

Design of a Logic Game for Introducing Computational Thinking and Computability Concepts in Basic Education

Original Title: Projeto de um jogo de lógica para introduzir o Pensamento Computacional e conceitos de computabilidade na Educação Básica

James Roberto Bombasar¹, André Raabe^{1,2}, Rafael de Santiago³

¹ Mestrado em Computação Aplicada, Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

² Programa de Pós-Graduação em Educação, Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

³ Bacharelado em Ciência da Computação, Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

ARTICLE INFO

Article history:
Received 23 June 2016
Accepted 25 October 2016
Available online 1 August 2017

Keywords:
Computational thinking
Computability
K-12

ISSN: 2594-5602

DOI:
10.14210/ijcthink.v1.n1.p3

ABSTRACT

INTRODUCTION: Since 2006, when Jeannette Wing popularized the term "Computational Thinking" as an essential skill for people in the twenty-first century, there is great interest from the scientific community and governments of various countries in the Computational Thinking exploration in K-12. Although Wing describe Computational Thinking as a process of problem solving that is based on the limits of computing, and more fundamentally addresses the question "What is computable?", little attention has been given to this question in practice. **OBJECTIVES:** The primary objective of this study was to present a theoretical reflection about the importance of the notion of computability for Computational Thinking, and present a game project for its introduction in K-12. **METHODS:** Through the bibliographical research method, a study was carried out on the Computational Thinking, Theory of Computation and technologies used in the exploration of Computational Thinking. **RESULTS:** Based on the studies carried out, the design of a logic game inspired by computational models was elaborated. **CONCLUSION:** The game proposed in this article may represent a new strategy for the Computational Thinking exploration in K-12, because in addition to bringing notions of computability to the fore, it leads to a thinking way based in states changes, which represents an adequate model of the brain conscious functioning.

RESUMO

INTRODUÇÃO: Desde 2006, quando Jeannette Wing popularizou o termo "Computational Thinking" como sendo uma habilidade essencial para as pessoas do século XXI, há um grande interesse da comunidade científica e de governos de diversos países pela exploração do Pensamento Computacional na Educação Básica. Muito embora Wing tenha descrito o Pensamento Computacional como um processo de resolução de problemas que se baseia nos limites da computação, e que mais fundamentalmente aborda a questão "O que é computável?", pouca atenção tem sido dada a este questionamento na prática. **OBJETIVOS:** O principal objetivo deste estudo foi apresentar uma reflexão teórica sobre a importância da noção de computabilidade para o Pensamento Computacional, bem como apresentar o projeto de um jogo para a sua introdução na Educação Básica. **MÉTODOS:** Através do procedimento metodológico da pesquisa bibliográfica, foi realizado um estudo sobre os temas Pensamento Computacional, Teoria da Computação e tecnologias utilizadas na exploração do Pensamento Computacional. **RESULTADOS:** Com bases nos estudos realizados, foi elaborado o projeto de um jogo de lógica inspirado em modelos computacionais. **CONCLUSÃO:** Entende-se que o jogo proposto neste artigo pode representar uma nova estratégia para a exploração do Pensamento Computacional na Educação Básica, pois além de trazer noções de computabilidade à tona, conduz a uma forma de pensar baseada em trocas de estados, que representa um modelo adequado do funcionamento do cérebro consciente.

1. Introdução

Em 2006, Jeannette Wing despertou a atenção da comunidade científica quando propôs a equiparação da importância do Pensamento Computacional (do inglês, *Computational Thinking*) com a das habilidades básicas de leitura, escrita e aritmética. Basicamente, Wing (2006) descreveu o Pensamento Computacional (PC) como sendo a habilidade de empregar de técnicas e conceitos da Ciência da Computação para resolver problemas do cotidiano, tal como decidir entre comprar um carro ou andar de táxi.

Deste então, a ideia de que o PC é uma habilidade essencial para as pessoas do século XXI tem motivado a pesquisa sobre a exploração do PC na Educação Básica. Em 2011, surgiu a primeira proposta de se construir um modelo para o ensino-aprendizagem do PC nas escolas. Por meio de um esforço conjunto entre a CSTA (*Computer Science Teachers Association*), a ISTE (*International Society for Technology in Education*) e líderes de várias escolas, foi elaborado o conjunto de materiais *Computational Thinking Teachers Resources* (CSTA, 2011), que fornece uma definição operacional do PC descompactada em nove conceitos e uma matriz de atividades para a exploração desses conceitos em diferentes níveis escolares.

Nos anos recentes, o interesse pela exploração do PC na Educação Básica tem sido alimentado pela disponibilidade de linguagens de programação de fácil utilização (LYE; KOH, 2014). As estatísticas da revisão sistemática de Bombasar et al (2015) revelam que as linguagens de programação visuais (*Visual Programming Languages - VPLs*), tal como o ambiente Scratch¹, foram as ferramentas mais utilizadas no ensino-aprendizagem do PC de 2006 até o início de 2015. Assim, muito embora Wing (2006) tenha destacado que “*pensar como um cientista da computação significa mais do que ser capaz de programar um computador*”, a programação tornou-se uma das principais atividades utilizadas no ensino-aprendizagem do PC.

