

Jogos Infantis e Pensamento Computacional: em Busca de um Conjunto de Diretrizes de Design

João Paulo da Silva, Taciana Pontual Falcão

¹Departamento de Estatística e Informática (DEINFO). Universidade Federal Rural de Pernambuco (UFRPE), Rua Dom Manoel de Medeiros, s/n, Dois Irmãos, CEP: 52171-900, Recife–PE – Brasil

jpaulo_silva@yahoo.com.br, taciana.pontual@ufrpe.br

Abstract. *Computational thinking has become an ability increasingly important in contemporary society. Educational digital games that deal with algorithmic sequences and structures can have a key role in the development of such ability in basic education, bringing fun and increasing students' involvement. However, many games developed with this goal are not always intuitive enough to facilitate the development of children's computational thinking. Building on previous work, this paper reports the evaluation of three children's games of this type, with the aim of consolidating a set of guidelines for interface and interaction design to help developers of these games.*

Resumo. *O pensamento computacional vem se tornando uma habilidade cada vez mais importante na sociedade contemporânea. Jogos digitais educacionais que trabalham sequências e estruturas algorítmicas podem ser aliados no desenvolvimento dessa habilidade no ensino básico, agregando ludicidade e aumentando o envolvimento dos alunos. Todavia, muitos jogos desenvolvidos com esse propósito nem sempre são intuitivos o suficiente a ponto de facilitar o desenvolvimento do pensamento computacional das crianças. Dando continuidade a trabalhos anteriores, este artigo relata a avaliação de três jogos infantis desse gênero com o objetivo de consolidar um conjunto de diretrizes de design de interação e interface para guiar os projetistas.*

1. Introdução

Diante dos desafios do mundo moderno, o pensamento computacional, composto de estratégias para resolver problemas aplicando competências e habilidades da Computação, vem se tornando algo importante para todos e não apenas para cientistas da computação. Sendo assim, sua implementação e integração no ensino básico têm sido discutidas ao longo dos anos [Wing 2006, 2008], sendo defendidas por importantes órgãos educacionais internacionais como o *CSTA K-12 Computer Science Standards* [CSTA 2011]. O pensamento computacional ajuda os alunos a lidarem com problemas complexos, conceituando-os e analisando-os através de seleção e aplicação de estratégias e ferramentas adequadas [Von Wangenheim et al. 2014].

Além disso, a apresentação de conceitos de Computação no ensino básico pode ajudar a fomentar o interesse de jovens aprendizes pelos cursos superiores da área de tecnologia e consequentemente a aumentar o número de profissionais em Computação [Melo, Costa e Batista 2013], visto que ainda existe um déficit na formação de mão de obra qualificada, devido à alta evasão existente em cursos da área de Tecnologia da

Informação [Prietch e Pazeto 2010].

No entanto, a inserção do pensamento computacional na educação básica brasileira ainda é muito pouco explorada. Em geral, as escolas que possuem aulas de informática preocupam-se mais em ensinar a manusear aplicativos de escritório, navegar na internet e fazer pesquisas, entre outras tarefas básicas. Isso faz com que conceitos da Computação como o pensamento crítico, a capacidade de tomar decisão e resolver problemas, entre outros, raramente sejam trabalhados nessas aulas [Melo, Costa e Batista 2013]. Além do mais, ainda não se tem um método regulamentado de se trabalhar o pensamento computacional e com isso, alternativas têm sido discutidas em fóruns da área e paralelamente surgem diversas iniciativas relacionadas à difusão do pensamento computacional ao público infanto-juvenil [Pontual Falcão, Gomes e Albuquerque 2015]. No campo internacional, sob o lema de que todos os alunos em todas as escolas devem ter a oportunidade de aprender ciência da computação, o Code.org vem estimulando o raciocínio lógico e a resolução de problemas por meio do ensino de programação. Além de incluir aulas gratuitas de codificação, a proposta visa estimular várias escolas a fazerem o mesmo [Code.org 2016].

