

Investigando Dificuldades em Recursos do Code.org: Aplicação do Método do Percurso Cognitivo ao Tutorial Artista da Hora do Código

Andrea Schwertner Charão¹, Francisco Ritter²

¹ Departamento de Linguagens e Sistemas de Computação
Programa de Pós-Graduação em Ciência da Computação

² Departamento de Psicologia
Universidade Federal de Santa Maria
Santa Maria, RS

andrea@inf.ufsm.br, franciscoritter@gmail.com

Abstract. *Tutorial games on the Code.org platform have become popular as large-scale activities to promote Computational Thinking and to develop programming skills. Previous research works evaluate aspects of these games based on data collected from users. In this work, we propose another approach, exploring a method for usability evaluation – the cognitive walkthrough. This method proposes a detailed inspection of the user interface, conducted by specialists. We applied the method to the tutorial “Artist” from the Hour of Code, on the Code.org platform, and identified points where the game presents cognitive obstacles to users. Such result can be useful to all initiatives that rely on Code.org to develop Computational Thinking.*

Resumo. *Jogos tutoriais da plataforma Code.org têm se popularizado como atividades em larga escala para promover o Pensamento Computacional e desenvolver habilidades de programação. Há pesquisas que avaliam aspectos desses jogos a partir de dados coletados junto a usuários. Neste trabalho, propomos outra abordagem, explorando um método de avaliação de usabilidade – o percurso cognitivo (cognitive walkthrough). Tal método propõe uma inspeção detalhada da interface com o usuário, conduzida por especialistas. Aplicamos o método ao tutorial “Artista” da Hora do Código, na plataforma Code.org, e identificamos pontos em que o jogo apresenta obstáculos cognitivos aos usuários. Tal resultado pode ser útil a iniciativas que se apoiam no Code.org para desenvolver o Pensamento Computacional.*

1. Introdução

Existe, atualmente, uma grande diversidade de recursos que se propõem a desenvolver o Pensamento Computacional e habilidades de programação desde cedo, junto a crianças e jovens. Na plataforma Code.org, há vários jogos tutoriais que oferecem uma iniciação a esta temática. Tais recursos se caracterizam por um formato favorável para aplicação em larga escala, com lições de curta duração (cerca de uma hora) e em ordem crescente de dificuldade. Tutoriais com diferentes temas também são inseridos em cursos de maior

duração (cerca de 20h). Os recursos^{1,2} são classificados de acordo com a faixa etária ou escolaridade dos usuários.

Dada sua popularidade, a plataforma Code.org tem inspirado trabalhos de pesquisa com diferentes objetivos e abordagens. Jogos como “Labirinto Clássico” ou “Hour of Code Minecraft”, por exemplo, já foram alvo de análises detalhadas [Martins et al. 2016, GhasemAghaei et al. 2017]. No geral, as pesquisas têm confirmado a efetividade de jogos tutoriais do Code.org no desenvolvimento do Pensamento Computacional, embora identifiquem lacunas e questões ainda a serem exploradas.

Uma abordagem de avaliação de jogos educacionais considera aspectos de usabilidade desta categoria de software. Dentre os diversos métodos disponíveis para isso, Yáñez Gómez et al. [2017] identificaram que grande parte dos trabalhos realiza avaliações com usuários finais, embora inspeções conduzidas por especialistas também tenham potencial para revelar problemas de usabilidade, que se traduzem em obstáculos ao usuário.

Seguindo este caminho pouco explorado, neste trabalho apresentamos uma aplicação do método do Percurso Cognitivo [Polson et al. 1992] ao jogo tutorial “Artista” do Code.org, que, ao nosso conhecimento, ainda não foi alvo deste tipo de análise. Ao desenvolver este trabalho, lançamos um novo olhar sobre o jogo Artista, buscando uma forma de antecipar e explicar possíveis dificuldades que seus usuários possam enfrentar.

O restante deste artigo está organizado como segue: na seção 2, apresentamos uma fundamentação para o trabalho e discutimos alguns trabalhos relacionados; na seção 3, detalhamos as fases e procedimentos do Percurso Cognitivo realizado; na seção 4, apresentamos e discutimos os resultados e, na seção 5, formulamos as considerações finais sobre o trabalho.

2. Fundamentação e Trabalhos Relacionados

Nesta seção, primeiramente apresentamos uma fundamentação sobre o método do Percurso Cognitivo e suas relações com teorias de aprendizagem. Em seguida, discutimos outros trabalhos de pesquisa cujo foco se estabeleceu sobre um ou mais jogos do Code.org e, logo depois, apresentamos o jogo tutorial Artista, que é objeto deste trabalho.

2.1. Método do Percurso Cognitivo

O método do Percurso Cognitivo (*Cognitive Walkthrough* – CW) foi proposto por Polson et al. [1992] para inspecionar a usabilidade de interfaces com usuário. O método é aplicado por especialistas, considerando a perspectiva de um usuário sem contato prévio com o produto. O CW é um procedimento precisamente definido, simulando processos cognitivos de um usuário que interage com uma interface, a fim de realizar tarefas específicas para atingir objetivos.