Neste contexto, é importante revisar que o PC não se baseia somente nos limites dos computadores, mas também nos limites dos processos de computação, sejam eles executados por máquinas ou seres humanos (WING, 2006). Ribeiro et al (2013) explica que, enquanto a noção sobre o limite dos computadores reside em aspectos mais práticos, tais como tempo e espaço, a noção sobre os limites da computação reside sobre aspectos mais teóricos, tais como a existência de um algoritmo para o problema e a completa execução deste algoritmo por uma máquina.

Assim, considerando que mais fundamentalmente o PC aborda a questão “O que é computável?” (WING, 2006), e que a revisão sistemática de Bombasar et al (2015) demonstra a ausência de ferramentas que tragam essa noção de computabilidade à tona para os alunos da Educação Básica de forma mais direta, sem depender das estratégias utilizadas pelo professor/mediador, o presente artigo tem como objetivo apresentar uma reflexão teórica sobre a importância da noção de computabilidade para o PC, bem como apresentar o projeto de um jogo para a sua introdução na Educação Básica.

As próximas Seções (2 e 3) apresentam os principais conceitos relacionados com o PC, a Teoria da Computação e a Computabilidade. Em seguida (Seção 4), é realizada uma análise da Computabilidade sob a perspectiva do PC. A Seção 5 apresenta uma revisão sobre as principais ferramentas utilizadas no ensino-aprendizagem do PC. A Seção 6 apresenta o projeto de um jogo para a introdução da noção de computabilidade na Educação Básica. Ao final, são apresentadas as considerações finais (Seção 7) e os trabalhos futuros (Seção 8).

2. Pensamento Computacional

Em seu artigo “*Computational Thinking*”, Wing (2006) define o Pensamento Computacional (PC) como sendo a habilidade de recorrer aos conceitos fundamentais da Ciência da Computação para a resolução de problemas, a concepção de sistemas e a compreensão do comportamento humano. Wing (2006) explora o conceito de PC por meio de uma série de questionamentos como: O que os seres humanos podem fazer melhor do que os computadores? O que os computadores podem fazer melhor do que os

¹ <https://scratch.mit.edu/>

seres humanos? Quão difícil é resolver um problema? Qual a melhor maneira de resolvê-lo? O que é computável?

Wing (2006) também apresenta exemplos de aspectos computacionais presentes em problemas do cotidiano, tais como: colocar na mala os itens necessários para uma viagem (busca e *cache*); decidir entre comprar um carro ou andar de táxi (algoritmo online); refazer os passos para localizar um objeto perdido (*backtracking*). Alguns anos mais tarde, a ideia central de Wing foi bem expressa pela Royal Society (2012, p. 29), que definiu o PC como “o processo de reconhecer os aspectos computacionais no mundo que nos rodeia, e aplicar ferramentas e técnicas da Ciência da Computação para compreender e raciocinar sobre os sistemas e processos naturais e artificiais”.

Nas definições apresentadas por Wing (2006) e pela Royal Society (2012), duas características do PC podem ser claramente identificadas: o uso de conhecimentos da Ciência da Computação e a não limitação ao ambiente dos computadores. Assim, o PC pode ser entendido como uma capacidade de aplicar técnicas e ferramentas da Ciência da Computação não somente para resolver problemas que envolvem computadores, mas qualquer problema onde essas técnicas e ferramentas possam ser aplicadas.

Com o objetivo de atribuir um significado comunicável ao conceito de PC para a sua exploração na Educação Básica, a CSTA (2011) definiu operacionalmente o PC como um processo de resolução de problemas que inclui (não somente) as seguintes características: (i) formulação de problemas de forma que computadores e outras ferramentas possam ajudar a resolvê-los; (ii) organização lógica e análise de dados; (iii) representação de dados através de abstrações como modelos e simulações; (iv) automatização de soluções através do pensamento algorítmico; (v) identificação, análise e implementação de soluções visando a combinação mais eficiente e eficaz de passos e recursos; (vi) generalização e transferência de soluções para uma ampla gama de problemas.

Ainda que o processo de formulação de problemas de forma que computadores e outras ferramentas possam ajudar a resolvê-los (item i) implique a noção sobre o limite desses computadores e ferramentas, nenhuma das características apresentadas pela CSTA (2011) implica de maneira direta a noção sobre os limites da computação (O que é computável?). No entanto, é importante perceber que a própria CSTA destaca que a lista por ela fornecida não é exaustiva, sugerindo que outras características possam ser identificadas e investigadas.

Em seu artigo “*Computation and Computational Thinking*”, Aho (2011) entende que parte importante do processo de pensar computacionalmente consiste em encontrar modelos apropriados para formular os problemas e derivar as suas soluções, e destaca que a Teoria da Computação explora justamente as capacidades e limitações desses modelos, citando como exemplos os autômatos finitos e a máquina de Turing, que serão abordados na próxima Seção. Para Aho (2011), esses modelos encontram-se no centro da computação e do PC.

3. Teoria da Computação e Computabilidade

A computação teórica surgiu principalmente do trabalho de Alan Turing e Alonzo Church na década de 1930. Em 1936, Turing introduziu um modelo computacional formal, denominado posteriormente de Máquina de Turing (MT), que permitiu demonstrar a existência de alguns problemas que não têm solução computacional, e assim, fundou um novo campo de pesquisa conhecido como Teoria da Computabilidade (do inglês, *Computability Theory*), que estuda as limitações dos computadores (PETZOLD, 2008; SAVAGE, 1997).

