

Computational Thinking and Digital Games: Developing Skills With Fun

Original Title: Raciocínio Computacional e Jogos Digitais: Desenvolvendo Habilidades com Diversão

Diego Zabot¹, Saulo Ribeiro de Andrade¹, Ecivaldo de Souza Matos¹

¹ Departamento de Ciência da Computação, Universidade Federal da Bahia (UFBA) – Salvador, BA – Brasil

ARTICLE INFO

Article history:
Received 24 February 2019
Accepted 07 April 2020
Available online 05 May 2020

Keywords:
Computational Thinking
Digital games
Game Mechanics
Systematic mapping
Game based learning

ISSN: 2595-9077

DOI: JCThink.2019.v3.n1.p80

ABSTRACT

INTRODUCTION: Several researchers consider the importance of Computational Thinking being presented and developed from the earliest years of basic education and, furthermore, that digital games can be one of the vehicles to introduce it to children in schools. However, before developing new game solutions for this purpose, it is important to recognize how games can actually contribute to develop Computational Thinking, as well as to identify which skills have been worked on. **OBJECTIVE:** In this sense, this article presents the synthesis of a systematic mapping, whose objective was to identify how digital games can be used to develop Computational Thinking skills. **METHOD:** The objective was met by a systematic literature mapping executed by two reviewers and an expert. **RESULTS:** It was possible to identify some games used to stimulate the development of Computational Thinking skills, as well as the mechanics used by these games. **CONCLUSION:** It has been found that puzzle games are most commonly used to develop skills in Computational Reasoning. It has also been observed that the abilities of Abstraction and Algorithmic Thinking are the main skills developed in these games.

RESUMO

INTRODUÇÃO: Diversos pesquisadores consideram a importância de o Raciocínio Computacional ser apresentado e desenvolvido desde os primeiros anos da educação básica e, além disso, que os jogos digitais podem ser um dos veículos para apresentá-lo às crianças em idade escolar. Todavia, antes de se desenvolver novas soluções de jogos para esse fim, é importante reconhecer como de fato os jogos podem contribuir ao desenvolvimento do Raciocínio Computacional, bem como identificar quais habilidades têm sido trabalhadas. **OBJETIVO:** Nesse sentido, este artigo apresenta a síntese dos resultados de um mapeamento sistemático, cujo objetivo foi identificar como os jogos digitais podem ser usados para o desenvolvimento das habilidades do Raciocínio Computacional. **MÉTODO:** O objetivo foi alcançado mediante mapeamento sistemático de literatura executado por dois revisores e um especialista. **RESULTADOS:** Foi possível identificar alguns jogos utilizados para estimular o desenvolvimento de habilidades do Raciocínio Computacional, assim como as mecânicas usadas por esses jogos. **CONCLUSÃO:** Verificou-se que jogos de quebra-cabeça são os mais utilizados para desenvolver habilidades do Raciocínio Computacional. Também observou-se que as habilidades de Abstração e Raciocínio Algorítmico são as principais habilidades desenvolvidas nesses jogos.

1. Introdução

Em 2006, a pesquisadora Jeannette Wing popularizou o termo *Computational Thinking* (CT) em seu artigo para a revista *Communications of the ACM*. Nesse artigo, Wing indica que o CT inclui uma gama de ferramentas mentais para a resolução de problemas (Wing, 2006).

No Brasil, o CT foi traduzido por diferentes autores como Pensamento Computacional (PC) e Raciocínio Computacional (RC). Neste artigo utilizou-se o termo RC seguindo a argumentação de Paiva *et al.* (2015) de que o pensamento procede o raciocínio, encadeando juízos para estabelecer a verdade ou falsidade de alguma coisa.

Para Wing (2006) o RC pode ser considerado um conjunto de habilidades aplicáveis em qualquer contexto, que todos podem ter proveito em aprender e utilizar e deveria ser adicionado ao lado das habilidades de ler, escrever e aritmética na educação das crianças.

Uma possibilidade para apresentação do RC às crianças apontadas por Wing (2008) seria por meios digitais, dado o nível de familiaridade delas com tecnologias

digitais. Jogos digitais já são usados como mecanismos de aprendizagem em outros aspectos, seja por sua ludicidade intrínseca ou pelo desenvolvimento de habilidades importantes, como atenção e raciocínio lógico. Isto posto, surge um novo desafio: encontrar os elementos necessários para que um jogo possa contribuir no desenvolvimento das habilidades do RC de maneira lúdica e divertida.

Sendo assim, este trabalho apresenta a síntese dos resultados de um mapeamento sistemático de literatura, cujo objetivo foi a identificação de mecânicas usadas por jogos digitais que contribuam para a promoção do desenvolvimento das habilidades do RC. Seguindo as diretrizes apresentadas por Kitchenham *et al.* (2016), buscou-se por artigos em quatro bases de dados digitais. Foram encontrados 36 artigos descrevendo 28 jogos digitais. As mecânicas desses jogos foram caracterizadas. Também foram identificados os relacionamentos entre esses jogos e as habilidades do RC que potencialmente desenvolvidas.

O artigo está organizado em 5 seções. Na Seção 2 é apresentada uma visão geral sobre as habilidades do RC, gêneros de jogos e alguns dos trabalhos relacionados. A Seção 3 apresenta a metodologia de mapeamento sistemático conduzido no contexto de uso de jogos digitais para o desenvolvimento de habilidades de RC e as evidências encontradas. Na Seção 4 são apresentadas e discutidas as respostas às questões de pesquisa. Por fim, na Seção 5 são apresentadas as conclusões e perspectivas para futuros projetos de pesquisa.

2. Background

Nesta seção serão apresentados algumas definições das habilidades do RC e gêneros de jogos encontrados na literatura científica. Essas definições serão usadas como parâmetros norteadores para responder as questões de pesquisa do mapeamento sistemático.