No Code.org é possível perceber a popularidade que os jogos digitais que estimulam o pensamento computacional das crianças vêm ganhando. Pesquisas apontam que jogos digitais podem atuar de forma positiva no processo de ensino-aprendizagem oferecendo momentos lúdicos, interativos e motivadores, ao mesmo tempo em que apresentam desafios relacionados a conteúdos curriculares [Calixto Silva, Tedesco e Melo 2014].

Todavia, um fator que vem limitando o uso de jogos voltados ao estímulo do pensamento computacional é o desconhecimento dessas ferramentas por parte de educadores [Pontual Falcão, Gomes e Albuquerque 2015], como também a falta de um consenso no que diz respeito à avaliação de jogos educacionais e de como inseri-los nas práticas pedagógicas [Brito Junior e Aguiar 2014]. Os atuais modelos propostos de avaliação de jogos digitais educacionais ainda são genéricos e avaliam apenas critérios de usabilidade, funcionalidade e jogabilidade [Pontual Falcão, Gomes e Albuquerque 2015]. Assim, os educadores não possuem referenciais sólidos que possam ser usados para aferir a qualidade e orientar a adoção dessas novas tecnologias, e os projetistas tampouco possuem referenciais que os ajudem no desenvolvimento desse tipo de jogo.

O presente trabalho visa contribuir para a disseminação de jogos que envolvem pensamento computacional, fazendo uma análise de vários aspectos relativos à sua interface e à interação de crianças, por meio de avaliações empíricas. Dando continuidade a trabalhos anteriores, o objetivo é identificar elementos-chave que possam, futuramente, culminar em diretrizes para que projetistas produzam jogos que permitam uma melhor interação do público-alvo, e consequentemente facilitem o desenvolvimento do pensamento computacional. O presente artigo encontra-se dividido da seguinte forma: a próxima seção apresenta os trabalhos relacionados que embasaram esta pesquisa. Em seguida a metodologia utilizada nas avaliações empíricas. Por fim, os resultados encontrados seguidos das conclusões.

2. Trabalhos Relacionados

Pontual Falcão e Barbosa (2015) avaliaram o jogo *Lightbot*¹, que envolve pensamento computacional, com o intuito de analisar aspectos pedagógicos e técnicos da interação de crianças. Seguindo a proposta de Oliveira, Costa e Moreira (2001), foram realizadas avaliações formativa (com um grupo de crianças entre 6 e 9 anos) e objetiva (segundo heurísticas de usabilidade de software de Nielsen e Molich (1990)). Os autores identificaram problemas na interação com alguns elementos da interface, como a ineficácia da ajuda disponível em forma de tutorial, o não entendimento de elementos gráficos importantes para o controle do personagem e a falta de correspondência do contexto onde o personagem do jogo “vive” com o mundo real.

Gomes et al. (2015) também procuraram fazer uma avaliação do jogo *The Foos*² sob a ótica objetiva e formativa. A avaliação objetiva seguiu um conjunto de 10 critérios considerados importantes pelos autores que vão desde a facilidade de uso até design da tela e motivação. Na avaliação formativa, foram observadas as ações de crianças entre 5 e 7 anos diante do jogo e suas dificuldades. O principal obstáculo relatado foi acerca do sistema de ajuda do jogo não ter sido capaz de auxiliar as crianças em estágios mais avançados, evidenciando que a ajuda de um educador se faz necessária. Dando continuidade a esse trabalho com *The Foos*, estudos empíricos de Pontual Falcão, Gomes e Albuquerque (2015) com o mesmo público-alvo levaram à sugestão de aspectos-chave relativos à apresentação dos conceitos de pensamento computacional, formas de feedback e ajuda, forma de execução de comandos, e representação de comandos parametrizados.

Para que se gere um conjunto de diretrizes que consolide os resultados que vêm sendo disseminados, é preciso expandir os estudos empíricos, utilizando outros jogos com propósitos semelhantes. Neste artigo, investigamos de forma similar três outros jogos sobre pensamento computacional. Na seção de Discussão, aprofundamos o diálogo com os trabalhos relacionados.

3. Método

Três jogos educacionais infantis que envolvem pensamento computacional foram avaliados empiricamente com representantes do público-alvo. Os detalhes do método são descritos nesta seção.