A MT (Figura 1) possui uma unidade de controle, que pode estar em um dentre um número finito de estados representados por q_x , e uma fita infinita, onde cada célula pode conter um dos símbolos de um determinado alfabeto finito ($\{0,1\}$, por exemplo). As células não preenchidas recebem o símbolo especial branco (B). A comunicação entre a unidade de controle e a fita ocorre através de uma cabeça de leitura e escrita, que pode se mover para a esquerda ou direita (HOPCROFT; ULLMAN; MOTWANI, 2002).

A cada símbolo lido, a MT vai para o próximo estado, escreve um símbolo na fita (podendo manter símbolo lido) e move-se para a esquerda ou direita, conforme a lógica programada para o estado em que ela se encontra. A computação é encerrada quando a unidade de controle assume um estado de aceitação ou rejeição. Assim, a programação de uma MT consiste basicamente na definição da sua função de transição, onde para cada tupla *estado x símbolo* são definidos o próximo estado, o símbolo a ser escrito e a direção para a qual a cabeça de leitura deve se movimentar (SIPSER, 1997).

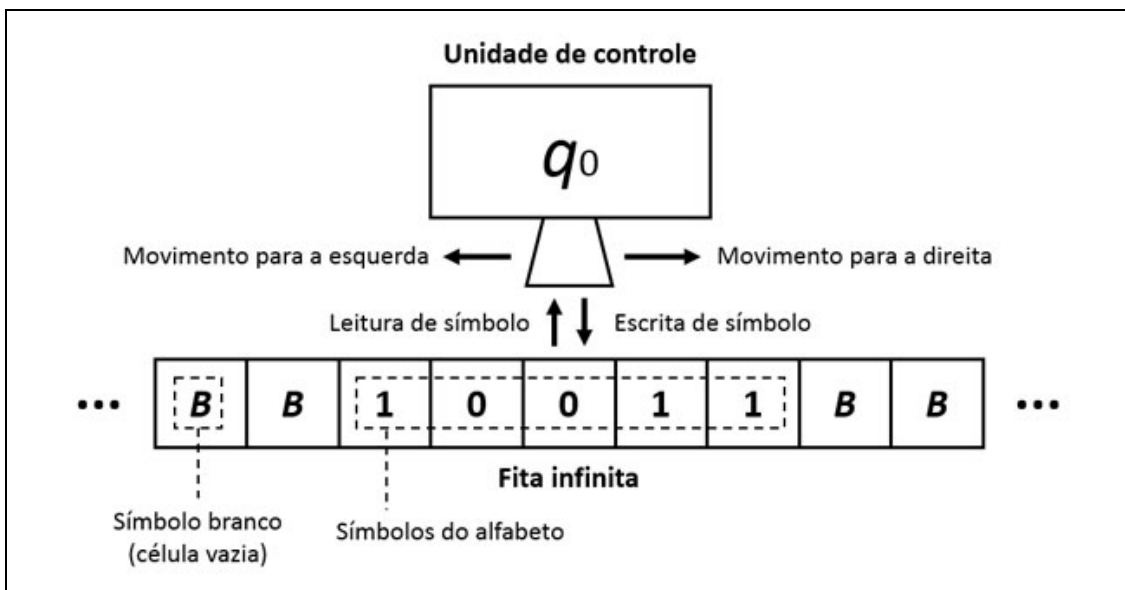


Figura 1. Máquina de Turing.

Fonte: Adaptado de Hopcroft, Ullman e Motwani (2002), Savage (1997) e Sipser (1997).

Nas décadas seguintes (1940 e 1950), modelos computacionais mais simples, originalmente propostos para modelar o funcionamento da parte consciente do cérebro, foram objeto de estudo de diversas pesquisas. Esses modelos, denominados Autômatos Finitos (AFs) (Figura 2), permitem representar sistemas ou componentes que a todo momento podem ser considerados como estando em um de um número finito de estados, e que se deslocam de estado para estado em resposta a “estímulos” externos (HOPCROFT; ULLMAN; MOTWANI, 2002).

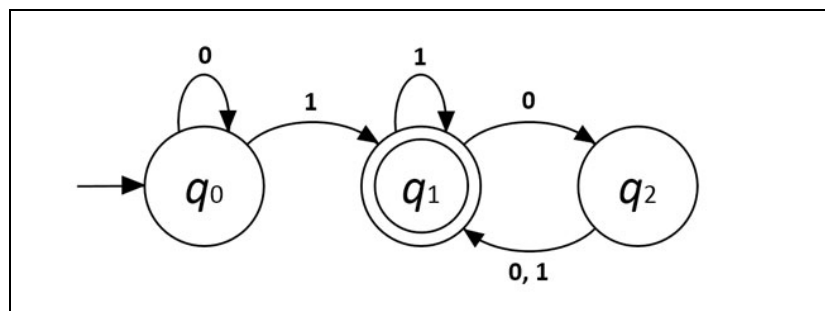


Figura 2. Autômato finito.
Fonte: Sipser (1997).

Segundo Sipser (1997), a computabilidade e os autômatos são áreas da Teoria da Computação que estão diretamente ligados à questão: *quais são as capacidades fundamentais e limitações dos computadores?* Enquanto a computabilidade estuda os limites da computação, os autômatos fornecem a definição precisa e formal do que é um computador.

