

Gramers: Agentes Pedagógicos para uma plataforma de jogos baseada em Gramática de Grafos^{*†}

Júlia Veiga da Silva¹, Braz Araujo da Silva Junior¹, Luciana Foss¹,
Simone André da Costa Cavalheiro¹

¹Laboratório de Fundamentos da Computação – Universidade Federal de Pelotas (UFPel)
CEP 96.010-610 – Pelotas – RS – Brasil

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

Abstract. *This paper proposes pedagogical agents to GameStation, a Graph Grammar-based game engine. In GameStation, creating a game corresponds to specifying a Graph Grammar. Although intuitive, the specification of a Graph Grammar may not be trivial to people who do not have previous knowledge. It makes the pedagogical agents for the platform necessary to help users during their experience. Therefore, four agents, called gramers, are proposed. Examples of the interaction of gramers in the platform are illustrated.*

Resumo. *Este artigo apresenta a proposta de agentes pedagógicos para o GameStation, um motor de jogos baseado em Gramática de Grafos. No GameStation, a criação de um jogo corresponde à especificação de uma gramática. Ainda que intuitiva, a especificação de uma Gramática de Grafos pode não ser trivial àqueles que não possuem conhecimento prévio. Isto torna necessária a presença de agentes pedagógicos, com o objetivo de auxiliar os usuários durante sua experiência. Neste sentido, quatro agentes, denominados gramers, são propostos. Exemplos da interação dos gramers na plataforma são ilustrados.*

1. Introdução

Em relação aos jogos digitais, ensinar a novos usuários as mecânicas e regras de um jogo pode ser desafiador. Assim, tutoriais são recursos frequentemente utilizados para auxiliar nesse processo. Para Gohl (2016), um tutorial divertido e fundamentado representa um importante recurso ao jogador, uma vez que problemas relacionados à mecânica do jogo, por exemplo, podem impedir seu progresso durante a partida, gerando frustração e, conseqüentemente, a perda de interesse.

Os jogos educacionais, além do entretenimento, visam o ensino de conteúdos específicos e o desenvolvimento de habilidades [Abt 1987]. Nesses jogos, a presença de personagens animados – os agentes – é comumente observada. De maneira geral, os agentes são responsáveis por apresentar tutoriais a respeito do funcionamento dos jogos, bem como guiar o usuário por fases/etapas, interagindo por meio de mensagens, alertas e elogios [Alencar e Netto 2017].

*Trabalho concluído.

†O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, do MCTIC/CNPq (Rede Sacci), da FAPERGS, do CNPq, da SMED/Pelotas e da PREC e PRPPG/UFPel.

Diversos trabalhos propõem agentes para a apresentação de tutoriais. Santana et al. (2017), por exemplo, apresentam um agente robô desenvolvido para ser vinculado a ferramentas de ensino on-line, cujo objetivo é mediar videoaulas acerca do desenvolvimento de jogos digitais. Já Helms et al. (2015) propõem agentes que visam contribuir com o aprendizado da Matemática no Ensino Médio, auxiliando na resolução de equações algébricas e reforçando conteúdos apresentados em sala de aula de maneira lúdica.

Neste contexto, o GameStation (GS)¹ é um motor de jogos baseado em Gramática de Grafos (GG), uma linguagem utilizada na Engenharia de Software para especificar sistemas e verificar propriedades. No GS, a especificação de um jogo corresponde à especificação de uma GG, com a definição de grafos e regras. Entretanto, ainda que a compreensão de uma GG seja intuitiva, sua especificação pode não ser trivial a quem não está habituado a esse formalismo, considerando-se, portanto, a necessidade de um tutorial para a utilização da plataforma. Assim, com o intuito de auxiliar a compreensão da manipulação/criação das GGs no GS, são propostos quatro agentes pedagógicos, denominados *gramers*, com a intenção de resolver esse problema.

O artigo está organizado como segue. A Seção 2 expõe a fundamentação teórica acerca de tutoriais e agentes pedagógicos, bem como a definição de GG e a apresentação do GS. A Seção 3 descreve a metodologia e proposta deste trabalho: agentes pedagógicos para o GS. A Seção 4 apresenta exemplos da interação dos agentes na plataforma. A Seção 5 discorre sobre as considerações finais e trabalhos futuros para esta proposta.

