

Adaptação do processo ENgAGED para o desenvolvimento de conteúdos curriculares em 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. *Educational games stand out because they contribute to develop skills in a ludic way. However many of them do not have a theoretical foundation regarding the creation, validation and evaluation processes. GameStation is a graph grammar-based game creation platform that can be used in Basic Education to foster Computational Thinking. This paper proposes an adaptation of a method for developing educational games (ENgAGED) using the GameStation platform. An example of instantiation is given to illustrate the proposed method.*

Resumo. *Os jogos educacionais destacam-se por contribuir para o desenvolvimento de habilidades de maneira lúdica. Entretanto, muitos deles não apresentam uma fundamentação teórica em relação aos processos de criação, validação e avaliação. O GameStation é um motor de jogos baseado em Gramática de Grafos que pode ser utilizado na Educação Básica para desenvolver o Pensamento Computacional. Neste artigo, propõe-se uma adaptação de um método de desenvolvimento de jogos educacionais (ENgAGED) utilizando a plataforma GameStation. Um exemplo de instanciação é apresentado a fim de ilustrar o método proposto.*

1. Introdução

Os jogos educacionais são propostas idealizadas para além do entretenimento, visando o ensino de conteúdos específicos e o desenvolvimento de habilidades [Abt 1987]. Representam um importante recurso em projetos de ensino, pois além proporcionar entretenimento, motivam e auxiliam os estudantes no desenvolvimento de habilidades e estratégias, possuindo um elevado potencial educativo [Gros 2003]. Ainda, tanto os jogos educacionais, quanto os tradicionais, são capazes de ampliar as possibilidades de assimilação e o aprendizado de novos conhecimentos, uma vez que proporcionam experiências que contribuem para o aprimoramento de habilidades cognitivas e processos emocionais [Ramos et al. 2016].

Atualmente, ainda que o desenvolvimento de jogos educativos seja crescente, poucos contam com o suporte de técnicas bem consolidadas. Em uma revisão sistemática da

*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.

literatura realizada por [Battistella 2016], acerca do desenvolvimento de jogos para o ensino de computação na Educação Superior, entre os 107 jogos encontrados, a maioria, segundo o autor, não possuía descrições quanto ao seu desenvolvimento ou avaliações, além de não apresentar teorias acerca do design instrucional e design de jogos utilizados. Também foram encontrados 47 artigos que apresentavam descrições quanto ao processo utilizado para o desenvolvimento dos jogos. Entre os 47, 12 foram considerados relevantes e apenas 5 descreviam claramente as fases do processo de criação.

O autor também pesquisou por descrições quanto ao alinhamento dos jogos às teorias instrucionais – relacionadas ao contexto educacional, público-alvo e estratégias de ensino –, porém, na maioria dos casos, não foram encontradas. Quanto aos métodos de avaliação, entre os 107 jogos, 55 não continham informações a respeito e 43 apresentavam uma abordagem não-experimental como método avaliativo – i.e., a aplicação do jogo em sala de aula, seguida de uma avaliação destinada aos alunos. Neste contexto, [Battistella 2016] propôs o ENgAGED: um processo de desenvolvimento de jogos educacionais que compara processos existentes e integra elementos do design.

A gamificação tem se popularizado como estratégia educacional e conta com recursos como, por exemplo, o Educaplay¹, uma plataforma educativa on-line que possibilita a criação de atividades, no formato de jogos, por professores. A plataforma disponibiliza tabuleiros e temas pré-definidos, separados de acordo com o nível de ensino, configurações básicas e avançadas para a concepção de tarefas e a possibilidade de exportação de atividades para plataformas de apoio à aprendizagem, como o Moodle². Já em relação a ambientes de desenvolvimento, o Scratch [Resnick et al. 2009] é uma linguagem de programação visual utilizada para o ensino de diversos conteúdos além de programação. O ambiente permite a criação de conteúdos como jogos, histórias interativas e animações pelos alunos.