4. O Conceito de Computabilidade sob a Perspectiva do Pensamento Computacional

No artigo “*Computational Thinking*”, Jeannette Wing (2006) descreve o Pensamento Computacional (PC) como uma habilidade de resolução de problemas que se baseia no nos limites da computação, e que mais fundamentalmente deve ser capaz de lidar com a questão “O que é computável?”. Wing entende que, para se estabelecer a dificuldade de um problema, deve-se levar em conta o poder da “máquina” que irá executar a solução, tendo em vista o seu conjunto de instruções, as suas limitações de recursos e seu ambiente operacional (WING, 2006).

O conceito de “máquina” foi explorado por Wing dois anos mais tarde no artigo “*Five deep questions in computing*” como sendo o conjunto de substratos físicos ou abstratos para a resolução de um problema, e que pode incluir desde a internet até um computador quântico. Cada um dos substratos possui um modelo próprio de “Máquina de Turing”, que determina o seu limite de computação (WING, 2008).

Assim, para estabelecer se um problema é ou não computável, torna-se necessário o entendimento sobre os limites computacionais dos substratos disponíveis como um todo, sejam eles utilizados sequencial ou paralelamente. Como exemplo, Wing (2008) cita um ser humano e um computador trabalhando como uma única “máquina” para resolver um problema que não seriam capazes de resolver sozinhos.

Ribeiro et al (2013) também descrevem o PC como uma habilidade que exige a compreensão sobre os limites da computação e dos computadores. Enquanto a noção sobre os limites da computação reside sobre aspectos mais teóricos, tais como a existência de um algoritmo para o problema e a possibilidade da completa execução deste algoritmo por uma máquina, a noção sobre o limite dos computadores reside sobre aspectos mais práticos, tais como tempo e espaço (RIBEIRO et al, 2013).

A exigência da compreensão sobre os limites dos computadores também pode ser verificada no conjunto de materiais fornecido pela CSTA (2011), que define o PC como um processo de resolução de problemas que inclui como característica a formulação de problemas de modo que computadores e outras ferramentas possam ajudar a resolvê-los. Logicamente, para que ferramentas possam ajudar a resolver problemas, torna-se necessário compreender as limitações dessas ferramentas, seja em relação às suas linguagens ou capacidades.

O Quadro 1 apresenta uma síntese das habilidades relacionadas com a noção de computabilidade descritas em Wing (2006), Ribeiro et al (2013) e CSTA (2011), bem como situações exemplo que podem auxiliar a desenvolver o pensamento computacional associado a cada noção.

Habilidade	Exemplo
Determinar o poder da máquina que irá executar a solução tendo em vista	Determinar se um valor X pode ser sacado em um caixa eletrônico com cédulas de 20 e 50

as suas limitações.	reais.
Determinar o poder da máquina que irá executar a solução tendo em vista o seu conjunto de instruções.	Determinar se um cálculo pode ser realizado através da planilha Excel.
Determinar o poder da máquina que irá executar a solução tendo em vista o seu ambiente operacional.	Identificar “gargalos” em uma linha de produção.
Verificar a existência de um algoritmo para o problema.	Distribuir 25 pessoas em grupos com o mesmo número de pessoas.
Verificar a possibilidade da completa execução de um algoritmo por uma máquina.	Distribuir aleatoriamente 24 pessoas entre 3 grupos e repetir o sorteio até que todos os grupos contenham um número ímpar de pessoas.
Compreender os limites dos computadores em termos de tempo e espaço.	Avaliar a viabilidade de se desenvolver um programa que analise todos os possíveis trajetos entre Itajaí/SC e Salvador/BA.
Formular problemas de modo que computadores e outras ferramentas possam ajudar a resolvê-los.	Obter o resultado de 2 x 999 através de um aplicativo que realiza somente operações de soma.

Quadro 1. Habilidades relacionadas com a noção de computabilidade.

5. Ferramentas para a Exploração do Pensamento Computacional

Com o objetivo de identificar as tecnologias utilizadas pelos professores ao se trabalhar com os conceitos do Pensamento Computacional (PC) em sala de aula, Mannila et al (2014) aplicou um questionário online voltado para professores de diversos países da Europa. Com base na síntese de 961 respostas, os resultados do estudo revelaram o uso de diversos tipos de ferramentas, tais como linguagens de programação, recursos da web, robótica e simulação. Lye e Koh (2014), por sua vez, apresentam outro estudo que investiga como a programação foi incorporada nos currículos da Educação Básica. Nos resultados, destacam-se as ferramentas Logo² e Scratch³.

Mais recentemente, Bombasar et al (2015) realizou uma revisão sistemática com o objetivo de identificar as principais ferramentas utilizadas no ensino-aprendizagem do PC de 2006 até o início de 2015. A Figura 3 apresenta as ferramentas que tiveram ao menos dois relatos de utilização.