2.1. Habilidades do RC

Existem diferentes definições dos “componentes” do RC na literatura científica: habilidades, categorias, princípios-chave ou técnicas. Há também muita variação sobre quantos e quais são esses componentes. Brackmann *et al.* (2016), por exemplo, destacam quatro habilidades como pilares do RC: decomposição, reconhecimento de padrões, abstração e algoritmos. Aos propósitos desse mapeamento decidiu-se por aderir à lista apresentada por Bocconi *et al.* (2016) que define um grupo de seis habilidades apresentadas a seguir:

- **Abstração.** Simplificar um problema, escolhendo apenas os detalhes necessários para sua resolução. Uma entidade pode ter diferentes abstrações dependendo do problema a ser resolvido.
- **Pensamento Algorítmico.** Saber criar uma sequência clara de passos para alcançar a solução do problema.
- **Automação:** Habilidade de prover instruções com o objetivo de executar tarefas repetitivas de maneira rápida e eficiente.
- **Decomposição.** Saber como separar o problema principal em pedaços menores que podem ser solucionados individualmente, tornando o problema mais fácil de ser solucionado.
- **Depuração.** Habilidade de analisar e avaliar o problema com o intuito de testar, detectar erros e verificar resultados.
- **Generalização.** Habilidade de reconhecer similaridades entre problemas e utilizar estratégias para resolver classes de problemas. Também chamada por muitos autores por “reconhecimento de padrões”.

2.2. Gêneros de jogos digitais

Nessa pesquisa temos utilizado a definição de jogos digitais apresentada por Rogers (2014). Nessa definição, os jogos podem ser classificados em gêneros e subgêneros, dependendo dos elementos de jogo: ação, de tiro, aventura, construção/ gerenciamento, simulação de vida, música/ritmo, party game, quebra-cabeça, esportes, estratégia e simulação de veículos. O autor também especifica que essa classificação é puramente indicativa e alguns jogos podem ser enquadrados em mais gêneros. Além disso, *jogos sérios* e *jogos educacionais* podem se encaixar em diferentes gêneros.

2.3. Trabalhos relacionados

Lockwood e Mooney (2018) apresentam um mapeamento sistemático sobre RC na educação básica com o objetivo de compor um currículo escolar. Os autores identificaram que linguagens de programação (como Python e Java) e ferramentas de criação de jogos (como Scratch¹) são os principais artefatos usados para o ensino do RC. Eles também destacam projetos de pesquisa que descrevem o uso de Minecraft² e atividades de Computação Desplugada³. Porém não são identificados artigos que relatam o uso de jogos com o propósito de desenvolver RC.

Horn *et al.* (2016b) desenvolveram um jogo com o objetivo de estimular habilidades do RC em estudantes do ensino médio usando o problema de teoria dos grafos das Árvore Geradoras Mínimas (AGM). O jogador deve ajudar um rato e um coelho a encontrar comida e os quebra-cabeças são apresentados como grafos. Para encontrar e coletar toda a comida sem esgotar a energia do coelho, o jogador deve indicar o caminho menos custoso usando o rato. O jogo limita o jogador a ver apenas os nós mais próximos, imitando o funcionamento de algoritmos de AGM.

Por fim, destacamos a diferença entre (a) desenvolver habilidades de RC *durante o processo de design* do jogo (usando linguagens de programação ou ferramentas de criação de jogos) e (b) desenvolver habilidades de RC *jogando* jogos. Até onde sabemos, nenhum mapeamento sistemático foi encontrado na literatura científica abordando o segundo propósito, especialmente conectando habilidades do RC com elementos e mecânicas de jogos.

3. Metodologia do Mapeamento Sistemático

Este mapeamento sistemático foi conduzido seguindo as diretrizes apresentadas por Kitchenham *et al.* (2016). O principal objetivo foi a identificação de trabalhos relevantes que usem jogos digitais para potencializar o desenvolvimento das habilidades do RC.

3.1. Planejamento

Destacamos que o objetivo dessa etapa da pesquisa foi a investigação do relacionamento entre mecânicas de jogos usadas em jogos digitais durante a interação com o jogo (ou seja, jogando de fato) e o desenvolvimento das habilidades do RC.

Neste mapeamento levantou-se três questões de pesquisa principais:

- **RQ1** - quais mecânicas de um jogo digital permitem que ele possa ser capaz de desenvolver habilidades do RC?

Rationale: destacar elementos promotores do RC em um jogo.

1 <https://scratch.mit.edu/>

2 <https://minecraft.net/>

3 <https://csunplugged.org>

- **RQ2** - quais são as habilidades tratadas por esses jogos digitais e como elas são estimuladas?

Rationale: identificar as habilidades do RC cobertas por esses jogos.

- **RQ3** - quais são os gêneros de jogos mais usados ou adequados para suportar o desenvolvimento das habilidades do RC?

Rationale: identificar os gêneros de jogos e verificar quais desses gêneros são mais frequentes.

Com o objetivo de responder essas questões de pesquisa definiu-se como palavras-chave: *computational thinking*, *raciocínio computacional*, *pensamento computacional*, *games* e *jogos*. Conduziu-se buscas em bases de dados com o objetivo de identificar trabalhos relevantes sobre RC e jogos digitais utilizando a string de busca apresentada no Quadro 1.

("raciocínio computacional" OR "pensamento computacional" OR "computational thinking") AND ("jogos" OR "games")

Quadro 1. String de busca

Versões anteriores da string de busca usavam o termo "*digital games*" no lugar de "*games*". Por fim aderiu-se pelo último para expandir a cobertura da busca, mesmo se as bases de dados retornassem mais trabalhos e, como consequência, levando mais tempo para processar os resultados durante os filtros.