Neste contexto, o GameStation (GS)³ é um motor de jogos baseado em Gramática de Grafos (GG), uma linguagem utilizada na Engenharia de Software para especificar sistemas e verificar propriedades. Comparado às propostas de ambientes educativos apresentadas, além de proporcionar o desenvolvimento de conteúdos curriculares de maneira lúdica, o GS possibilita o desenvolvimento de habilidades relacionadas ao Pensamento Computacional (PC), definido como um processo de resolução de problemas fundamentado na Ciência da Computação [Wing 2006]. Para a autora, o PC é um processo de propósito geral que não deve ser explorado somente por profissionais da computação, mas por todos. Em trabalhos recentes, observa-se a aplicação de diferentes práticas aliadas ao PC na Educação Básica, como o PC Desplugado [Souza e Nunes 2019], Programação em Blocos [Sousa et al. 2020] e Robótica Educacional [Avila et al. 2017]. 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. Sendo assim, para que o processo de concepção de jogos seja embasado em técnicas consolidadas, propõe-se, neste trabalho, uma adaptação do processo ENgAGED para o desenvolvimento de jogos no GS. No que se refere a trabalhos relacionados, não foram encontradas outras propostas que estendem/adaptam o

¹Disponível em: <https://www.educaplay.com/>.

²Disponível em: <https://moodle.org/>.

³Disponível em: <https://wp.ufpel.edu.br/pensamentocomputacional/gramestation-pt/>.

método ENgAGED.

O artigo está organizado como segue. A Seção 2 define GG. A Seção 3 apresenta o GS, plataforma baseada em GG destinada à criação de jogos. A Seção 4 expõe brevemente o processo ENgAGED e suas fases. A Seção 5 descreve o processo proposto para este artigo, baseado no modelo ENgAGED. A Seção 6 apresenta um exemplo de instanciação seguindo as etapas de criação propostas. A Seção 7 discorre sobre as considerações finais e trabalhos futuros.

2. Gramática de Grafos

Uma GG é uma generalização das gramáticas de Chomsky, substituindo *strings* por grafos [Ehrig et al. 1973]. Trata-se de uma linguagem visual que permite a representação de determinado problema e/ou sua solução com um elevado grau de abstração de dados e processos. Os estados são representados como grafos (definidos por vértices e arestas com origem e destino estabelecidos) e os 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 [Albane et al. 2018, Gutierrez e Schrum 2020, Albane et al. 2020].

Formalmente, uma GG é definida por um grafo tipo, cuja função é restringir os elementos (vértices e arestas) permitidos em um determinado sistema; um grafo inicial (composto por elementos de tipos definidos pelo grafo tipo) que especifica o estado inicial do sistema; e um conjunto de regras que definem os possíveis comportamentos do sistema. Uma regra é composta por dois grafos: o lado esquerdo (**Left Hand Side - LHS**) e o lado direito (**Right Hand Side - RHS**). O primeiro descreve a condição para a aplicação da regra e o segundo, seu efeito. Para que uma regra seja aplicada, deve-se encontrar uma ocorrência (*match*) do seu LHS no grafo que descreve o estado. Uma vez que a ocorrência é encontrada, a regra é aplicada, substituindo o subgrafo imagem do *match* pelo RHS da regra. Dessa forma, é possível realizar a simulação de um sistema a partir da especificação de uma GG, verificando estados alcançáveis e detectando e evitando estados indesejados.

3. 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 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 do grafo tipo, que desempenha na plataforma a função de uma área de declaração. A etapa seguinte corresponde à criação do grafo inicial, que exhibe a organização do jogo ao início da partida e, por último, a etapa de criação de regras, que definem as ações do jogo.

A Figura 1 ilustra parte da especificação do Jogo da Velha no GS. No grafo tipo (à esquerda) são definidos todos os elementos referentes ao jogo e suas 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 quando determinada casa não está marcada por um vértice. Já as arestas amarela, vermelha e cinza indicam, respectivamente, as relações dos vértices \times , \circ e \triangle com uma determinada casa. Por fim, as demais arestas representam a relação entre

casas nas direções estabelecidas pelo jogo: horizontal (azul), vertical (laranja), diagonal principal (rosa) e diagonal secundária (verde). O grafo inicial – ou grafo estado – (à direita) indica a disposição dos elementos do jogo no início da partida. Visto que o Jogo da Velha deve iniciar com um tabuleiro composto por 9 casas desocupadas, é atribuído, a cada uma, a relação com o elemento vazio, bem como as relações entre as casas nas direções estabelecidas.