² <http://el.media.mit.edu/logo-foundation/>

³ <https://scratch.mit.edu/>

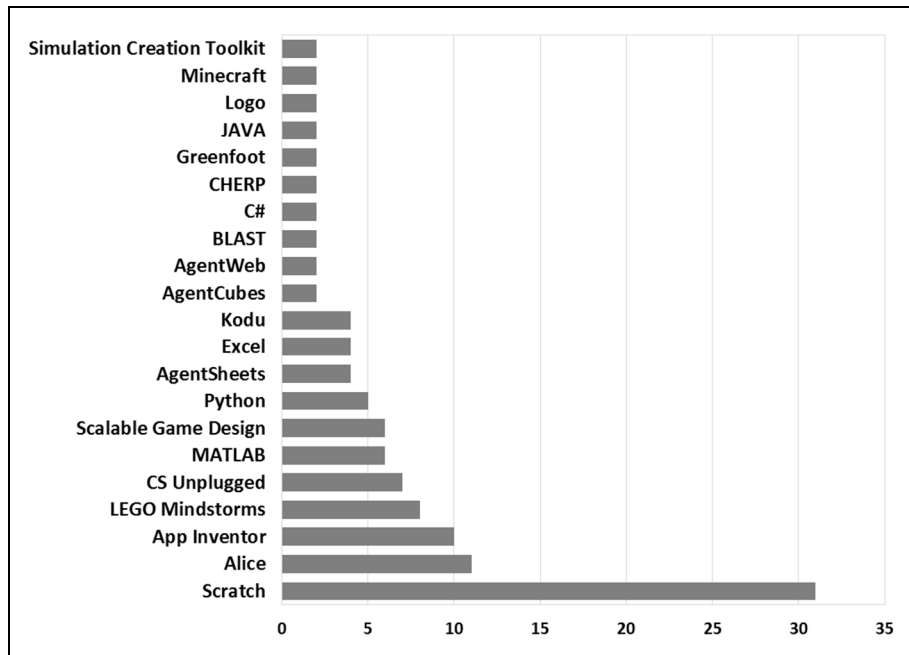


Figura 3. Ferramentas utilizadas em práticas de ensino-aprendizagem do pensamento computacional x quantidade de relatos de utilização.

Fonte: Adaptado de Bombasar et al (2015).

O Scratch é uma linguagem de programação visual (VPL - *Visual Programming Language*) que permite a construção de programas através de blocos. Foi desenvolvida em 2003 pelo *Media Lab* do *Massachusetts Institute of Technology* (MIT) para pessoas que estão começando a programar. Alice⁴ é um ambiente de programação que permite criar animações 3D, e foi projetado para ser o primeiro contato de um estudante com a programação orientada a objetos.

A ferramenta App Inventor⁵, também desenvolvida no MIT, permite aos novatos em programação criar aplicativos para dispositivos Android. A linha de brinquedos LEGO Mindstorms⁶, por sua vez, é o resultado de uma parceria entre o grupo LEGO e o MIT. Trata-se de uma ferramenta que permite criar robôs que executam funções básicas pré-programadas.

Em seguida, com 7 relatos de utilização, está a ferramenta CS Unplugged⁷, um projeto desenvolvido na universidade de Canterbury para o ensino de Ciências da Computação sem o uso do computador. Scalable Game Design⁸ (SGD) e MATLAB⁹ apresentaram 6 relatos de utilização. O MATLAB é um software interativo para cálculos numéricos de alta performance, muito utilizado em cursos de Engenharia. SGD é um projeto da Universidade do Colorado que visa reinventar o ensino de computação em escolas públicas nos EUA através da programação de jogos. Python¹⁰ é uma linguagem de programação que prioriza a legibilidade do código, possui uma sintaxe clara e concisa.

⁴ <http://www.alice.org/>

⁵ <http://appinventor.mit.edu/>

⁶ <http://www.lego.com/en-us/mindstorms/>

⁷ <http://csunplugged.org/>

⁸ http://sgd.cs.colorado.edu/wiki/Scalable_Game_Design_wiki

⁹ <http://www.mathworks.com/products/matlab/>

¹⁰ <https://www.python.org/>

Com 4 relatos de utilização, aparecem as ferramentas AgentSheets¹¹, Excel¹² e Kodu¹³. AgentSheets é um ambiente que permite criar jogos e simulações baseados em agentes e publicá-los na web. O Excel é um construtor de planilhas com ferramentas de cálculo e de construção de gráficos. A ferramenta Kodu permite que crianças criem jogos no PC e no Xbox através de uma VPL simples.

Dentre as ferramentas com apenas dois relatos de utilização foram identificadas 5 linguagens e ambientes de programação (JAVA; C#; Logo; Greenfoot; CHERP), 1 algoritmo de busca (BLAST) 1 ambiente de simulação (Simulation Creation Toolkit), 1 jogo (Minecraft) e 2 ambientes para a criação de jogos (AgentWeb; AgentCubes).

Uma lacuna¹⁴ que pode ser identificada na revisão sistemática de Bombasar et al (2015) é a ausência de ferramentas que utilizem os modelos apresentados na Seção 3 e que, ao mesmo tempo, sejam voltadas para públicos mais jovens. Assim, considerando a importância da noção de computabilidade para o PC demonstrada na Seção 4, a próxima Seção apresenta o projeto de um jogo para a introdução de noções de computabilidade na Educação Básica.

6. Computability Game: um Jogo para a Introdução do Pensamento Computacional e Noções de Computabilidade na Educação Básica

Apesar dos modelos da Teoria da Computação permitirem definir precisa e formalmente o que uma “máquina” é capaz de fazer, eles envolvem conceitos que podem ser de difícil exploração na Educação Básica.

No entanto, há um trabalho sobre lógica de computabilidade desenvolvido por Giorgi Japaridze, pesquisador em lógica e ciência da computação teórica na Universidade de Villanova (Filadélfia), que tem o objetivo de estabelecer as bases para uma Teoria da Computação interativa.