3.1 Jogos Avaliados

Procurou-se escolher jogos com métodos de interação distintos (ou seja, os comandos eram inseridos clicando em botões ou arrastando blocos de funções específicas ou digitando linhas de código) e disponíveis gratuitamente na Internet. Foram escolhidos Operação Big Hero - Code Baymax³, Labirinto Clássico⁴ e CodeMonkey⁵.

3.1.1 Operação Big Hero – Code Baymax

Esse é um jogo da *Disney* que apresenta pequenos desafios a cada estágio, fazendo com

¹ lightbot.com

² thefoos.com

³ <http://jogos.disney.com.br/operacao-big-hero-code-baymax>

⁴ <https://studio.code.org/hoc/1>

⁵ <https://www.playcodemonkey.com/>

que o jogador use estratégias, por meio de lógica de programação, para vencê-los. O jogo é contextualizado em uma história na qual um personagem controla um robô para poder salvá-lo. Sendo assim, durante o andamento do jogo, são passadas orientações por meio de *pop-ups* que inserem o jogador na história.

Os desafios começam de maneira simples, com ações como: andar para frente, virar à esquerda, à direita ou pegar um objeto. Esses comandos de instruções fazem com que o robô se desloque em uma área quadriculada, como mostrado na Figura 1, a fim de chegar no quadrado verde e completar a etapa. À medida que etapas mais avançadas são alcançadas, novos comandos são inseridos no jogo, como também, a incorporação de até três ações do robô em um único comando sendo utilizado em etapas mais complexas. Os comandos são agrupados em um painel à direita e para executá-los é necessário clicar em um botão ‘play’. *Operação Big Hero – Code Baymax* está disponível *online* e pode ser jogado através de navegadores *web*.



Figura 1. Estágio de Operação Big Hero – CodeBaymax

3.1.2 Labirinto Clássico

Esse jogo se baseia na condução de diversos personagens através de labirintos, como exemplificado na Figura 2, para se chegar ao objetivo de cada estágio. Para isso é necessário o uso de estratégia e lógica de programação para se avançar no jogo. As ações dos personagens são definidas arrastando e soltando blocos com ações específicas para um painel de execução. De início os movimentos são simples, com poucas ações para se completar o estágio, como avançar, virar à esquerda ou virar à direita. A cada introdução de um novo bloco de ação, um vídeo explicativo é mostrado. Dessa forma, ações mais complexas como blocos de *loops* e blocos condicionais vão aumentando o nível do jogo, fazendo com que o jogador precise refletir bastante para gerar o movimento dos personagens. O jogo encontra-se disponível *online* e faz parte da Hora do Código, uma iniciativa da *Code.org*.

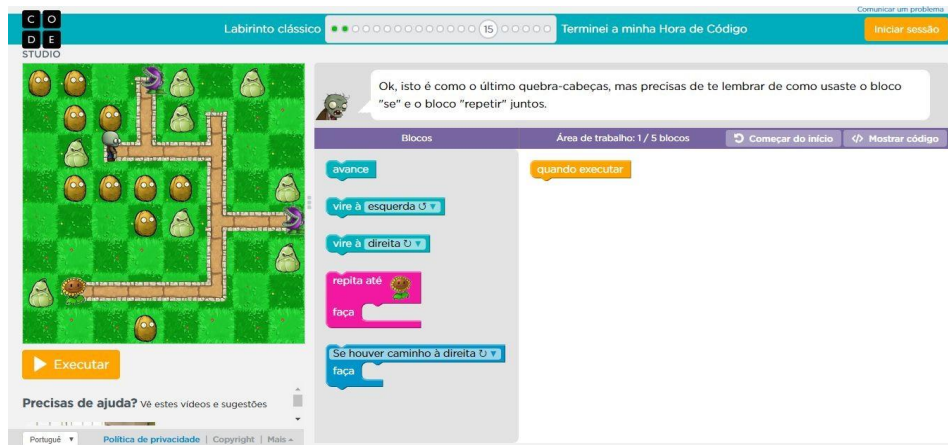


Figura 2. Labirinto Clássico, estágio 15

3.1.3 CodeMonkey

CodeMonkey é um jogo mais avançado no que diz respeito à difusão do pensamento computacional por apresentar uma linguagem de programação real, a *CoffeeScript*⁶ em seu contexto. O jogo traz diversos desafios, cujo objetivo de cada um é fazer um pequeno macaco, como mostrado na Figura 3, se deslocar e capturar as bananas da melhor forma possível. Nesse jogo a criança de fato escreve as linhas de código com as instruções ou utiliza os botões pré-definidos que insere algumas automaticamente, restando apenas completá-las com seus respectivos parâmetros.