Essa string foi aplicada as seguintes bases de dados: ACM Digital Library; IEEEExplore Digital Library; Engineering Village; Science Direct. Essas bases de dados foram escolhidas por indexarem a maior quantidade de artigos de Ciência da Computação sobre RC.

A partir das questões de pesquisa, estabeleceu-se critérios de inclusão e exclusão (cf. Quadro 2) de forma que apenas as produções mais relacionadas a pesquisa fossem selecionados. Foram incluídos apenas os artigos publicados após a publicação do artigo de Jeannette Wing (Wing, 2006). No caso de artigos duplicados, apenas a primeira ocorrência foi considerada, de acordo com a ordem das bases de dados listada acima. Nos casos onde o artigo não estivesse livremente disponível buscou-se contatar os autores diretamente (via email ou perfil do Research Gate⁴) para tentar obter o artigo diretamente com os autores. Nos casos em que essa abordagem falhou, o artigo foi desconsiderado (excluído do mapeamento).

Após aplicação dos critérios de exclusão dos artigos identificados, foram aplicados dois filtros para selecionar apenas os trabalhos mais relevantes para esse mapeamento. No primeiro filtro executou-se a leitura do título e do resumo de cada artigo para verificar se o trabalho, *a priori*, atendia aos critérios de inclusão listados anteriormente. No segundo filtro também foi realizada a leitura das seções de introdução e conclusão, uma vez que apenas a leitura superficial do artigo não era suficiente para a seleção dos trabalhos mais relevantes. Por fim, todos os artigos selecionados que passaram nesses filtros tiveram o seu texto completo analisado para extração de dados.

	Inclusão	Exclusão
Origem	Periódicos ou anais de conferências	Literatura cinza, capítulos de livros, artigos duplicados e/ou pagos

4 <https://www.researchgate.net/>

Ano publicação	Entre 2007 e 2017	Anterior a 2007 e posterior a 2017
Idioma	Inglês ou Português	Outros idiomas
Objetivo	Apresentam o uso de jogos digitais para desenvolvimento das habilidades do RC	Objetivo diferente do objetivo deste mapeamento

Quadro 2. Critérios de inclusão e exclusão

Em cada filtro, dois pesquisadores atuaram como revisores colaborativos, analisando cada artigo individualmente e escolhendo aqueles que deveriam ser incluídos no próximo filtro e aqueles que deveriam ser excluídos do mapeamento. Em caso de divergência de opinião entre revisores, um especialista da área de Informática na Educação foi consultado para estabelecer o veredicto final sobre o artigo selecionado. Se, mesmo com a ajuda do especialista, a divergência não fosse resolvida, o artigo passava para o próximo filtro para uma análise mais apurada.

3.2. Resultados

A partir das buscas nas bases de dados apresentadas na seção anterior, 348 publicações científicas foram identificadas. Após aplicação do 1º. filtro, 104 artigos foram selecionados. Desses, 36 passaram pela aplicação do 2º. filtro e foram aprovados para a fase de extração de dados. A Tabela 1 apresenta mais detalhes sobre o processo em cada base de dados.

Base de dados	Identificação	1º Filtro	2º Filtro
ACM	102	47	14
IEEE	59	19	9
Engineering Village	107	32	10
Science Direct	80	6	3
Total	348	104	36

Tabela 1. Artigos identificados e filtrados

A lista completa dos artigos selecionados pode ser visualizada no Apêndice (http://bit.do/ijcthink_jogos_apendice).

Uma análise preliminar mostrou que houve um aumento considerável de publicações sobre o tema em 2017: 14 publicações em contraste com uma média de três publicações em anos anteriores (cf. Figura 1).

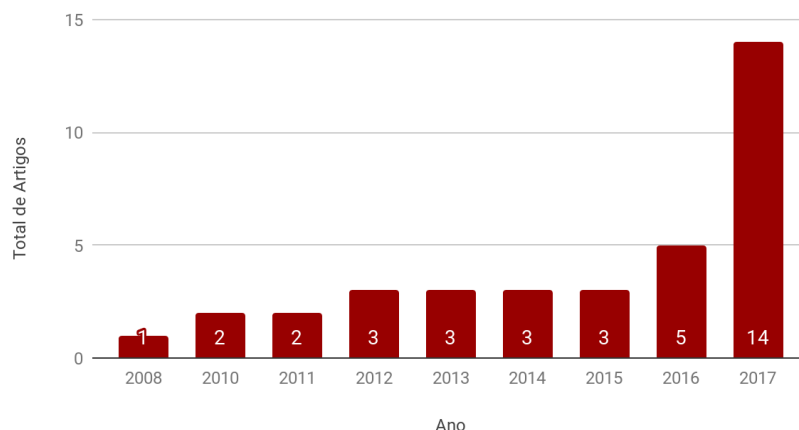


Figura 1. Total de artigos por ano de publicação

A maioria dos artigos são dos Estados Unidos (53,8%) e do Reino Unido (12,8%). O principal público-alvo são estudantes da faixa etária de 5 a 17 anos (92%) com um pico na concentração no intervalo de 11 a 13 anos (42%). Além disso, 85% dos artigos selecionados foram publicados em conferências e 15% em periódicos.

Realizou-se análise sobre os tópicos cobertos nas conferências e periódicos onde os artigos selecionados foram publicados. Além de eventos genéricos de Ciência da Computação, cinco áreas específicas foram identificadas: *visual languages*, *human-computer interaction*, *design with children*, *games*, *education*. A área mais central é *human-computer interaction (HCI)*, com nove artigos em conferências/periódicos com temas de jogos, design com crianças e linguagens visuais: *Visual Languages and Human-Centric Computing*, *Computer-Human Interaction in Play*, *Interaction design and children* e *Human Factors in Computing Systems*. Outras áreas importantes encontradas foram *Digital Games* e *Education*, ambas com oito publicações cada.