2. Fundamentação Teórica

Nesta seção são apresentados os principais conceitos abordados neste artigo.

2.1. Tutoriais

De acordo com White (2014), um tutorial é qualquer componente de um jogo digital destinado a ensinar alguém a jogar. Neste sentido, para Sweetser e Wyeth (2005), a importância de um tutorial aumenta de acordo com a complexidade de determinado jogo. Além disso, o aprendizado por meio desse recurso não deve ser entediante, mas fazer parte da diversão.

Visando a criação de tutoriais eficazes, Gohl (2016) estabelece um conjunto composto por 16 diretrizes para o desenvolvimento de tutoriais envolventes, entre elas: 1) Envolver os jogadores e chamar sua atenção no início do jogo; 2) Fornecer uma quantidade apropriada de dicas; 3) Ensinar antecipadamente as habilidades que devem ser utilizadas posteriormente ou logo antes de serem necessárias pela primeira vez; 4) Apresentar os objetivos prioritários com clareza e antecedência; 5) Fornecer desafios em um ritmo adequado para não frustrar os jogadores; 6) Proporcionar uma experiência divertida e positiva durante o aprendizado do jogo e domínio dos desafios; 7) Dar suporte na recuperação de erros e não penalizar os jogadores, repetidamente, pela mesma falha; 8) Fornecer feedbacks sobre o progresso no jogo; 9) Prover um feedback positivo imediato para a primeira ação do jogador.

2.2. Agentes Pedagógicos

O termo “agente” é utilizado para denotar desde simples processos de hardware/software, até entidades capazes de realizar tarefas complexas. Entre as propriedades básicas de um

¹Disponível em: <https://wp.ufpel.edu.br/pensamentocomputacional/gramestation-pt/>.

agente estão a autonomia, comunicação, mobilidade e flexibilidade. Já a noção de agentes pedagógicos passou a ser observada na literatura em virtude da utilização do paradigma de agentes em sistemas desenvolvidos para fins educacionais [Giraffa 1999].

Os agentes pedagógicos, portanto, atuam como tutores ou companheiros virtuais, auxiliando os estudantes no processo de aprendizagem [Giraffa 1999]. Giraffa (1999) divide os agentes pedagógicos em dois tipos: *goal-driven* (guiados por objetivos) e *utility-driven* (guiados por sua utilidade no ambiente). Agentes *goal-driven* são tutores que interagem com o usuário, realizando atividades em conjunto. Já agentes *utility-driven* realizam tarefas ligadas a atividades pedagógicas, como auxiliar o estudante a encontrar programas específicos, arquivos, diretórios, entre outros. Segundo Gratch et al. (2002), a presença de agentes em ambientes educacionais motiva os estudantes e, conseqüentemente, amplia o efeito da aprendizagem, dado que sua atenção é facilmente cativada por meio dos recursos visuais.

2.3. Gramática de Grafos

Uma GG é uma generalização das gramáticas de Chomsky, substituindo *strings* por grafos [Ehrig et al. 1973]. Em uma GG, os estados de um sistema são representados como grafos (definidos por vértices e arestas com origem e destino estabelecidos) e seus eventos (transições entre estados) por regras de transformação de grafos. Dessa forma, é possível realizar a descrição visual de sistemas complexos por meio de suas características e comportamentos.

Formalmente, uma GG é definida por um grafo tipo, cuja função é restringir os elementos (vértices e arestas) permitidos em um sistema; um grafo inicial (tipado sobre o grafo tipo) que especifica o estado inicial do sistema; e um conjunto de regras que definem os possíveis comportamentos do sistema. Sendo assim, a partir da especificação de uma GG, é possível simular esse sistema a fim de verificar estados alcançáveis e detectar e evitar estados indesejados.

2.4. GameStation: Motor de Jogos Baseado em GG

O GS é um motor de jogos baseado em GG desenvolvido na plataforma Unity, na linguagem de programação C#. A plataforma possibilita o desenvolvimento de habilidades associadas ao Pensamento Computacional (PC) que, por sua vez, é definido como um processo de resolução de problemas fundamentado na Ciência da Computação [Wing 2006]. Para Wing, o PC é um processo de propósito geral que não deve ser explorado somente por profissionais da computação, mas por todos. Silva Junior (2020) relaciona características das GGs a diversas habilidades do PC, como coleta, análise e representação de dados, decomposição de problemas, abstração, algoritmos e processos, simulação e paralelismo.