Figura 3. CodeMonkey, desafio número 10

3.2 Procedimento

As avaliações se deram com um total de 7 crianças, recrutadas por meio de contatos pessoais do primeiro autor, sendo 3 meninos e 4 meninas com idades variando de 8 a 12 anos. Por meio de um contato pessoal, três crianças foram avaliadas em um clube escolar de fim de semana, e quatro outras, recrutadas por meio de outros contatos, fizeram os testes nas suas respectivas casas, em um momento a sós com o pesquisador.

As crianças escolhidas, apesar de terem experiência com computadores, nunca tinham

⁶ <http://coffeescript.org/>

jogado esse tipo de jogo sobre pensamento computacional, fator que contribuiu para a participação da pesquisa.

Nenhuma instrução preliminar acerca dos jogos foi dada, deixando as crianças explorarem-nos livremente. Como os jogos estão disponíveis online, isso permitiu que cada criança os testasse em navegadores *web* instalados em um *notebook*. Para os testes, o pesquisador (primeiro autor deste trabalho) conduziu estudos empíricos com os 3 jogos descritos na seção 3.1 (Operação Big Hero – Code Baymax, Labirinto Clássico e CodeMonkey), que deveriam ser jogados por cada participante por no máximo 30 minutos, para evitar sessões demasiadamente longas.

Os dados foram coletados por meio de observações qualitativas, ou seja, guiadas por aspectos de interesse para a pesquisa, porém sem uma preocupação em registrar dados numéricos e quantificáveis, mas sim anotações de campo sobre o comportamento dos jogadores, suas dúvidas e as dificuldades encontradas durante a interação com cada jogo. A cada dúvida relatada em qualquer estágio dos jogos, o pesquisador poderia ajudar os participantes no entendimento acerca daquela fase (mas sem fornecer a solução do problema) e também sobre aspectos gerais dos jogos. A necessidade de ajuda também foi um ponto analisado.

Os aspectos identificados foram analisados qualitativamente e são discutidos à luz das questões identificadas anteriormente em estudos com o jogo *The Foos* [Pontual Falcão, Gomes e Albuquerque 2015].

4. Resultados

As três principais categorias decorrentes da análise dos dados são apresentadas a seguir: motivação; entendimento da interface; e compreensão dos conceitos de lógica de programação.

4.1 Motivação

De forma geral, o comportamento das crianças ao jogar indicou motivação, o que reforça que utilizá-los para o ensino-aprendizagem dos conceitos de pensamento computacional é algo proveitoso. Nos três jogos avaliados, os estágios iniciais apresentam uma maior facilidade; isso serviu para instigar a curiosidade das crianças em querer ir além e vencer os desafios. O envolvimento emocional foi claro diante do entusiasmo apresentado em querer fazer com que os personagens dos jogos avançassem de fase. O jogo Operação Big Hero – Code Baymax foi o que mais envolveu as crianças, pois apresenta uma jogabilidade um pouco melhor, ou seja, uma facilidade maior de ser jogado, que fazia com que as crianças não pensassem que estavam trabalhando conceitos de pensamento computacional, mas sim tentando vencer um desafio divertido.

Em Labirinto Clássico as crianças se empolgaram com o personagem de um pássaro e comemoravam quando conseguiam fazê-lo chegar no final do labirinto e “matar” o personagem de um porco. Por outro lado, apesar de despertar uma motivação inicial, CodeMonkey foi o jogo em que as crianças mais sentiram dificuldades por terem que digitar linhas de código que muitas vezes eram feitas da forma errada e isso fez com que as desmotivassem, tendo como consequência não irem muito além nas fases do jogo.