3.3. Limitações e ameaças à validade

Uma limitação deste estudo é aquela intrínseca a um mapeamento sistemático de apenas mostrar e categorizar a literatura científica sobre o assunto e permitir apenas discussões superficiais. Há também aspectos de subjetividade dos revisores durante a aplicação dos filtros e extração de dados que pode impactar a validade do estudo.

4. Discussão

Esta seção apresenta as respostas às questões de pesquisa apresentadas na seção 3.

Os artigos referenciam 28 jogos digitais utilizados para promoção de habilidades do RC. Além das seis habilidades apresentadas na subseção 2.1, foram encontrados alguns artigos que apresentam outra habilidade, a **Socialização**. Essa habilidade está diretamente relacionada ao uso de jogos para desenvolver habilidades do RC, por isso, decidiu-se incluí-la. Essa habilidade foi, inicialmente, identificada no trabalho de Kazimoglu *et al.* (2012). De acordo com os autores, a Socialização envolve aspectos de interação entre jogadores (de maneira cooperativa e/ou competitiva). Dessa forma, o jogo deve promover meios de estimular esse tipo de interação. Em *Program Your Robot* os jogadores podem discutir entre eles as estratégias utilizadas para resolver os desafios e buscar maximizar a sua pontuação. Em *Engage* é implementado um modo multijogador que simula a programação em pares.

Uma mecânica identificada é a proposta de um desafio na forma de um problema ou enigma. Tal desafio apresentado no jogo deve ser intrigante e deve estimular o jogador a buscar uma forma de o solucionar.

Além das temáticas dos jogos serem bastante distintas, foi possível agrupar os jogos encontrados em algumas categorias, de acordo com o mecanismo de interação utilizado.

Categoria/Sub-categoria	Jogos
Jogos com interação algorítmica / Interação baseada em blocos de instruções (5)	Dragon Architect, Engage, Kodetu, Labirinto Code.org, Tuk Tuk
Jogos com interação algorítmica / Interação baseada em abstração conceitual (8)	Bots & (Main)Frames, CodeFruits, Lightbot, Meleon, Operação Big Hero: Code Baymax, Program Your Robot, RuBoT, TAPASPlay
Jogos com interação algorítmica / Interação com instruções textuais (2)	CodeMonkey, Gidget
Máquina de estados (4)	CodeCraft, CTArcade, Escape Machine, Robot Chronicle
Estímulo do RC mediante observação e reflexão sobre as mecânicas do jogo (4)	Bounce, Ghost-Busters, Pac-Man, Travel Salesman Problem
Outros (5)	BeadLoom Game, GrACE, Minecraft, Reduct, Zoombinis

Quadro 3. Categorias e jogos

Detectou-se que parte considerável do conjunto amostral, ou seja, dos jogos (15) se enquadram no padrão de “**jogos com interação algorítmica**”, onde parte da tela é dedicada à composição de instruções passo-a-passo para serem executadas pelo personagem do jogo.

Nessa categoria foram identificadas, em geral, mecânicas comuns, que levam a poder desenvolver as seguintes habilidades:

- *depuração* - análise do comportamento do personagem durante a execução do código;
- *raciocínio algorítmico* - identificação e aplicação de estratégias de resolução de problemas;
- *automação* - uso de laços de repetição para movimentos repetidos;
- *generalização* - reconhecimento de padrões de movimento;
- *decomposição* - quando o percurso é mais complexo, o jogo obriga a fragmentar essa complexidade, dividir o percurso em fragmentos e tentar fazer o personagem conseguir se deslocar naquele fragmento.

Nesse grupo podemos distinguir três tipos de interação: *interação baseada em blocos de instruções*, *interação baseada em abstração conceitual* e *interação com instruções textuais*.

Interação baseada em blocos de instruções (Figura 2). Jogos dessa categoria utilizam a biblioteca Blockly⁵ do Google®, ou representação similar, como meio de interação com o jogo. Blockly fornece blocos gráficos para representar conceitos de

5 <https://developers.google.com/blockly/>

códigos que podem ser encadeados para indicar a sequência/passos das ações e construir o algoritmo que soluciona a tarefa dada. Pode-se representar conceitos como expressões numéricas/lógicas, condicionais, laços de repetição, dentre outros. Blockly é usado também em diferentes aplicativos Web, tais como Code.org⁶ e App Inventor⁷. Até mesmo Scratch utiliza-se de uma forma de interação muito parecida com Blockly.

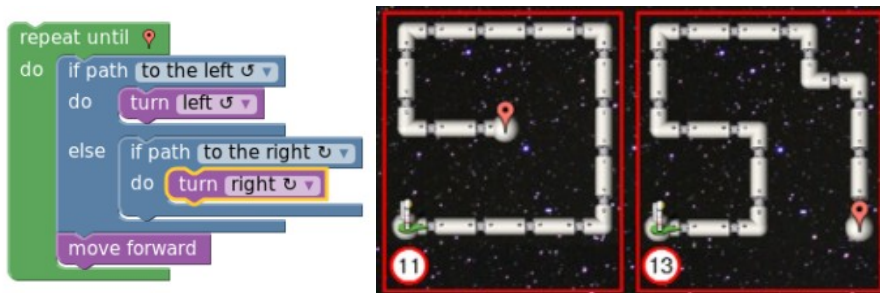


Figura 2. Interação baseada em blocos de instrução (Fonte: Eguiluz *et al.*, 2017)

Em seguida a descrição dos jogos que foram agrupados nessa categoria.