No GS, o grafo tipo corresponde a uma área de declaração, o grafo inicial refere-se à organização do jogo ao início da partida e as regras representam as possíveis ações a serem realizadas pelo jogador. A plataforma é estruturada em módulos, dividindo-se em: Game Tutorial (treinamento), Game Builder (criação) e Game Player (execução). No primeiro módulo, o usuário é conduzido na criação de um jogo – o Jogo da Velha – no formato de uma GG. Já no segundo, o usuário é capaz de criar seus próprios jogos, especificando as GGs. Por fim, no terceiro módulo, o usuário consegue executar seus jogos simulando as GGs, ou seja, selecionando regras e as aplicando, de maneira adequada, no grafo inicial.

A primeira etapa de especificação de uma GG no GS refere-se à importação de recursos externos (como imagens) a serem incluídos no jogo, seguida da criação dos grafos tipo e inicial e das regras. Na especificação dos elementos do grafo tipo, algumas propriedades devem ser definidas para cada um deles: para os vértices, deve-se definir nome, aparência, cor, posição (horizontal e vertical) e tamanho; já para as arestas são definidos nome, aparência, cor, vértice-origem, vértice-destino e tamanho. Os elementos dos demais grafos herdam essas propriedades ao terem seus tipos definidos, com exceção do nome (para vértices e arestas) e da posição (apenas para vértices), também definidos nos outros grafos. Na plataforma, ainda, existem formulários que registram os valores definidos para cada uma dessas propriedades, para cada grafo da gramática. A Figura 1 ilustra, à esquerda, o grafo tipo do Jogo da Velha, responsável por definir todos os elementos permitidos na especificação adequada do jogo, definindo seus tipos e possíveis relações. Os vértices \times e \circ representam as marcações dos dois jogadores da partida, enquanto o \square corresponde às subdivisões do tabuleiro – denominadas “casas” – nas quais os jogadores colocam suas respectivas marcações. O \triangle , por sua vez, representa o vazio, utilizado para indicar uma casa que não está marcada por um vértice. As arestas amarela, vermelha e cinza indicam, respectivamente, a relação dos vértices \times , \circ e \triangle com uma determinada casa. Por fim, as demais arestas sinalizam a relação entre casas nas direções estabelecidas pelo jogo: horizontal (azul), vertical (laranja), diagonal principal (rosa) e diagonal secundária (verde). O grafo inicial, ao centro, indica a organização dos elementos do jogo no início da partida. Visto que o tabuleiro do Jogo da Velha deve iniciar com nove casas desocupadas, é atribuída, a cada casa, a relação com o elemento vazio. As relações entre as casas nas direções estabelecidas também são indicadas.

As regras “Marcar \times ” (à direita, acima) e “Marcar \circ ” (à direita, abaixo), por sua vez, especificam as possíveis ações dos jogadores em seus respectivos turnos. Para a aplicação dessas regras, uma condição deve ser atendida: existir uma casa desocupada no tabuleiro. Após a aplicação, um efeito é produzido: o vértice \times ou o vértice \circ é marcado nessa casa e o vértice \triangle é eliminado. Além dessas, são definidas as regras responsáveis por indicar o momento em que um dos jogadores vence a partida. Para isso, deve-se obter a sequência de três elementos iguais em uma das quatro direções permitidas. Sendo assim, são especificadas 8 regras: duas possibilidades de marcação (\times e \circ) para quatro possibilidades de direção (horizontal, vertical, diagonal principal e diagonal secundária).