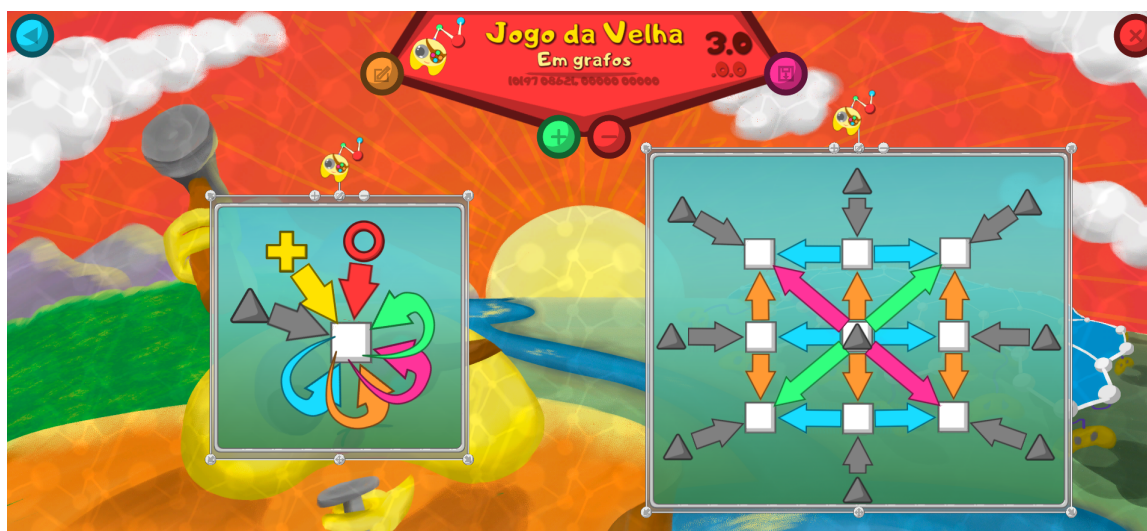


Figura 1. Grafos tipo (esquerda) e inicial (direita)

Os estados do jogo podem ser alterados por meio da aplicação de regras. Na Figura 2, as regras “Marcar \times ” (à esquerda) e “Marcar \circ ” (à direita) especificam as ações possíveis aos jogadores em seus respectivos turnos. Para a aplicação de qualquer uma dessas regras, uma condição deve ser atendida (LHS): a presença de uma casa desocupada no tabuleiro. Após a aplicação, um efeito é produzido (RHS): o vértice \times ou o vértice \circ é marcado nesta casa e o vértice \triangle é deletado.



Figura 2. Regras “Marcar \times ” (esquerda) e “Marcar \circ ” (direita)

No GS, o PC é desenvolvido tanto por quem cria um jogo (especifica uma GG), quanto por quem o executa (simula uma GG). Em [Silva Junior et al. 2019] é detalhado como os conceitos de representação de dados, decomposição de problemas, abstração, algoritmos e processos, paralelismo, entre outros, são desenvolvidos por meio do design ou da simulação de um jogo (GG).

4. O Processo ENgAGED

O ENgAGED (EducatioNAI GamEs Development) é um processo de desenvolvimento de jogos educacionais para o ensino de computação, que considera aspectos de design instrucional e de jogos [Battistella 2016]. O design instrucional é entendido como um sistema organizado e estruturado, concentrado na execução de tarefas, com o objetivo de promover a aprendizagem [Branch 2009]. Já o design de jogos, refere-se às decisões acerca dos elementos de um jogo, considerando aspectos como regras, comportamento, aparência e recompensas [Schell 2014]. O processo é modelado em 5 fases relativas ao design instrucional, com a Fase 3, Desenvolvimento do Jogo Educacional, sendo composta por outras 5 fases relativas ao design de jogos (Figura 3).

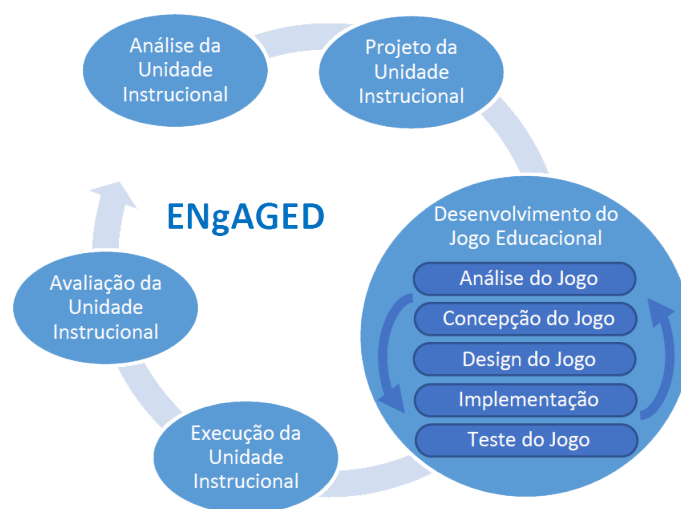


Figura 3. Fases do processo ENgAGED, adaptado de [Battistella 2016]