Dragon Architect - O jogador controla um pequeno dragão ao criar programas para construir estruturas dentro do jogo. Existem dois modos de jogo: um modo livre (estilo caixa-de-areia), onde o jogador pode construir o que quiser -- e um modo quebra-cabeça onde o jogador deve resolver desafios envolvendo estruturas de programação (como funções, condicionais e laços de repetição). O êxito na resolução desses desafios garante ao jogador o uso daquele bloco de código no modo livre relacionado ao conceito abordado no desafio. São desenvolvidas as habilidades de raciocínio algorítmico (a partir da identificação e aplicação de estratégias de resolução de problemas) e decomposição (usando o conceito de Divisão e Conquista).

Engage - Os jogadores assumem o papel de cientistas da computação enquanto resolvem um mistério de relevância social. Os jogadores escrevem programas para avançar no jogo. São desenvolvidas as habilidades de raciocínio algorítmico (dado que as soluções dos desafios são expressas na forma de programas) e socialização (a partir de um modo multijogador que simula a programação em pares).

Kodetu - o jogador deve programar um astronauta que deve navegar até o final de um labirinto, sem cair no espaço sideral. São desenvolvidas as habilidades de raciocínio algorítmico (pelo uso de blocos de código e resolução algorítmica dos desafios), decomposição (em especial no 15º desafio do jogo, onde deve-se resolver fragmentos do desafio para posterior resolução completa) e depuração (a partir da análise do comportamento do personagem durante a execução do código).

Labirinto Code.Org - O jogador deve conduzir o personagem até a saída de um labirinto. O personagem é programado a partir do uso de blocos de comando (usando Blockly). São desenvolvidas as habilidades de Raciocínio algorítmico (a partir da construção de algoritmos para solução do desafio) e depuração (a partir da análise do comportamento do personagem durante a execução do código).

Tuk Tuk - é um jogo baseado em blocos de código onde os jogadores usam estes blocos para conduzir um carro e fazer algumas tarefas (como pegar um amigo e levá-lo pro hospital). O jogo desenvolve as habilidades de raciocínio algorítmico (a partir da execução sequencial de cada bloco de código e pelo recurso de blocos de combustível. Tal recurso estimula a criação eficiente dos algoritmos) e depuração (a partir da visualização da execução do código).

⁶ <https://code.org>

⁷ <https://appinventor.mit.edu/explore/>

Interação baseada em abstração conceitual (Figura 3). Jogos dessa categoria utilizam elementos de interação representados por abstrações dos conceitos a serem trabalhados na forma de imagens ou ícones. Neste mapeamento foram identificados jogos que representam instruções como sinais ou ícones gráficos. Destacamos que *Bots & (Main)Frames* apresenta outras duas versões do jogo, usando realidade aumentada e blocos físicos tangíveis.



Figura 3. Interação baseada em abstração conceitual (Fonte: Melcer, 2017b)

Em seguida a descrição dos jogos que foram agrupados nessa categoria.

Bots & (Main)Frames. O jogador programa um robô e deve conduzi-lo até a saída da fase usando comandos como andar, girar 90° e estruturas de programação como laços de repetição, funções e condicionais. Com esse jogo pode-se desenvolver as habilidades de abstração (a partir do encapsulamento dos comandos em funções), raciocínio algorítmico (a partir do desenvolvimento de um algoritmo como solução do desafio), automação (a partir do uso de laços de repetição) e depuração (a partir da análise da execução do programa desenvolvido pelo jogador).

Code Fruits. O jogador deve conduzir um caminhão de frutas do ponto de partida até a casa de uma senhora idosa (ponto de chegada). O jogador programa o trajeto do caminhão usando uma linguagem gestual própria do jogo. O jogo atua no raciocínio algorítmico (pela elaboração de uma sequência de ações como solução do desafio) e depuração (a partir da análise do comportamento do programa desenvolvido).

Lightbot. O jogador controla um robô em um tabuleiro quadriculado. Em cada fase o jogador deve acender todas as caselas azuis do tabuleiro. Para isso, o jogador deve desenvolver um programa para o robô utilizando comandos como andar ou girar 90°, por exemplo. *Lightbot* contribui às habilidades de abstração (a partir do uso desses comandos para resolução dos desafios), raciocínio algorítmico (uma vez que a solução proposta pode ser interpretada como um algoritmo), decomposição (via estratégias de fragmentação do tabuleiro e criação de funções para resolução de tais fragmentos) e generalização (a partir da identificação de padrões nas fases e criação de algoritmos para solucionar tais padrões).

Meleon. Trata-se de um jogo casual para celular. Nele, o jogador deve ajudar um camaleão com deficiência visual a se adaptar ao ambiente. O jogo usa a câmera do celular para determinar as cores que serão usadas nos desafios do jogo. *Meleon* possui dois modos de jogo: challenge mode e exploration mode. Challenge mode atua no desenvolvimento da habilidade de Raciocínio algorítmico (com a apresentação de conceitos de programação), enquanto exploration mode desenvolve a depuração (na exploração via tentativa-e-erro, e análise das entradas e saídas).

Operação Big Hero - Code Baymax. O jogador deve conduzir um robô por um terreno quadriculado até a saída. Programa-se o robô a partir de blocos de comando, de maneira similar ao *Lightbot*. O jogo desenvolve as habilidades de raciocínio algorítmico e depuração, de maneira similar ao *Labirinto Code.org*.

Program Your Robot. O jogador deve programar um robô a fugir de um labirinto. A programação é realizada mediante uso de comandos para movimentar o robô e comandos de programação (como funções, laços de repetição e condicionais). O jogo potencializa o desenvolvimento das habilidades de raciocínio algorítmico (a partir do uso dos comandos para construção de soluções para os desafios), decomposição (a partir do objetivo-problema principal do jogo: conduzir o robô até a saída.), depuração (a partir de um botão de depuração que permite analisar e detectar erros na solução proposta pelo jogador) e socialização (a partir da discussão sobre as soluções propostas e estratégias com outros jogadores).