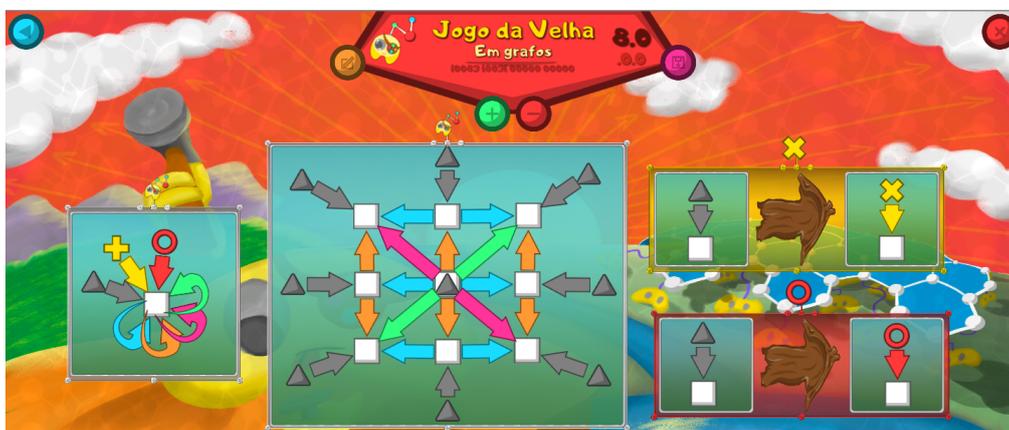


Figura 1. Grafo tipo (à esquerda), grafo inicial (ao centro) e regras (à direita)

A Figura 2 ilustra as regras que definem a vitória do jogador com marcação \times na horizontal (na parte superior da figura) e do jogador com marcação \circ na vertical (na parte inferior da figura). Após a criação do jogo, é possível executá-lo e jogá-lo selecionando as regras especificadas e as aplicando no grafo inicial (Figura 3).



Figura 2. Regras “Vitória \times Horizontal” (acima) e “Vitória \circ Vertical” (abaixo)

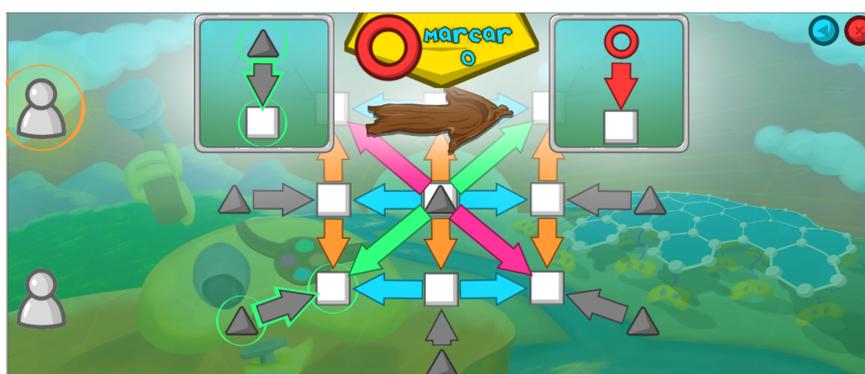


Figura 3. Aplicação da regra “Marcar \circ ” durante a execução do jogo

3. Agentes Pedagógicos no GS: Metodologia e Proposta

A primeira etapa deste trabalho consistiu na familiarização com o formalismo de GG e com a ferramenta. A partir das primeiras experiências, por meio da criação de jogos, foram identificadas dificuldades que, posteriormente, guiaram o processo de proposta deste trabalho, como: (1) dúvidas em relação aos termos e ações da plataforma, além dos termos e conceitos relacionados à GG – como grafo, tipo e morfismo; (2) dificuldade na definição das relações entre os elementos (estabelecidas por arestas em uma GG), as quais independem de posição ou coordenada; e (3) falta de percepção quanto às restrições impostas pelo grafo tipo, o qual exige que os demais grafos do sistema possuam, apenas, elementos e relações previamente declarados. Como resultado dessas experiências, são propostos quatro agentes pedagógicos *goal-driven* denominados gramers, que auxiliam o jogador em diferentes contextos ao longo da experiência na plataforma.

Gruly (Figura 4), a primeira gramer do GS, é proposta para contornar as dificuldades elencadas de maneira geral. Com a aparência de um *headset* de realidade virtual, apresenta o “mundo dos grafos” ao usuário, o orientando por meio de tutoriais – da execução

de ações na plataforma, à criação/manipulação de um jogo como GG. Uma abordagem frequentemente utilizada em jogos é a apresentação do tutorial em uma área de treinamento específica, na qual a mecânica do jogo é ensinada [Sweetser e Wyeth 2005]. Gruly, portanto, está localizada em uma área específica do GS, destinada à execução de tutoriais. No tutorial proposto, o usuário é guiado por Gruly a fim de realizar a criação passo a passo do Jogo da Velha – o que permite a visualização das estruturas que compõem as GGs à medida que o jogo é especificado. Para atender a problemática (1), Gruly está presente desde a tela inicial – informando sobre as funcionalidades disponíveis no GS (como criar, editar e executar um jogo) –, até a área de tutoriais, onde são apresentados os principais conceitos de GG. Já as dificuldades apontadas em (2) e (3), são facilmente contornadas por meio da criação gradual do jogo com o auxílio das explicações da gramer a cada nova etapa.