No processo original, a Fase 1, **Análise da Unidade Instrucional (UI)**, refere-se à definição do escopo da UI, do público-alvo e dos objetivos de aprendizagem. A Fase 2, **Projeto da Unidade Instrucional (UI)**, refere-se à definição de estratégias instrucionais para alcançar os objetivos de aprendizagem propostos na Fase 1, bem como o desenvolvimento de instrumentos avaliativos. A Fase 3, **Desenvolvimento do Jogo Educacional**, refere-se ao desenvolvimento do jogo e é composta por fases e atividades que orientam o professor no processo de implementação. Entre as 5 subfases estão: **Análise do Jogo**, com o objetivo de realizar o levantamento de requisitos do jogo; **Concepção do Jogo**, com o objetivo de definir características do jogo, como gênero, plataforma, regras etc.; **Design do Jogo**, com o objetivo de definir as linguagens de programação, *game engine* e componentes (bibliotecas) a serem utilizados; **Implementação do Jogo**, com o objetivo de codificar os elementos produzidos na fase anterior; e **Teste do Jogo**, com o objetivo de identificar erros durante a execução do jogo. A Fase 4, **Execução da Unidade Instrucional (UI)**, refere-se ao planejamento e execução da UI. Por fim, a Fase 5, **Execução da Unidade Instrucional (UI)**, refere-se à definição, planejamento e execução da avaliação da UI.

Dessa forma, considerando que a criação de um jogo modelado como uma GG possui tarefas bem definidas, sendo um processo consideravelmente diferente da criação

em outros modelos, uma série de adaptações do processo ENgAGED é proposta, alvejando a criação de jogos educacionais por meio da plataforma GS.

5. O Processo Adaptado

O processo proposto adapta e considera determinadas etapas da metodologia original, bem como inclui fases novas específicas para a plataforma. As adaptações propostas foram baseadas no processo de definição de uma GG no GS. Uma vez que o processo é voltado para uma plataforma específica, algumas etapas originais do ENgAGED ocorrem de maneira automática, como, por exemplo, a fase Design do Jogo – por se tratar de uma *game engine*, especificações acerca da linguagem de programação, bibliotecas e demais configurações utilizadas ocorrem de maneira implícita no GS.

A proposta adaptada é modelada pelas seguintes fases: Fase 1. Análise da Unidade Instrucional (UI); Fase 2. Concepção de Mecânicas de Jogo; Fase 3. Desenvolvimento do Jogo Educacional; Fase 3.1. Definição do Jogo Educacional; Fase 3.2. Verificação de Novos Objetos de Conhecimento; Fase 3.3. Implementação do Jogo; Fase 3.3.1. Importação de Recursos; Fase 3.3.2. Criação do Grafo Tipo; Fase 3.3.3. Criação do Grafo Inicial; Fase 3.3.4. Criação de Regras; Fase 3.4. Teste do Jogo; Fase 4. Execução da Unidade Instrucional (UI); e Fase 5. Avaliação da Unidade Instrucional (UI).

No fluxograma (Figura 4), as fases demarcadas em azul correspondem às fases mantidas originais (1, 3.4, 4 e 5), as etapas em amarelo correspondem às fases adaptadas (2, 3, 3.3, 3.3.1) e as etapas em rosa correspondem às fases incluídas (3.1, 3.2, 3.3.2, 3.3.3 e 3.3.4). As linhas tracejadas indicam não ser necessário seguir a ordem sugerida estritamente. Por se tratar de um processo iterativo, as etapas podem repetir-se ao longo do desenvolvimento.