Rubot. O jogador controla um robô que deve se deslocar com instruções de movimentação e acendeu luzes em posições específicas da tela. O jogo ajuda no desenvolvimento das habilidades de abstração (a partir do processo de criação de funções e do uso de comentários para explicitar o conceito por trás da função criada) e decomposição (a partir da decomposição de ações em ações menores e do reconhecimento de padrões de caselas e criação de funções para resolução desses padrões).

Tapasplay. É um jogo multijogador competitivo. Nele os jogadores assumem o papel de alquimistas com o objetivo de criarem as melhores armas e escudos. Combinam-se fragmentos de metal e aplicam-se transformações para composição das armas e determinação do poder delas. Ao final, os jogadores podem visualizar a batalha entre eles com as armas criadas usando óculos de realidade virtual. O jogo contribui ao desenvolvimento das habilidades de abstração (a partir do processo de formulação da estratégia pelo jogador para maximizar os pontos de força da espada ou escudo criado levando em conta os fragmentos de metal, suas características e seus atributos), raciocínio algorítmico (a partir do processo de formulação da estratégia pelo jogador para maximizar os pontos de força da espada ou escudo criado. Além disso, o jogo oferece um espaço de busca muito grande, forçando o jogador a elaborar a estratégia ao invés de recorrer a uma solução mais direta), automação (a partir do uso de Realidade Virtual para visualização do resultado final), decomposição (a partir dos fragmentos de metal e suas características para uso na forja das armas do jogo e no processo de formulação da estratégia para maximizar os pontos de força da arma), depuração (a partir do uso de realidade virtual para visualização do resultado final) e socialização (a partir da discussão entre jogadores das experiências e estratégias usadas no jogo).

Interação com instruções textuais (Figura 4). Além dos jogos apresentados anteriormente, *Gidget* (Lee *et al.* 2014a) e *CodeMonkey* (Da Silva e Falcão, 2017) utilizam instruções textuais.

Em seguida a descrição dos jogos que foram agrupados nessa categoria.

Codemonkey. O jogador deve conduzir um macaco para coletar banana. Programa-se o macaco a partir de uma linguagem textual de programação. São potencialmente desenvolvidas as habilidades de Raciocínio algorítmico (a partir da construção de algoritmos para solução do desafio) e depuração (a partir da análise do comportamento do personagem durante a execução do código).

Gidget. Os jogadores ajudam um pequeno robô, programando-o para que este complete os objetivos de cada fase. O robô provê um código-fonte que deve ser executado por ele para a conclusão do objetivo da fase, porém o código apresenta falhas. Cabe ao jogador, a partir do feedback das ações do robô, depurar e consertar o código. O jogo parece contribuir ao desenvolvimento da habilidade de depuração.

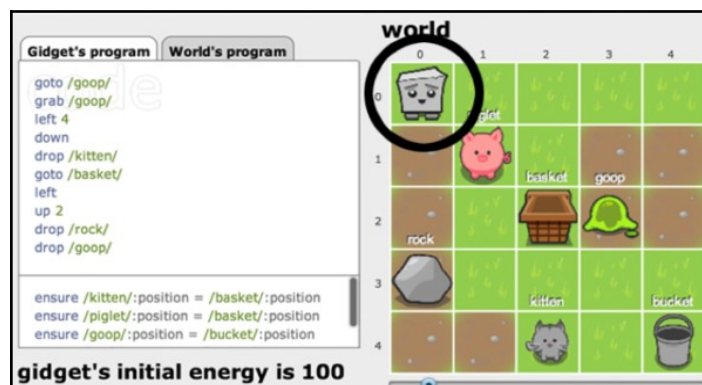


Figura 4. Interação com instruções textuais (Lee et al., 2014a)

Brennan e Resnick (2012) associam elementos do Scratch com aspectos de Raciocínio Computacional, divididos em Conceitos (comandos, variáveis e estruturas de controle), Práticas (desenvolvimento incremental, depuração-teste-execução) e Perspectivas (relacionadas a fatores de socialização) de RC. Apesar destes aspectos serem mais evidentes no âmbito de desenvolvimento de projetos com Scratch, alguns deles refletem habilidades de RC desenvolvidas pelos jogos levantados.

Identificou-se outra categoria de jogos que utilizam **máquina de estados** (Figura 5) para representar o jogo. Nessa categoria foram identificadas as seguintes habilidades:

- *raciocínio algorítmico* - a partir do processo de criação de estratégias especificadas mediante algoritmo via design da máquina de estados;
- *abstração* - a partir da definição individual dos estados ou regras da máquina.

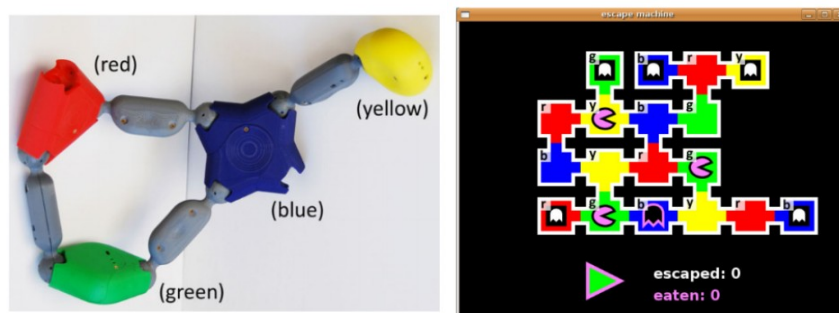


Figura 5. Interface tangível e tela do jogo (Fonte: Weller, Do and Gross 2008)

Em seguida os jogos agrupados nessa categoria.

Codecraft. O jogador deve programar um robô para resolver quebra-cabeças em um ambiente 3D a partir de uma interface de máquina de estados. O jogo parece contribuir ao desenvolvimento das habilidades de Raciocínio Algorítmico (a partir da definição dos estados da máquina e de comandos para transição entre estados), Automação (a partir da própria interface de Máquina de Estados) e decomposição (ao decompor as tarefas nos eventos de cada estado da máquina).