A escolha do Jogo da Velha para o tutorial considerou, além da simplicidade e popularidade, o fato do jogo apresentar-se de maneira distinta da convencional em uma GG – tanto pelo modo como os elementos são localizados, quanto em relação à forma que a matriz do jogo é “construída”. Além disso, o jogo oferece a possibilidade de destacar pontos, durante a implementação, que não são óbvios àqueles que não estão habituados a esse formalismo, principalmente em relação à definição da relação entre elementos.

Conforme apontado na subseção 2.3, as GGs – tal como o GS – contêm diversos termos próprios. Logo, para habituar o usuário a essas expressões – e complementar as problemáticas (1) e (2) – propõe-se o gramer Graz (Figura 4). Com a aparência de um *joystick* de console de videogame, carrega um glossário de A a Z com palavras referentes às GGs e ao GS, bem como suas definições formais e informais, possibilitando ao usuário consultar/estudar os termos durante sua experiência. De certo modo, o jogador é introduzido a uma nova linguagem e, portanto, deve ser apresentado ao vocabulário da mesma.

Dra. Grafoss (Figura 4), gramer com a aparência do console portátil Game Boy, utiliza os elementos de um determinado grafo como fita e, quando solicitada, carrega suas informações no formulário, apresentando-as ao jogador. Esse recurso auxilia a edição de elementos pois facilita o acesso a dados específicos. Assim, o usuário é capaz de saber, exatamente, o atributo que deseja alterar.

De acordo com o modelo GameFlow, proposto por Sweetser e Wyeth (2005), entre os 8 elementos fundamentais para uma experiência divertida em jogos está o feedback. Segundo os autores, os jogadores devem receber feedback imediato sobre suas ações, além de sempre estarem cientes sobre seu status ou pontuação. Neste sentido, propõe-se a gramer Grimone (Figura 4). Também com a aparência de um *joystick* de console de videogame, sua função é exibir mensagens de acertos e erros na aplicação de regras (*matches*) e *warnings* (mensagens de aviso) durante a execução do jogo.

4. Exemplos de Interação dos Gramers no GS

O módulo Game Tutorial é destinado a usuários não habituados com o formalismo de GG. Dessa maneira, a gramer Gruly conduz o usuário para a criação de todos os grafos necessários para a execução adequada do Jogo da Velha, comunicando-se por meio de caixas de diálogo. Para a especificação do grafo tipo, por exemplo, Gruly descreve todos os elementos que o compõem, bem como o papel de cada um para o funcionamento do



Figura 4. Gamers, da esquerda para a direita: Gruly, Graz, Dra. Grafoss (em cima) e Grimone (embaixo)

jogo. As explicações são apresentadas à medida que tarefas – como arrastar elementos para áreas indicadas na tela – são solicitadas, de modo que, ao final do tutorial, o usuário tenha criado completamente todos os grafos do jogo.

O módulo Game Builder é destinado à criação de jogos. Neste ambiente, o gramer Graz está localizado na parte superior da tela. Sua função é exibir ao jogador, por meio de caixas de diálogo, as definições formais e informais a respeito de elementos já especificados. Ao selecionar, por exemplo, o ícone relativo ao grafo tipo, Graz apresenta as seguintes informações: informalmente, o grafo tipo define os diferentes tipos de vértices e arestas que podem aparecer em uma especificação. Todos os demais grafos de uma GG são compostos por instâncias de vértices e arestas do grafo tipo; formalmente, um grafo G é uma tupla $G = (V_G, E_G, src_G, tgt_G)$, onde V_G é um conjunto de vértices, E_G , um conjunto de arestas e $src_G, tgt_G : E_G \rightarrow V_G$ são funções totais que definem, respectivamente, o vértice de origem e destino de cada aresta.