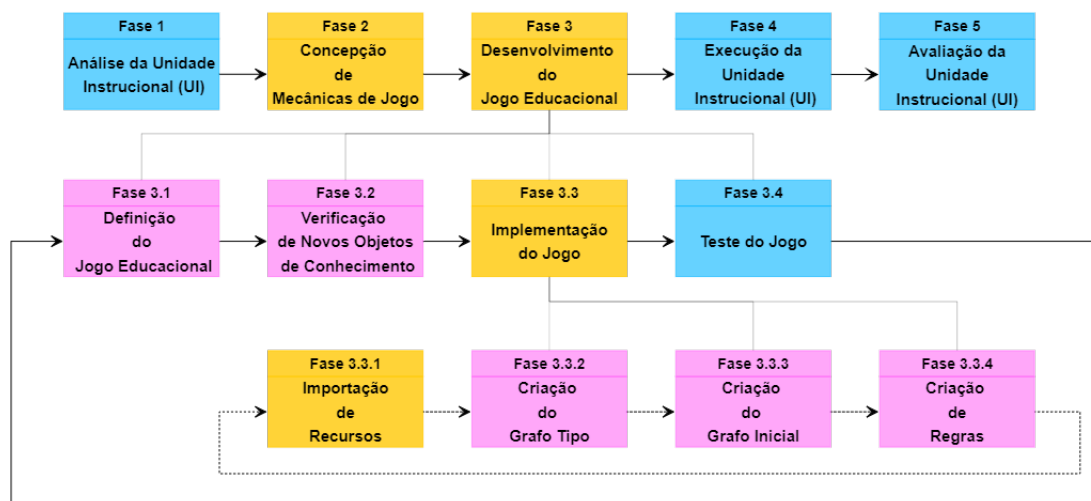


Figura 4. Fluxograma das etapas do processo adaptado

Nesta proposta, a Fase 1, **Análise da Unidade Instrucional (UI)**, é mantida na forma original do processo ENgAGED.

Substituindo a Fase 2 da metodologia original (Projeto da Unidade Instrucional (UI)), a fase **Concepção de Mecânicas de Jogo** refere-se ao momento em que o professor

analisa mecânicas capazes de desenvolver a habilidade inicialmente definida como objeto de conhecimento. Mecânicas são definidas como ações a serem realizadas pelo jogador durante o jogo [Brathwaite e Schreiber 2008] e, em uma GG, são análogas às regras.

A Fase 3, **Desenvolvimento do Jogo Educacional**, refere-se à criação da gramática que define o jogo e está dividida em 4 subfases: as fases 3.1 (Análise do Jogo) e 3.2 (Concepção do Jogo), do modelo original, foram identificadas em uma única Fase 3.1, **Definição do Jogo Educacional**. Refere-se à definição da ideia principal do jogo, considerando a contemplação das mecânicas concebidas na fase anterior. Nesta fase, ainda, o professor tem a possibilidade de optar por um jogo pré-existente e adaptá-lo ou conceber um jogo original — conforme a primeira opção da Fase 3.3, original do ENgAGED. A Fase 3.3 foi excluída: por se tratar de uma *game engine*, o GS não necessita de definições acerca da linguagem de programação utilizada ou da criação manual de elementos do jogo. Devido às especificidades do GS, é proposta a inclusão da Fase 3.2, **Verificação de Novos Objetos de Conhecimento**, que se refere à revisão do jogo com o objetivo de encontrar novas habilidades a serem exploradas e que não eram o foco inicialmente. A Fase 3.3 (originalmente 3.4), **Implementação do Jogo**, refere-se à concepção de elementos do jogo. Sendo assim, a Fase 3.3.1, **Importação de Recursos** é uma adaptação da Fase 3.4 original e refere-se à seleção de recursos externos a serem incluídos no jogo. A Fase 3.3.2, **Criação do Grafo Tipo**, refere-se à definição dos tipos, ou seja, a especificação de todos os elementos que fazem parte do jogo. A Fase 3.3.3, **Criação do Grafo Inicial**, refere-se à definição do estado inicial do jogo, ou seja, a organização do jogo ao início da partida. A Fase 3.3.4, **Criação de Regras**, refere-se à definição de eventos do jogo, ou seja, as ações dos jogadores durante a partida. A Fase 3.4 (originalmente 3.5), **Teste do Jogo**, manteve-se com a definição original, referindo-se à verificação do funcionamento do jogo a fim de encontrar possíveis erros. Por configurar um processo iterativo, é possível retornar à etapa inicial e percorrer novamente as fases, realizando os ajustes necessários em função de possíveis novos objetos de conhecimento identificados.

A Fase 4, **Execução da Unidade Instrucional (UI)**, e a Fase 5, **Avaliação da Unidade Instrucional (UI)**, independem da utilização do GS, mantendo-se na forma original.