Ctarcade. Os jogadores desenvolvem estratégias para avatares virtuais jogarem jogo-da-velha e competirem contra outros avatares virtuais. jogo parece contribuir ao desenvolvimento das habilidades de abstração, raciocínio algorítmico (a partir do processo de criação de estratégias), depuração (ao permitir a visualização dos resultados das partidas) e generalização (a partir de operações de transformação dos padrões das estratégias definidas pelo jogador).

Escape Machine. O jogador deve conduzir um fantasma por um labirinto para resgatar outros fantasmas, evitando serem devorados por um pac-man. A movimentação

dos fantasmas e do pac-man é determinada pela construção de uma máquina de estados física usada como dispositivo de entrada. São potencialmente desenvolvidas as habilidades de abstração (ao focalizar em um componente da máquina de estados por vez para elaboração da solução), raciocínio algorítmico (a partir da especificação algorítmica do comportamento dos personagens via design da máquina de estados), depuração (pelo *feedback* da movimentação dos personagens após a execução do código) e generalização (a partir das estratégias com base na movimentação específica dos personagens).

Robot Chronicle. O jogador deve programar um robô para solucionar diversos quebra-cabeças do jogo. Tanto o jogador como a interface do jogo objetivam ser o mais simples possível para que o jogador aprenda, aos poucos, conceitos básicos de programação. O jogo ajuda a desenvolver as habilidades de raciocínio algorítmico (dado que a solução dos desafios são expressos na forma de algoritmos) e depuração (a partir do controle da execução do programa).

Outros quatro jogos, *Bounce*, *Pac-Man*, *Travel Salesman Problem* e *Ghost-Busters* foram utilizados na pesquisa de Theodosi e Papadimitriou (2011) com o objetivo de estimular o RC mediante a observação e reflexão sobre as mecânicas do jogo. Uma versão demonstrativa do jogo foi apresentada aos participantes que tinham a tarefa de identificar as regras do jogos, os atores e os comportamentos (colisões, eventos). Nesse caso são desenvolvidas a habilidade de abstração (a partir do isolamento e identificação dos elementos do jogo) e decomposição (analisando singularmente os grupos específicos de elementos, atores, comportamentos, para depois reunir o que foi inferido na composição do jogo).

Bounce. Esse é um remake do clássico jogo *Breakout* desenvolvido em Scratch, em que o jogador deve rebater uma esfera contra uma parede de tijolos coloridos, removendo-os.

Ghostbusters. O jogador deve caçar os fantasmas em um determinado mapa. Os fantasmas aparecem no cenário em posições e períodos de tempo aleatórios.

Pacman. Este é um remake do clássico jogo *Pac-man* desenvolvido em Scratch, onde o jogador navega por um labirinto comendo pílulas enquanto foge de fantasmas.

Travel Salesman (caixeiro viajante). Baseado no problema de computação de mesmo nome, o jogador deve fazer um gato coletar todas as bananas do cenário com o menor número de passos.

Também foram identificados os seguintes jogos que não se encaixam em nenhuma das categorias listadas acima.

Beadloom Game. O jogador deve recriar uma imagem usando o menor número de chamadas de função (movimentos) possível. O jogo parece contribuir ao desenvolvimento da habilidade de automação a partir do uso de funções de iteração para criação de padrões mais complexos. Os autores do artigo chamam essa habilidade de iteração.

Grace. Centrado em um rato e um coelho que estão coletando comida. O jogo trabalha com quebra-cabeças representados por grafos e o objetivo do jogador é encontrar uma árvore geradora mínima que representa o caminho que o coelho gasta menos energia para escavar em busca da comida. O jogo desenvolve as habilidades de abstração (pela representação em grafos dos desafios, fazendo com que o jogador considere apenas a porção relevante do mapa da fase) e raciocínio algorítmico (ao limitar o campo de visão do jogador, levando-o a pensar no passo-a-passo da solução. Dessa forma o jogo também imita os algoritmos de Árvore Geradora Mínima usados na literatura).

Minecraft. Trata-se de um jogo em que os jogadores devem coletar recursos para sobreviver em um mundo gerado proceduralmente. Todos os elementos do cenário do jogo são representados na forma de cubos (pedras, madeira, terra). O jogo permite ao jogador poder construir as suas próprias estruturas usando esses cubos, de maneira similar a construir coisas usando LEGO. No caso da pesquisa de Towhidnejad *et al.* (2014), os autores criaram um mundo dentro do jogo para ensino de conceitos de circuitos lógicos usando as mecânicas de *redstone*, um tipo de bloco que atua como fonte de eletricidade e pode ser usado para construção de circuitos lógicos dentro do jogo. Dessa forma ajuda-se a desenvolver as habilidades de abstração (ao isolar os elementos do cenário e se concentrar naqueles relacionados com circuitos lógicos), depuração (a partir da análise do fluxo de *redstone* do interruptor de origem do fluxo até o receptor) e decomposição (decompor problemas mais complexos em problemas menores, por exemplo A or (B and C)).

Reduct. O jogo utiliza uma mecânica de blocos, focalizado especificamente em ensinar habilidades de programação usando uma representação gráfica dos conceitos tais como funções, instruções, condicionais e funções para mapeamento de conjuntos. O jogo parece contribuir no desenvolvimento das habilidades de raciocínio algorítmico (com desafios cujo espaço de busca é muito grande e mecanismos para limitação do número de ações disponíveis do jogador, forçando-o a planejar cada passo da solução) e decomposição (a partir da mecânica de redução de expressões).