Na lógica de computabilidade desenvolvida por Japaridze (2003), a noção de problemas computacionais é formalizada através de um jogo entre máquina e ambiente (Figura 3), e a computabilidade é entendida como a existência de uma Máquina de Turing (MT) interativa que ganha o jogo contra qualquer possível cenário.

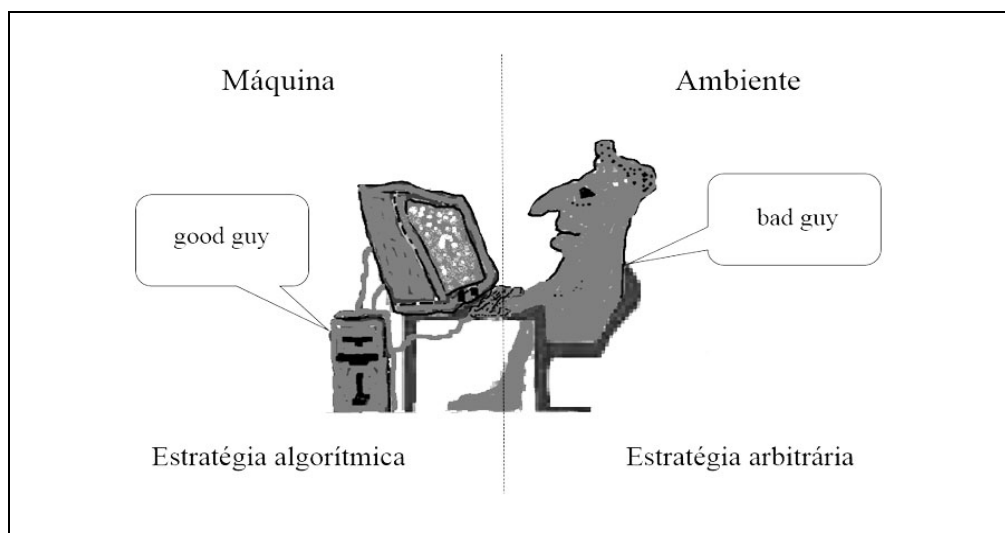


Figura 3. Problemas computacionais tradicionais como jogos.

¹¹ <http://www.agentsheets.com/>

¹² <https://products.office.com/excel>

¹³ <http://www.kodugamelab.com/>

¹⁴ Conforme os dados disponibilizados em: <https://goo.gl/At4WnJ>.

Fonte: Adaptado de Japaridze (2016).

A ideia do jogo apresentado Japaridze (2003) apoia-se no fato de que a maioria das tarefas que as máquinas e seres humanos realizam são interativas, com múltiplas e/ou contínuas entradas e saídas intercaladas. Enquanto o comportamento humano pode ser arbitrário, o da máquina só pode seguir estratégias baseadas em algoritmos.

A máquina e o ambiente interagem através de ações/movimentos observáveis (Figura 4), representados por meio de strings finitas de um alfabeto. Os movimentos realizados pela máquina são chamados de saídas (*outputs*) e os realizados pelo ambiente de entradas (*inputs*). A máquina perde o jogo quando há uma entrada do ambiente e nenhuma saída é gerada por ela.

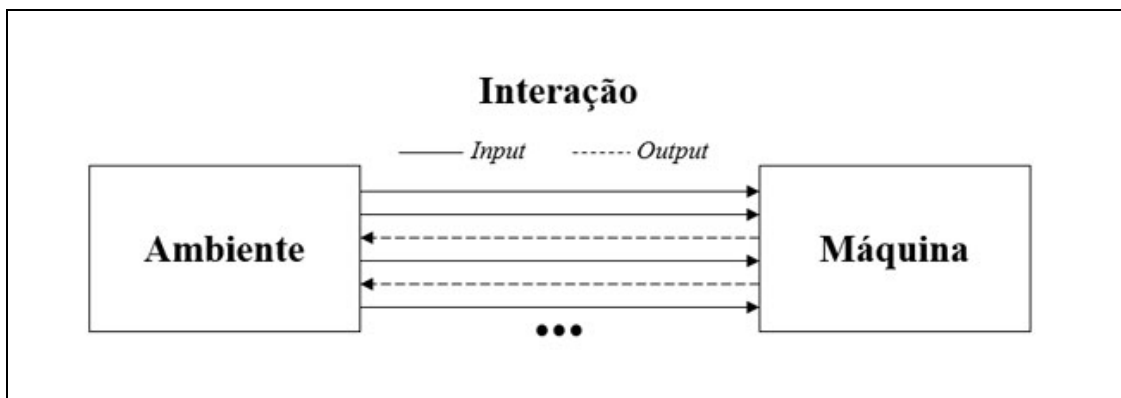


Figura 4. Processo interativo entre ambiente e máquina na lógica de computabilidade.

Fonte: Adaptado de Japaridze (2016).

Assim, surgiu a ideia do *Computability Game* (Figura 5), um jogo de lógica inspirado nos modelos da Teoria da Computação e na lógica de computabilidade de Japaridze (2003), em que o jogador é desafiado a fornecer sequências de símbolos para uma máquina de tal forma que o seu programa não consiga processá-las. Trata-se de um jogo do gênero quebra-cabeça (*puzzle*) que tem como proposta o desenvolvimento de noções de computabilidade em alunos a partir dos últimos anos do Ensino Fundamental.