6. Exemplo de Instanciação: Jogo da Velha

Nesta seção é apresentada uma instanciação do processo proposto. A Figura 7 sintetiza as etapas instanciadas. Para a definição da UI na Fase 1, foram utilizados objetos de conhecimento da Base Nacional Comum Curricular (BNCC). Na BNCC, objetos de conhecimento referem-se a conteúdos, conceitos e processos organizados em unidades temáticas [Brasil 2017]. Tratando-se de um documento de referência nacional para a elaboração de currículos escolares e propostas pedagógicas para a rede pública e privada de ensino, a BNCC configura uma base adequada para a busca por sugestões de objetos de conhecimento. Baseado nisso, as seguintes habilidades foram definidas: EF01MA11 e EF01MA12 – sugeridas para o 1º ano do Ensino Fundamental –, pertencentes à unidade temática Geometria, da área de conhecimento Matemática. A primeira habilidade exercita a descrição da localização de objetos no espaço em relação à sua própria posição, utilizando termos como à direita, à esquerda, em frente, atrás. A segunda, por sua vez, exercita a descrição de objetos no espaço segundo um dado ponto de referência, compreendendo que, para a utilização de termos que se referem à posição, como direita, esquerda, em

cima e embaixo, é necessário explicitar-se o referencial.

Dessas habilidades, foi concebida a mecânica para o jogo na Fase 2: a manipulação de elementos alinhados, formando linhas na horizontal, vertical e diagonal. Na etapa seguinte, Fase 3.1, percebeu-se a existência de um jogo que apresenta a mecânica concebida: o Jogo da Velha. Na Fase 3.2 foram encontradas novas habilidades capazes de serem desenvolvidas por meio do jogo: EF01MA09 (sugerida para o 1º ano do Ensino Fundamental) e EM13MAT311 (sugerida para o Ensino Médio), também na Matemática. A primeira, da unidade temática Álgebra, exercita a organização e ordenação de objetos familiares por meio de atributos como cor, forma e medida. Já a segunda, pertencente à Competência Específica 3 do Ensino Médio, exercita a identificação e descrição do espaço amostral de eventos aleatórios, realizando a contagem das possibilidades, para resolver e elaborar problemas que envolvam cálculos de probabilidades. O Jogo da Velha, portanto, além de contribuir para o desenvolvimento do raciocínio lógico e de estratégias, trabalha conceitos matemáticos. Com o jogo, é possível explorar conceitos como a direção de linhas (horizontal, vertical e diagonal) a partir da ordenação de objetos semelhantes por meio de referenciais e o número de possibilidades de vitória para cada peça, em uma determinada posição. Além disso, por possuir poucas regras e jogabilidade simples, é possível aplicá-lo em diferentes níveis da Educação Básica.

Posteriormente, na Fase 3.3, foram definidos o grafo tipo, o grafo inicial e as regras. Na Fase 3.3.1 foram importadas imagens externas para o GS, a fim de representar, de maneira coerente, os elementos do Jogo da Velha. Para a criação do grafo tipo (Fase 3.3.1) foram considerados todos os elementos que fazem parte do jogo (\times , \circ , \square e \triangle), as arestas que estabelecem as relações entre as casas do tabuleiro (horizontal, vertical, diagonal primária e diagonal secundária) e as arestas que estabelecem as relações entre os elementos definidos com as casas do tabuleiro (Figura 1, à esquerda). Durante a especificação de elementos do grafo tipo no GS, certas propriedades são definidas para cada um deles: nome, aparência, cor, posição (horizontal e vertical) e tamanho para os vértices; e nome, aparência, cor, vértice-origem, vértice-destino e tamanho para as arestas.

Para a criação do grafo inicial (Fase 3.3.3), a organização do jogo ao início da partida foi definida. O tabuleiro inicia com 9 casas desocupadas que possuem relações entre si na horizontal (aresta azul), vertical (aresta laranja), diagonal principal (aresta rosa) e diagonal direita (aresta verde) (Figura 1, à direita). Durante a especificação do grafo inicial no GS, são definidas as propriedades nome e posição (horizontal e vertical) para os vértices; e tipo, nome, vértice-origem e vértice-destino para as arestas. Ainda, uma vez que jogos modelados como GGs possuem apenas elementos previamente declarados no grafo tipo, as opções de vértices e arestas para o grafo inicial limitam-se às declarações especificadas na etapa anterior.

Para a criação de regras (Fase 3.3.4), além de “Marcar \times ” e “Marcar \circ ” (Figura 2), foram definidas as regras responsáveis pela indicação do vencedor da partida, ou seja, o momento em que um dos jogadores obtém a sequência de três elementos idênticos em uma das quatro direções permitidas. São especificadas 8 regras: duas possibilidades de marcação (\times e \circ) para quatro possibilidades de direção (horizontal, vertical, diagonal esquerda e diagonal direita). A Figura 5 ilustra duas dessas regras: na parte superior da Figura, a vitória do jogador com marcação \circ , em uma linha vertical; e na parte inferior, a vitória do jogador com marcação \times , em uma linha horizontal.