4.2 Entendimento da Interface

Algumas explicações do pesquisador sobre elementos da interface foram necessárias quando as crianças não entendiam a funcionalidade dos componentes. Code Baymax e CodeMonkey apresentam formas muito similares para auxiliar e instruir o jogador, que eram caixas de diálogos (Figura 4) com informações e a imagem de um personagem. Essas instruções raramente foram levados em consideração pelas crianças, que fechavam rapidamente esses componentes.



Figura 4. Instruções por meio de caixas de diálogo em Baymax (à esquerda) e CodeMonkey (à direita)

Outra forma de ajuda do Code Baymax que funcionou melhor foi a utilização de uma mão como forma de ensinar como se deve proceder quando algo novo surge na interface. No entanto, essa ajuda não aparece para todos os elementos novos, trazendo alguns pontos de dificuldade. Por exemplo, quando os jogadores ao final da sequência de comandos inseridos, têm que colocar o comando para abrir uma porta, o jogo não apresenta nenhum *feedback* ou ajuda sobre a nova funcionalidade, o que causou uma certa confusão pelas crianças não estarem entendendo o porquê do estágio não terminar quando o robô chegava no ponto final.

Já em CodeMonkey, quando a sequência de linhas de comando é executada de forma errada por muitas vezes, o jogo como forma de ajuda mostra a solução quase pronta para ser usada, faltando apenas alguns parâmetros para completá-la. Mesmo assim, as crianças não conseguiam entender a sequência lógica daqueles comandos, o que resultou em pedidos de ajuda ao pesquisador. Uma dificuldade percebida também, foi na utilização dos botões de ações para controle do personagem. Virar à esquerda ou à direita, por exemplo, era uma tarefa que dependia da utilização de dois comandos em conjunto: “*turn*”, e “*left*” ou “*right*”, dependendo da direção pretendida. As crianças não conseguiram perceber essa utilização conjugada, o que as levava a utilizar apenas as setas para esquerda ou direita, o que não era correto pelas regras do jogo. As linhas de comando que muitas vezes precisavam de parâmetros para serem executadas não foram colocadas, em sua maioria, da maneira correta, gerando muitos erros e consequentemente frustrando os jogadores.

Em Labirinto Clássico, as crianças também apresentaram algumas dificuldades no entendimento da interface. O jogo mostra um tutorial inicial de como deve ser a forma de jogar, entretanto, o tutorial apresentado é um vídeo falado em inglês que apesar das legendas em português dificilmente conseguiu instruir as crianças causando o não entendimento acerca do que fazer. Dessa forma, essas instruções tiveram que ser dadas

às crianças por meio de ajuda do pesquisador. O mesmo problema aconteceu quando novos conceitos foram inseridos, como o de repetição (loop).

Uma outra forma de ajuda usada neste jogo é o *feedback* textual na parte superior da tela (Figura 5), exibida quando a criança não consegue passar pelo estágio por algum motivo. Essa forma de ajuda dificilmente foi percebida pelas crianças que muitas vezes acabavam pedindo ajuda ao pesquisador.

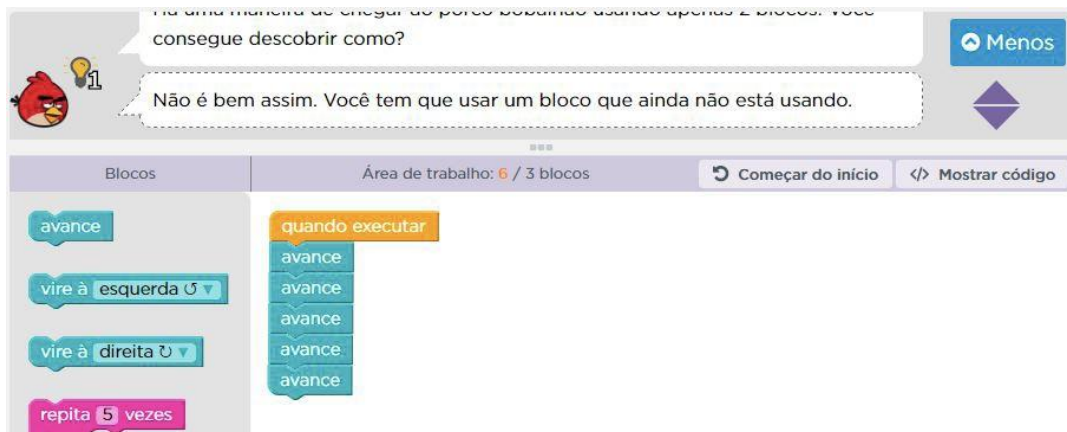


Figura 5. Feedback textual em Labirinto Clássico (no topo)