O jogador encontra-se na parte inferior do cenário e possui as ferramentas localizadas na área verde. A máquina encontra-se na parte superior e está representada pela cor bege. O cenário também possui uma fita, onde os símbolos são escritos pelo jogador e lidos pela máquina.

A máquina pode assumir 12 diferentes estados, que estão representados através de círculos. O estado atual da máquina é sinalizado com a cor verde, e os estados finais (em que a máquina vence o jogo) com a cor preta e um ícone que indica perigo. A transição entre os estados está representada através de setas. Cada seta exibe o símbolo necessário para que a transição ocorra.

O desafio inicia com o jogador escrevendo uma sequência de símbolos na fita através das ferramentas disponíveis na área inferior da tela. Ao terminar a escrita dos símbolos, o jogador clica em “*Compute!*” para que a máquina inicie a leitura da fita. A cada símbolo lido, a máquina verifica a seguinte condição: existe alguma transição para o símbolo a partir do estado atual (marcado em verde)?



Figura 5. Computability Game: cenário do jogo.

Se a condição não for satisfeita, a máquina perde o jogo e o jogador recebe as moedas coletadas até então, caso contrário, a máquina vai para o próximo estado e uma nova condição é verificada: o estado é final? Se a condição for satisfeita, a máquina vence o jogo e o jogador não recebe nenhuma moeda, caso contrário, uma moeda é coletada e o processamento do próximo símbolo da sequência é iniciado.

O design gráfico utiliza componentes da biblioteca de texturas Mobile Game GUI¹⁵. A concepção da interface e dos níveis do jogo está baseada no conceito de jogos casuais, que segundo Fernandes et al (2003) apresentam como características o design envolvente e regras bastante simples, que não exigem dons especiais e proporcionam uma curva de aprendizado rápida e intuitiva.

O princípio “*low floor, high ceiling*”¹⁶, normalmente utilizado na construção de ambientes de programação para públicos mais jovens, foi aplicado ao jogo através da implementação de 23 níveis organizados em ordem crescente de dificuldade. Os níveis estão distribuídos em quatro mundos: Autômatos Finitos Determinísticos, Autômatos Finitos Não Determinísticos, Alan Turing e Máquina de Turing.

A cada novo mundo, a máquina melhora as suas habilidades (Figura 6). No mundo primeiro mundo, a máquina consegue assumir apenas um estado a cada símbolo lido. No segundo, a máquina pode assumir dois ou mais estados com a leitura do mesmo símbolo. No terceiro, a máquina é capaz de alterar o sentido de leitura da fita a cada transição. O novo sentido de leitura é indicado através de uma dupla seta marrom sobre a seta de transição cinza. No último mundo, além de alterar o sentido de leitura, a

¹⁵ <http://graphicburger.com/mobile-game-gui/>

¹⁶ Significa que, embora o ambiente deva ser fácil para um iniciante (*low floor*), também deve ser poderoso e extenso o suficiente para satisfazer programadores mais avançados (*high ceiling*) (GROVER e PEA, 2013).

máquina também é capaz substituir o símbolo lido por um novo símbolo, que é indicado ao lado do símbolo de transição.

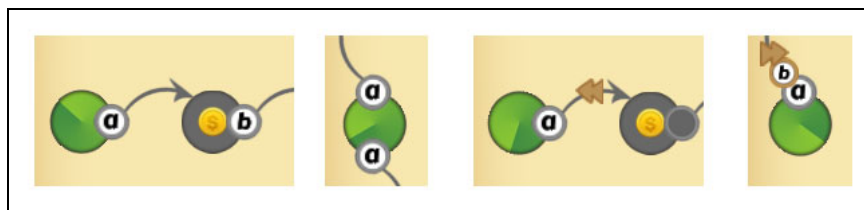


Figura 6. Evolução das habilidades da máquina ao longo dos níveis do jogo.

Atualmente, o jogo encontra-se em suas fases finais de desenvolvimento. Um experimento para a avaliação do efeito do jogo sobre a noção de computabilidade de aproximadamente 60 alunos dos últimos anos do Ensino Fundamental será realizado no segundo semestre de 2016.

7. Considerações Finais

O objetivo deste artigo foi apresentar uma reflexão teórica sobre a importância da noção de computabilidade para o Pensamento Computacional (PC), bem como apresentar o projeto de um jogo para a sua introdução na Educação Básica.

Com o grande interesse dos educadores e governos de diversos países pela inclusão do PC nos currículos da Educação Básica, o desenvolvimento e avaliação de metodologias e ferramentas para o ensino do PC com públicos mais jovens tornou-se um problema especialmente relevante para a área da Informática na Educação. Segundo Grover e Pea (2013), apesar de existirem alguns avanços na definição de currículos, grandes lacunas ainda precisam ser preenchidas através de investigações empíricas.

Investigar abordagens adequadas para a exploração do PC na Educação Básica é uma tarefa complexa, pois conforme ilustra a matriz de atividades do *Computational Thinking Teachers Resources* da CSTA (2011), as diversas características e conceitos do PC precisam ser trabalhados de forma diferenciada com os públicos de cada nível escolar. Além disso, o PC é uma habilidade multidisciplinar, e que pode ser utilizada não somente para solucionar problemas, mas também para criar novos conhecimentos e melhorar a compreensão sobre o poder e os limites da computação (SEEHORN, 2011).