Figura 5. Regras “Vitória ○ Vertical” (acima) e “Vitória × Horizontal” (abaixo)

As duas próximas fases, Teste do Jogo e Execução da UI, foram analisadas por meio do modo de execução do GS, no qual é possível selecionar o jogo criado⁴ e jogá-lo a partir da seleção das regras especificadas (Figura 6). A última fase, Avaliação da UI, será realizada após a aplicação da atividade proposta ao público-alvo. Ainda que a fase tenha sido mantida idêntica à do processo original, conta com o auxílio de dados coletados automaticamente pelo GS. As informações são separadas por jogador e referem-se a informações como tempos de jogada, erros e tipos de erros durante os mapeamentos, sendo capazes, portanto, de revelar dificuldades e/ou proficiências específicas dos alunos [Silva Junior et al. 2019].

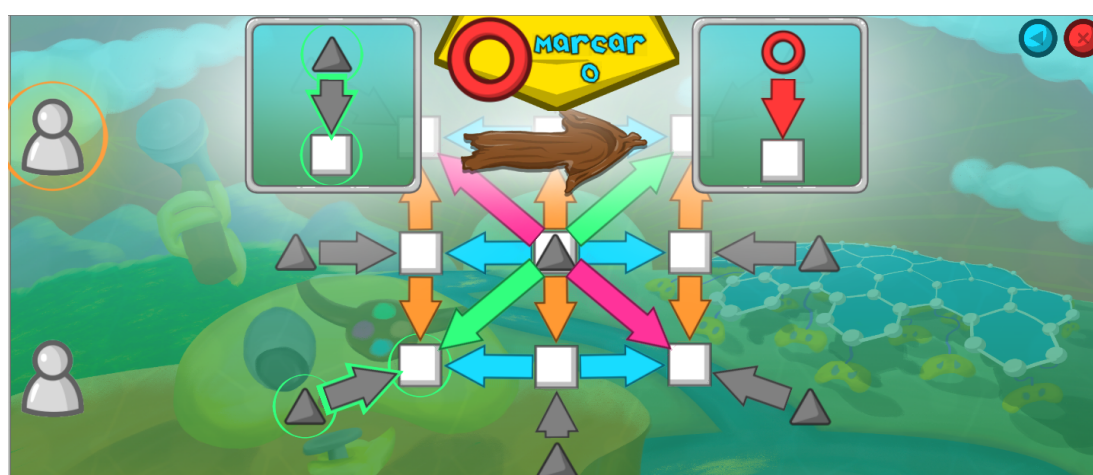


Figura 6. Modo de execução no GS

⁴Disponível em: <https://wp.ufpel.edu.br/pensamentocomputacional/gramestation-pt/jogos/>.