4.3 Conceitos de Lógica de Programação

Tentar passar conceitos de lógica de programação de forma lúdica e que não desmotive as crianças pelas dificuldades apresentadas é um grande desafio para os projetistas [Pontual Falcão, Gomes e Albuquerque 2015]. Os estágios/desafios iniciais são os mais simples, tratando de tarefas básicas como movimentar os personagens e seguir um caminho. Todas as crianças conseguiram passar os primeiros níveis, mas colocando comandos aleatórios, por tentativa e erro. Nesse caso percebeu-se que não conseguiram construir uma lógica adequada para resolução do problema e não construíram adequadamente os conceitos de pensamento computacional ao explorarem sozinhas o jogo em uma primeira interação.

As maiores dificuldades surgiram quando os conceitos de repetição começaram a ser apresentados, como já foi encontrado por outros pesquisadores [Gomes, Melo e Tedesco 2016]. Em Code Baymax a opção do comando de repetição muitas vezes não era utilizada, resultando no acúmulo de vários comandos simples no painel de execução, impossibilitando assim a conclusão do estágio sem sucesso, por não haver mais espaço livre para adição de mais comandos. Outras vezes o comando de repetição era utilizado, mas de forma aleatória e sem o entendimento real para que servia. No jogo Labirinto Clássico a grande dificuldade também foi com o bloco de repetição: nenhuma das crianças conseguiu ir adiante por não entender a sua funcionalidade que necessitava de parâmetros para o seu correto funcionamento. Os comandos de repetição dos dois jogos são exibidos na Figura 6.



Figura 6. Os comandos de repetição (indicados pelas setas vermelhas) no Baymax (à esquerda) e Labirinto Clássico (à direita)

Com o CodeMonkey, por ser um jogo que apresentou maior dificuldade, nenhuma criança chegou aos desafios que envolvessem comandos de repetição no tempo da sessão. Elas pediram bastante ajuda, pois o jogo é mais complexo e não apresentou-se intuitivo para elas. Nos desafios testados, o que foi trabalhado pelas crianças foi a programação procedural, colocando-se linhas de comandos em sequência para fazer com que o macaco se movimentasse na interface a fim de conseguir bananas.

5 Discussão

Os resultados das avaliações empíricas ratificam propostas anteriores, como os aspectos-chave sugeridos por Pontual Falcão, Gomes e Albuquerque (2015) e algumas heurísticas de Nielsen e Molich (1990), adaptadas para um contexto de *software* educacional, apresentadas no trabalho de Pontual Falcão e Barbosa (2015). Os principais pontos identificados são discutidos nesta seção.

As formas de ajuda e *feedback* fornecidas pelos jogos confirmaram-se como aspectos-chave para a interação bem-sucedida das crianças com os jogos, como colocado por Pontual Falcão e Barbosa (2015). A ajuda é uma peça fundamental em momentos de dificuldade para que o jogador consiga sanar possíveis dúvidas sem ajuda externa, embora jogos desse tipo devam ser mais intuitivos para que a ajuda não fosse acionada. Percebeu-se mais uma vez a falta de paciência das crianças para ler tutoriais ou assistir a vídeos demonstrativos, pois buscam interagir com o jogo de forma mais imediata. Nos jogos analisados neste trabalho, caixas de diálogos e vídeos informativos foram prontamente deixados de lado e não levados em consideração pelos jogadores. Sendo assim, uma forma de ajuda mais eficaz precisa ser investigada para guiar as crianças no processo de interação sem comprometer a construção dos conceitos de pensamento computacional. A utilização de demonstrações (como a mão de Baymax) pode ser um bom caminho, embora já questionado por Pontual Falcão, Gomes e Albuquerque (2015).

