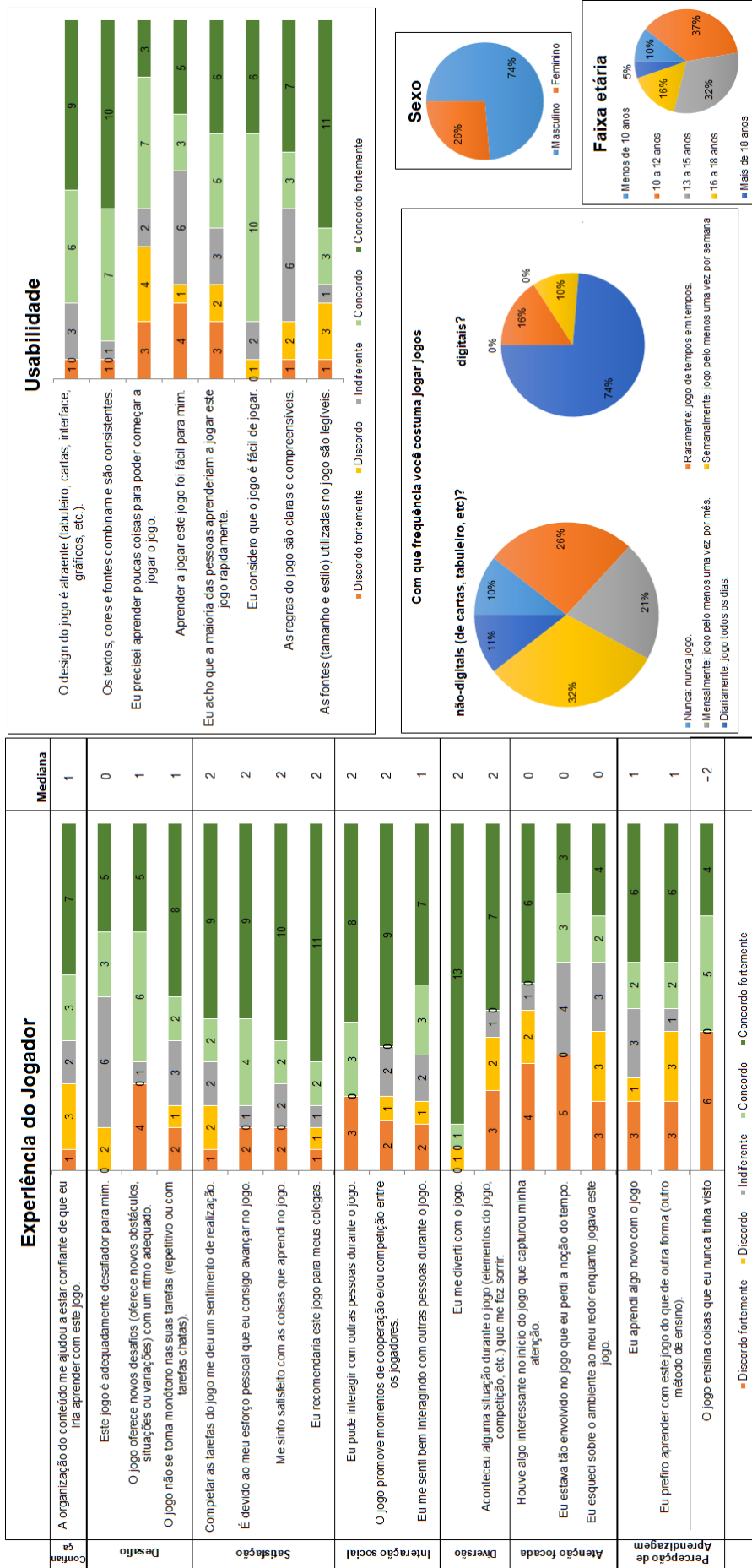


Figura 6. Resultados dos questionários de avaliação.