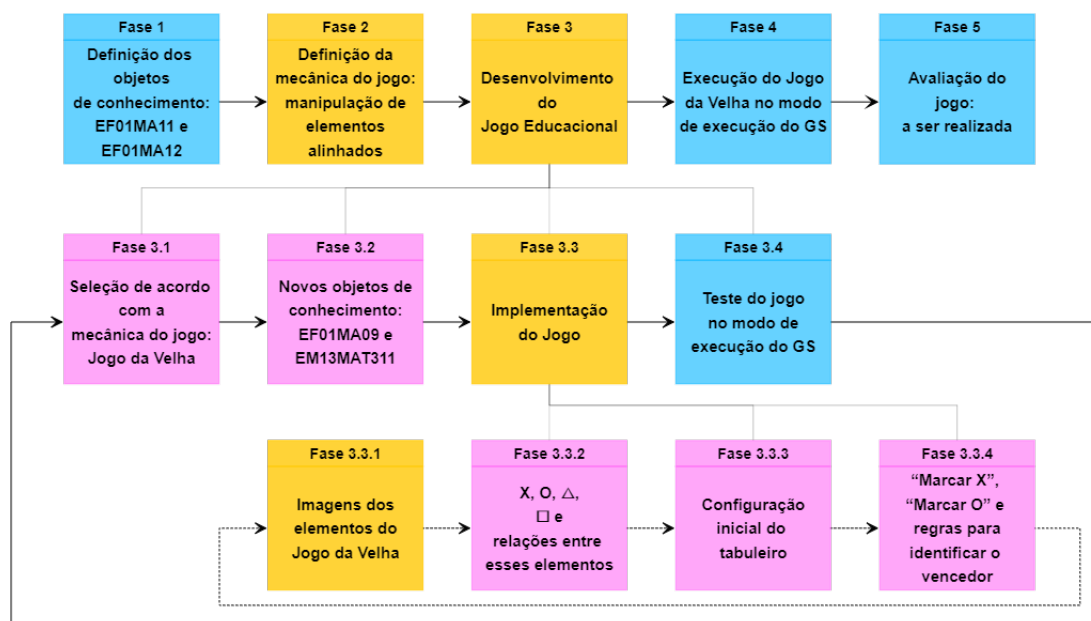


Figura 7. Fluxograma da instanciação do Jogo da Velha

7. Considerações Finais

Neste artigo foi apresentada uma adaptação para o processo de concepção de jogos educacionais ENgAGED, com foco no desenvolvimento de conteúdos curriculares utilizando o GS, motor de jogos baseado em GG. Os jogos educacionais são uma maneira lúdica e dinâmica de exercitar conteúdos em sala de aula e, ainda que muitos sejam desenvolvidos atualmente, poucos contam com o suporte de técnicas consolidadas. Neste contexto, por ser um método avaliado por um painel de especialistas, o processo ENgAGED representa uma estratégia eficaz para a criação de jogos.

A adaptação do processo ENgAGED para o GS facilita a criação de recursos educativos, a promoção de estratégias educacionais alternativas e a utilização de grafos na educação, uma vez que torna possível a relação de determinadas habilidades com a criação de uma GG. Por possuir características específicas, o GS possibilita o desenvolvimento de habilidades – tanto referentes ao PC, conforme defendido em [Silva Junior 2020], quanto referentes à BNCC, como apresentada da Seção 6 – pela maneira que o jogo é construído (baseado em grafos), centrado nas relações de localização dos elementos por meio das arestas no tabuleiro. No entanto, a GG é uma linguagem para representar jogos que possuem elementos (agentes, objetos ou entidades) distribuídos, cuja evolução pode se dar de forma paralela e distribuída. Para jogos cuja evolução é sequencial, a especificação do jogo pode requerer a inserção de estruturas de controle que não dizem respeito ao jogo em si.

O próximo passo deste trabalho refere-se à validação do método proposto por meio de experimento com professores da Educação Básica, tendo como grupo controle aqueles que irão desenvolver jogos educacionais ad hoc, sem a utilização de um método.

Referências

- Abt, C. C. (1987). *Serious Games*. University Press of America.
- Albane, S., Slimani, H., e Kheddouci, H. (2018). Graph grammars according to the type of input and manipulated data: A survey. *Computer Science Review*, 28:178–203.
- Albane, S., Slimani, H., e Kheddouci, H. (2020). A graph grammar and K_4 -type tournament-based approach to detect conflicts of interest in a social network. *Knowledge and Information Systems*.
- Avila, C., Cavalheiro, S., Bordini, A., e Marques, M. (2017). O Pensamento Computacional por meio da Robótica no Ensino Básico - Uma Revisão Sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, volume 28, pages 82–91.
- Battistella, P. E. (2016). *ENgAGED: Um processo de desenvolvimento de jogos para ensino em computação*. PhD thesis, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. Springer US.
- Brasil (2017). Base Nacional Comum Curricular. http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. On-line. Acesso em 08 mar. 2021.
- Brathwaite, B. e Schreiber, I. (2008). *Challenges for Game Designers*. Charles River Media, 1st edition.
- 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.
- Gros, B. (2003). The impact of digital games in education. *First Monday*, 8(7):6–26.
- Gutierrez, J. e Schrum, J. (2020). Generative Adversarial Network Rooms in Generative Graph Grammar Dungeons for The Legend of Zelda. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- Ramos, D. K., Lorenset, C. C., e Petri, G. (2016). Jogos Educacionais: Contribuições da Neurociência à Aprendizagem. *Revista X*, 2(1.2016):1–17.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., e Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11):60–67.
- Schell, J. (2014). *The Art of Game Design: A Book of Lenses*. A K Peters, Ltd., 2nd edition.
- Silva Junior, B. A. (2020). GGasCT: Bringing Formal Methods to the Computational Thinking. Master's thesis, Universidade Federal de Pelotas, Pelotas, RS, Brasil.
- Silva Junior, B. A., Cavalheiro, S. A. C., e Foss, L. (2019). Revisitando um Jogo Educacional para desenvolver o Pensamento Computacional com Gramática de Grafos. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, volume 30, pages 863–872.

Sousa, L. L., Farias, E. J. P., e 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. e 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.

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