Relacionado a esse tema, o controle e liberdade do usuário, principalmente por meio das opções de desfazer e refazer, colocado por Pontual Falcão e Barbosa (2015) como fundamental, foi percebido também nos nossos estudos. Em todos os jogos analisados é possível que as sequências algorítmicas criadas pelas crianças sejam

desfeitas e refeitas, caso elas não obtivessem êxito no estágio. Esse recurso foi bastante utilizado, permitindo que as crianças construam a sua lógica e interprete os resultados.

Em relação aos conceitos de pensamento computacional, Pontual Falcão, Gomes e Albuquerque (2015) afirmam que eles devem ser embutidos como condições implícitas para vencer os desafios, o que foi confirmado no presente trabalho. Para manter a motivação das crianças, deve-se trabalhar o pensamento computacional de forma divertida e lúdica. O jogo CodeMonkey, o único avaliado em que os jogadores tinham que inserir linhas de comando, foi o que causou mais dificuldade e desmotivação, enquanto que Code Baymax demonstrou ter uma melhor jogabilidade pelo modelo de interação escolhido para o jogo. É importante lembrar que essa discussão se situa no contexto de crianças iniciantes em programação, e que jogos como CodeMonkey podem ser usados após as crianças se familiarizarem com alguns conceitos.

Neste sentido, Pontual Falcão, Gomes e Albuquerque (2015) destacam a importância, em termos educacionais, de somente introduzir comandos parametrizados após a familiarização com os comandos básicos. De fato, no jogo CodeMonkey não houve um bom entendimento acerca de comandos parametrizados: as crianças inseriam parâmetros aleatórios. Essa dificuldade também foi apresentada em blocos de repetição presentes no jogo Labirinto Clássico, que exigia parâmetros.

Por fim, destacamos a importância de comandos simples e objetivos. Em CodeMonkey foi percebido que a necessidade de utilizar botões em conjunto para mover o personagem para esquerda ou direita não foi entendida. As crianças utilizavam apenas o botão que representava o movimento direcional (esquerda ou direita), quando deveriam utilizar em conjunto com o botão 'vire'. Dificuldades com a forma de representar movimentos de rotação também foram anteriormente reportadas por Raabe et al. (2015) com o brinquedo de programar Beebot.

6. Conclusões e Trabalhos Futuros

O pensamento computacional deve ser levado em consideração nos dias atuais, tendo em vista as demandas da sociedade contemporânea, principalmente por profissionais de Tecnologia da Informação. Apresentar ferramentas para que essa habilidade seja desenvolvida em crianças, a partir do ensino básico, é importante para aprimorar o raciocínio lógico desde cedo.

Os jogos digitais educacionais que trabalham sequências algorítmicas, como mostrado neste trabalho, podem atuar de forma positiva, deixando o processo de ensino-aprendizagem mais divertido. No entanto, muitas vezes por não levarem em conta o envolvimento do público-alvo durante a avaliação dos jogos em desenvolvimento, os projetistas cometem falhas de projeto de interação e interface que por sua vez não favorecem a construção dos conceitos apresentados. A necessidade de se ter referências para desenvolver e avaliar esses jogos tem levado a algumas iniciativas de pesquisadores no Brasil, incluindo o trabalho apresentado neste artigo. Estudos empíricos com três jogos que envolvem pensamento computacional indicaram questões-chave relativas a motivação, entendimento da interface e construção dos conceitos. Os resultados vêm consolidar e enriquecer propostas anteriores [Pontual Falcão, Gomes e Albuquerque 2015; Pontual Falcão e Barbosa, 2015], e somado a elas contribuem para a futura geração de diretrizes que podem deixar os jogos que trabalham o pensamento computacional mais

intuitivos perante o público infantil.

Outros trabalhos futuros incluem a verificação sistemática do quanto esses jogos facilitam o aprendizado dos conceitos de lógica de programação e se há realmente maior construção de conteúdo ao se utilizar os mesmos. Também é preciso verificar meios de proporcionar ajuda mais eficazes, de forma a sanar possíveis dúvidas que venham causar desestímulo e desistência do jogo.

Agradecimentos

Agradecemos as crianças participantes, pelo seu tempo e disponibilidade, e a Igor Lucena, por sua ajuda nos estudos empíricos.