A revisão sistemática de Bombasar et al (2015) demonstra que as principais ferramentas utilizadas no ensino do PC na última década foram as linguagens de programação visuais, que conduzem os alunos a um pensamento computacional baseado em laços de repetição, desvios condicionais e outras estruturas algorítmicas herdadas das linguagens de programação tradicionais.

Assim, entende-se que o jogo proposto neste artigo pode representar uma nova estratégia para a exploração do PC na Educação Básica, pois além de trazer a noção de “O que é computável?” (WING, 2006) à tona de forma mais direta, conduz a uma forma de pensar baseada em trocas de estados, que para Zylberberg et al (2011) representa um modelo adequado do funcionamento do cérebro consciente.

Além disso, ambientes *Game-Like* podem ajudar alunos da Educação Básica a relacionar conceitos abstratos com a experiência concreta do jogo (LI; CHENG; LIU, 2013). Estudos recentes (HUNG; HUANG; HWANG, 2014; WOO, 2014) apresentam resultados positivos da aprendizagem baseada em jogos digitais (DGBL - *Digital Game-Based Learning*), seja no resultado da aprendizagem, no sucesso cognitivo dos alunos ou na simples motivação.

8. Trabalhos Futuros

O presente artigo apresentou o atual estado de um trabalho de pesquisa que investiga o seguinte problema: *de que forma a noção de computabilidade pode ser abordada com estudantes da Educação Básica?* Como próximas etapas, tem-se: (i) a construção de um instrumento que permita avaliar a noção de computabilidade; (ii) a realização de um experimento com alunos da Educação Básica para a aceitação ou rejeição da seguinte hipótese: *Os alunos que utilizam o Computability Game apresentam uma melhor noção de computabilidade em relação aos alunos que não utilizam o Computability Game.*

Agradecimentos

Os autores agradecem pela bolsa de Taxas Escolares concedida pela CAPES, através do Programa de Suporte à Pós-Graduação de Instituições de Ensino Particulares (PROSUP).

Referências

- AHO, A. V. Ubiquity Symposium: Computation and Computational Thinking. Ubiquity, New York, January 2011.
- BOMBASAR, J. R.; RAABE, A.; SANTIAGO, R.; MIRANDA, E. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está Alan Turing? Anais do Congresso Brasileiro de Informática na Educação (CBIE). Maceió: [s.n.]. 2015. p. 81-90.
- CSTA. Computational thinking teacher resources second edition. CSTA, 2011. Disponível em: <http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf>. Acesso em: 25 Maio 2016.
- GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. Educational Researcher, v. 42, n. 1, p. 38-43, 2013.
- HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. Introdução À Teoria Dos Autômatos, Linguagens E Computação. 2. ed. Rio de Janeiro: Elsevier, 2002.
- HUNG, C. M.; HUANG, I.; HWANG, G. J. Effects of digital game-based learning on students' self-efficacy, motivation, anxiety, and achievements in learning mathematics. Journal of Computers in Education, v. 1, n. 2, p. 151-166, November 2014.
- FERNANDES, A. M. R. et al. Jogos Eletrônicos: mapeando novas perspectivas. Florianópolis: Visual Books, 2009.
- JAPARIDZE, G. Introduction to computability logic. Annals of Pure and Applied Logic, v. 123, p. 1-99, 2003.
- JAPARIDZE, G. Games. Disponível em: <<http://www.csc.villanova.edu/~japaridz/CL/Episodes/6.ppt>>. Acesso em: 25 Maio 2016.
- LI, Z.; CHENG, Y.; LIU, C. A constructionism framework for designing game-like learning systems: Its effect on different learners. British Journal of Educational Technology, v. 2, p. 208-224, March 2013.

LYE, S. Y.; KOH, J. H. L. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, v. 41, p. 51-61, December 2014.

MANNILA, L. et al. Computational Thinking in K-9 Education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference - ITiCSE-WGR '14*. New York: ACM. 2014. p. 1-29.

PETZOLD, C. *The annotated Turing: a guided tour through Alan Turing's historic paper on computability and the Turing machine*. Indianapolis: Wiley Pub, 2008.

RIBEIRO, L. et al. Computational Thinking: Possibilities and Challenges. 2nd Workshop-School on Theoretical Computer Science. Rio Grande: WEIT - 2nd Workshop-School on Theoretical Computer Science. 2013. p. 22-25.

ROYAL SOCIETY. *Shut down or restart? The way forward for computing in UK schools*. Royal Society. London. 2012.

SAVAGE, J. E. *Models of Computation: Exploring the Power of Computing*. Boston: Addison Wesley, 1997.

SEEHORN, D.; EDITOR. *K-12 Computer Science Standards - Revised 2011: The CSTA Standards Task Force*. CSTA & ACM. 2011.

SIPSER, M. *Introduction to the theory of computation*. Boston: PWS Publishing Company, 1997.

WING, J. M. Computational Thinking. *Communications of the ACM*, March 2006.

WING, J. Five deep questions in computing. *Communications of the ACM*, New York, v. 51, n. 1, p. 58-60, January 2008.

WOO, J.-C. Digital Game-Based Learning Supports Student Motivation, Cognitive Success, and Performance Outcomes. *Journal of Educational Technology & Society*, v. 17, n. 3, p. 291-307, July 2014.

ZYLBERBERG, A. et al. The human Turing machine: a neural framework for mental programs. *Trends in Cognitive Sciences*, v. 15, n. 7, p. 293-300, July 2011.