Alguns dos jogos levantados utilizam estratégias para prevenir que os jogadores resolvam os desafios por força-bruta, seja aumentando o espaço de busca da solução, seja limitando a disponibilidade de recursos. Isso faz com que o jogador tenha mais cautela e busque desenvolver soluções ótimas. Outro aspecto importante é o cuidado em buscar um equilíbrio em relação a dificuldade do desafio. Tais desafios não podem ser muito difíceis (desmotivando o jogador e levando-o a desistir do jogo) e não podem ser muito fáceis (levando o jogador a não pensar sobre a solução do problema e, conseqüentemente, buscando a solução via força-bruta).

A maioria dos jogos selecionados (23 dos 28) são jogos de quebra-cabeça. Esse resultado já era esperado, dado que tais jogos são baseados em lógica e preenchimento de padrões, de acordo com a definição de Rogers apresentada na Subseção 2.2 (Rogers 2014).

5. Considerações Finais

Apresentamos neste artigo uma síntese dos resultados de um mapeamento sistemático, cujo objetivo foi a identificação de jogos digitais usados no desenvolvimento das habilidades do RC, mesmo se não tivessem sido desenvolvidos com esse propósito. Este estudo identificou as principais mecânicas de jogos, os gêneros de jogos mais utilizados e quais habilidades são estimuladas e como (cf. Figura 8 para um resumo dos resultados). Também foi possível elaborar uma visão geográfica dos principais países que estão pesquisando o assunto e o público-alvo para o qual esses jogos se destina.

A principal contribuição para pesquisadores e desenvolvedores é um mapeamento dos artigos sobre jogos usados para estimular o desenvolvimento das habilidades do RC, sua categorização e a identificação dos mecanismos aos quais foram aplicados. Esse pode ser o primeiro passo para uma pesquisa mais profunda sobre RC e jogos digitais. Como trabalhos futuros esse mapeamento poderia ser expandido incluindo estudos que abordam o desenvolvimento das habilidades do RC durante o processo de desenvolvimento do jogo. Também seria interessante analisar se jogos não-digitais (como jogos de tabuleiro e jogos de cartas) podem ajudar no desenvolvimento

das habilidades do RC e como, considerando tanto a criação dos artefatos como o desfrutar deles.

Duas questões surgem a partir dos resultados dos gêneros de jogos mais usados nos artigos encontrados: (a) jogos de quebra-cabeça são, de fato, o gênero de jogos mais adequado para suportar o desenvolvimento do RC?; (b) há outros gêneros de jogos que podem ter elementos e mecânicas de jogos intrínsecos que possam suportar o desenvolvimento de habilidades do RC?

Agradecimentos

Os autores agradecem à Pró-Reitoria de Pesquisa, Criação e Inovação da Universidade Federal da Bahia (PROPCI/UFBA) e ao CNPq pelo apoio no âmbito do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI). Esse trabalho foi realizado como parte do edital PROPCI/UFBA 03/2017 - PIBITI, no qual o segundo autor atuou como bolsista.

Referências

- BOCCONI, S., CHIOCCARIELLO, A., DETTORI, G., FERRARI, A., ENGELHARDT, K., KAMPYLIS, P., and PUNIE, Y. (2016). Developing Computational Thinking: Approaches and Orientations in K-12 Education. In *edMedia 2016 – World conference on educational Media and Technology*, p. 13–18.
- BRACKMANN, C. P., CASALI, A., BARONE, D. A. C., and HERNÁNDEZ, S. (2016). Pensamento computacional: Panorama nas américas. In *XVIII Simpósio Internacional de Informática Educativa, SIIE 2016*, page 197.
- BRENNAN, K. and RESNICK, M. (2012). New frameworks for studying and assessing the development of computational thinking. *annual American Educational Research Association meeting, Vancouver, BC, Canada*, p. 1–25.
- HORN, B., HOOVER, A. K., BARNES, J., et al. (2016b). Opening the Black Box of Play: Strategy Analysis of an Educational Game. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play - CHI PLAY '16*. , CHI PLAY '16. ACM.
- KAZIMOGLU, C., KIERNAN, M., BACON, L. and MACKINNON, L. (2012a). A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming. *Procedia - Social and Behavioral Sciences*, v. 47, p. 1991–1999.
- KITCHENHAM, B., BUDGEN, D. and Brereton, P. (2016). *Evidence-based software engineering and systematic reviews*. Vol. 4. CRC Press.
- LOCKWOOD, J. and MOONEY, A. (2018). Computational Thinking in Secondary Education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, 2(1).
- PAIVA, L. F., FERREIRA, A. C., ROCHA, C., BARRETO, J., MELHOR, A., LOPES, R., and MATOS, E. D. S. (2015). Uma experiência piloto de integração curricular do raciocínio computacional na educação básica. In *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação*, p. 1300–1309.
- ROGERS, S. (2014). *Level Up!: The Guide to Great Video Game Design*. John Wiley & Sons.

RODRIGUEZ, F. J., KERBY, N. D. and BOYER, K. E. (2013). Informing the design of a game-based learning environment for computer science: A pilot study on engagement and collaborative dialogue. In CEUR Workshop Proceedings.

ROWE, E., ASBELL-CLARKE, J., GASCA, S. and CUNNINGHAM, K. (2017). Assessing implicit computational thinking in zoombinis gameplay. In Proceedings of the International Conference on the Foundations of Digital Games - FDG '17. , FDG '17. ACM Press.

SILVA, V., SOUZA, A. and MORAIS, D. (2016). Pensamento computacional no ensino de computação em escolas: Um relato de experiência de estágio em licenciatura em computação em escolas públicas. In CEUR Workshop Proceedings.

THEODOSI, A. and PAPADIMITRIOU, V. (2011). The synergy of three: Incorporating games, multimedia and programming in order to improve algorithmic skills. In 5th European Conference on Games Based Learning, ECGBL 2011.

WING, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33.

WING, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725.