Ainda no Game Builder, a gramer Dra. Grafoss também encontra-se na parte superior da tela. Sua função é exibir as informações de um determinado elemento pertencente a um grafo. Ao selecionar, por exemplo, um dos vértices \square do grafo inicial (Figura 1, ao centro), Dra. Grafoss o utiliza como fita – já que possui a aparência de um Game Boy – e exibe um formulário com todas as informações do elemento. Neste caso nome, aparência, cor, posição horizontal, posição vertical e tamanho.

Por fim, no módulo Game Player, o jogador consegue reproduzir o jogo criado no módulo anterior. Neste módulo, a gramer Grimone é a responsável por interagir com o usuário. Ao selecionar a regra “Marcar \circ ” (Figura 3), por exemplo, e aplicá-la corretamente, a gramer é exibida na parte inferior da tela junto a uma mensagem positiva ao jogador. No entanto, ao tentar aplicar a próxima regra – “Marcar \times ”, por exemplo – no mesmo vértice, Grimone é exibida junto a uma mensagem de erro, uma vez que a condição para a aplicação dessa regra é a existência de uma casa desocupada no tabuleiro.

5. Considerações Finais

Este trabalho apresenta a proposta da utilização de agentes pedagógicos no GameStation, motor de jogos baseado em Gramática de Grafos. Embora visual, a especificação de uma Gramática de Grafos pode não ser trivial. Dessa forma, espera-se que a presença de

agentes tutores contribua tanto com o processo de compreensão desse formalismo, quanto com a criação/manipulação dessas gramáticas na plataforma.

O próximo passo deste trabalho refere-se à finalização da implementação dos agentes na plataforma. Após, a fase de testes será iniciada. Serão realizados testes de usabilidade, com foco na interface de software e na experiência do usuário. A primeira avaliação será realizada com estudantes de cursos de computação, divididos em dois grupos: o primeiro utilizará a plataforma com a presença dos gamers e o segundo utilizará a plataforma sem a presença dos gamers. Posteriormente, deseja-se realizar os mesmos testes com professores da Educação Básica e, mais a frente, com os estudantes. A partir dos testes espera-se analisar, com maior aprofundamento, os aspectos técnicos relevantes em relação ao desenvolvimento dos agentes e avaliar sua utilização a fim de evidenciar sua adequação ao propósito inicial.

Referências

- Abt, C. C. (1987). *Serious Games*. University Press of America.
- Alencar, M. A. S. e Netto, J. F. (2017). Melhorando a Colaboração em um Ambiente Virtual de Aprendizagem usando um Agente Pedagógico Animado 3D. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 28(1):1417–1426.
- Ehrig, H., Pfender, M., e Schneider, H. J. (1973). Graph-grammars: An algebraic approach. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 167–180.
- Giraffa, L. M. M. (1999). *Uma arquitetura de tutor utilizando estados mentais*. PhD thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil.
- Gohl, L. (2016). Usability guidelines for the creation and evaluation of functional and engaging tutorials in computer games. Trabalho de Conclusão de Curso, Universidade de Ciências Aplicadas de Hamburgo, Hamburgo, Alemanha.
- Gratch, J., Rickel, J., Andre, E., Cassell, J., Petajan, E., e Badler, N. (2002). Creating Interactive Virtual Humans: Some Assembly Required. *IEEE Intelligent Systems*, 17(4):54–63.
- Helms, F., Loreto, A., Adamatti, D., Buss, C., e Ferreira, A. (2015). Modelando um sistema tutor multiagentes para auxiliar na aprendizagem da matemática. *Scientia Plena*, 11(8).
- Santana, A., Silva, T. R., e Aranha, E. (2017). Um Modelo de Tutor Virtual para Aulas Baseadas em Missões de Programação de Jogos Digitais. *Workshops do Congresso Brasileiro de Informática da Educação - CBIE*, 6(1):1059–1068.
- Silva Junior, B. A. (2020). GGasCT: Bringing Formal Methods to the Computational Thinking. Master's thesis, Universidade Federal de Pelotas, Pelotas, RS, Brasil.
- Sweetser, P. e Wyeth, P. (2005). GameFlow: A Model for Evaluating Player Enjoyment in Games. *ACM Computers in Entertainment*, 3(3):3–3.
- White, M. M. (2014). *Learn to Play: Designing Tutorials for Video Games*. CRC Press.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.