Referências

- Brito Junior, O. O., e Aguiar, Y. P. C. (2014) Análise de abordagens objetivas para avaliação de softwares educativos. Em Anais do XXIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais – IHC'14, Foz do Iguaçu-PR, Brasil.
- Calixto Silva, T. S., Tedesco, P. C. A. R., e Melo, J. C. B. (2014) A importância da motivação dos estudantes e o uso de técnicas de engajamento para apoiar a escolha de jogos no ensino de programação. Em Anais do XXV Simpósio Brasileiro de Informática na Educação - SBIE 2014, Dourados-MS, Brasil.
- CODE.ORG. (2016) “Todos os alunos em todas as escolas, devem ter a oportunidade de aprender ciência da computação”, <https://code.org/>, Setembro.
- CSTA (2011) “CSTA K–12 computer science standards”. http://csta.acm.org/curriculum/sub/currfiles/csta_k-12_css.pdf.
- Gomes, T. C. S., Barreto, P. P., Albuquerque, I. R. L., e Pontual Falcão, T. (2015) Avaliação de um Jogo Educativo para o Desenvolvimento do Pensamento Computacional na Educação Infantil. Em Anais do XXVI Simpósio Brasileiro de Informática na Educação - SBIE 2015, Maceió-AL, Brasil.
- Gomes, T.C.S., Melo, J. C. B. de, e Tedesco, P. C. A. R. (2016) Jogos Digitais no Ensino de Conceitos de Programação para Crianças. Em Anais do XXVII Simpósio Brasileiro de Informática na Educação - SBIE 2016, Uberlândia-MG, Brasil.
- Melo, L. A., Costa, T. K. L. e Batista, A. C. D. (2013) Pense bem: proposta e desenvolvimento de jogo digital para ensino de computação na educação básica. Em Anais do XXIV Simpósio Brasileiro de Informática na Educação - SBIE 2013.
- Nielsen, J., Molich, R. (1990) Heuristic evaluation of user interfaces. In Proceedings of ACM CHI'90 Conference, Seattle, EUA.
- Oliveira, C. C. de, Costa, J. W. da, e Moreira, M. (2001) Ambientes informatizados de aprendizagem: produção e avaliação de software educativo, Série Prática Pedagógica, Papirus.
- Pontual Falcão, T. e Barbosa, R. S. (2015) "Aperta o Play!" Análise da Interação Exploratória em um Jogo Baseado em Pensamento Computacional. Em Anais do XXVI Simpósio Brasileiro de Informática na Educação - SBIE 2015, Maceió-AL, Brasil.

- Pontual Falcão, T., Gomes, T. C. S. e Albuquerque, I. R. (2015) O Pensamento Computacional Através dos Jogos infantis: uma Análise dos Elementos de Interação. Em Anais do XIV Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais – IHC’15, Salvador, Brasil.
- Prietch, S. S. e Pazeto, T. A. (2010) Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias. Em Anais da X Escola Regional de Informática Bahia-Alagoas-Sergipe - Workshop de Educação em Informática Bahia-Alagoas-Sergipe - ERBASE 2010.
- Raabe, A., Rodrigues, A. J., Santana, A. M., Vieira, M. V., Rosário, T., Carneiro, A. C. R. (2015) Brinquedos de Programar na Educação Infantil: um Estudo de Caso. Em Anais do XXI Workshop de Informática na Escola - WIE 2015, Maceió-AL, Brasil.
- Von Wangenheim, C. G., Nunes, V. R., Santos, G. D. dos, Alves, N. C., Coan, E. S. e Mansur, C. (2014) “Resumo de Objetivos de Aprendizagem de Computação no Ensino Fundamental (Currículo de Referência CSTA/ACM K-12)”. Relatório Técnico, INCoD - Instituto Nacional para Convergência Digital, INE - Departamento de Informática e Estatística - INE, Universidade Federal de Santa Catarina.
- Wing, J. M. (2006) Computational Thinking, In Communications of the ACM, 49(3), pages 33-35.
- Wing, J. M. (2008) Computational thinking and thinking about computing, In Philosophical transactions of the royal society of London: mathematical, physical and engineering sciences, 366, pages 3717-3725.