Pode-se dizer que o CW não é um método amplamente utilizado para avaliar jogos com propósito educacional, embora existam trabalhos nesse sentido [Farrell e Moffat 2014, GhasemAghaei et al. 2017, Yáñez Gómez et al. 2017]. Qualquer que seja o propósito de um jogo, deve-se sempre avaliar se seu conteúdo e estrutura estão bem integrados, a ponto

¹<https://hourofcode.com/us/learn>

²<https://studio.code.org/courses>

que seja impossível distingui-los. Caso contrário, a discrepância entre estas dimensões fica logo evidente para os usuários.

Alguns pontos que devem ser considerados para se atingir esta condição ideal são a clareza da interface e também a clareza com relação ao desafio posto. Um problema mal definido pode comprometer toda a experiência do jogo, por melhor que seja sua interface. Além disso, é necessário que haja no jogo o que Jean Piaget chamou de “acomodação”, ou seja, a ampliação do entendimento através da construção de novos esquemas cognitivos. Nem todos os desafios vão poder ser resolvidos com os esquemas que o usuário já possui, e novas formas de pensar serão necessárias para vencer as etapas mais difíceis [Piaget 1972].

Do ponto de vista cognitivo, também é importante considerar que errar faz parte processo de aprendizado e do desafio de jogar. Sem isto, não há crescimento. A distância entre o que o usuário sabe ao iniciar o jogo e aquilo que vai lhe ser necessário para vencer os desafios e chegar vitorioso ao final é o que o psicólogo Lev Vigotsky chamava “zona do desenvolvimento proximal”, considerado o motor da aprendizagem [Bruner 1984]. Todos esses pressupostos são considerados, em maior ou menor grau, no embasamento teórico do método do Percurso Cognitivo.

2.2. Jogos do Code.org como Objeto de Pesquisa

A plataforma Code.org se desenvolveu na última década, tornando-se uma referência na popularização da educação em Computação [Wilson 2013]. Seus recursos, inicialmente direcionados ao evento *Hour of Code* (Hora do Código), foram sendo organizados para inserção em percursos mais longos de aprendizagem. Ao longo dessa evolução, muitos pesquisadores se interessaram em usar e examinar os recursos do Code.org, sob diferentes pontos de vista.

De forma geral, vários autores apontam aspectos positivos dos jogos tutoriais do Code.org, tanto do ponto de vista motivacional como de aprendizagem [Kalelioğlu 2015, Raabe et al. 2015, Cavalcante et al. 2016, Papat e Starkey 2019]. Também há autores que, em suas experiências com jogos do Code.org, apontam problemas relacionados ao ganho no aprendizado [Du et al. 2016] ou a dificuldades que causam impacto negativo em certos públicos [Lambić et al. 2020].

A maioria das pesquisas com foco nos recursos do Code.org se apoia em dados coletados junto a usuários, após a conclusão de atividades na plataforma. Trabalhos envolvendo inspeções e análises realizadas por especialistas são menos frequentes, mas também trazem observações relevantes [GhasemAghaei et al. 2017, Freitas e Morais 2019], com profundidade e detalhamento que não se costuma atingir em investigações com usuários finais. Nosso trabalho se soma a estes, aplicando um método pouco explorado neste contexto e analisando um jogo que também não tem recebido muita atenção de pesquisadores.

2.3. Tutorial “Artista” do Code.org

Na plataforma Code.org, a denominação “Artista” é usada em diferentes jogos tutoriais com um tema em comum: a criação de figuras e desenhos por meio de programação. A programação utiliza uma linguagem visual, com blocos que se encaixam como num quebra-cabeças, seguindo a mesma abordagem utilizada em vários outros tutoriais do Code.org. Os jogos neste tema estão distribuídos em planos de ensino para cursos de 20h e também em uma versão condensada para o evento “Hora do Código”. Ao todo, contabilizamos 23

jogos no Code.org com este tema. Na tabela 1, apresenta-se a localização destes jogos na plataforma (com nome do curso e posição da atividade entre as demais), os conceitos abordados e o número de desafios de cada tutorial.

Tabela 1. Versões do jogo tutorial Artista na plataforma Code.org

| Localização | Seção da plataforma | Conceitos | Desafios | Posição no curso |
|---|---------------------|--|----------|------------------|
| http://studio.code.org/s/course1/stage/8/puzzle/1 | Curso 1 | Sequência | 12 | 8/18 |
| http://studio.code.org/s/course1/stage/10/puzzle/1 | Curso 1 | Sequência / Formas geométricas | 10 | 10/18 |
| http://studio.code.org/s/course1/stage/18/puzzle/1 | Curso 1 | Laços | 10 | 18/18 |
| http://studio.code.org/s/course2/stage/4/puzzle/1 | Curso 2 | Sequência | 12 | 4/19 |
| http://studio.code.org/s/course2/stage/7/puzzle/1 | Curso 2 | Laços | 16 | 7/19 |
| http://studio.code.org/s/course2/stage/11/puzzle/1 | Curso 2 | Depuração | 12 | 11/19 |
| http://studio.code.org/s/course2/stage/19/puzzle/1 | Curso 2 | Laços aninhados | 13 | 19/19 |
| http://studio.code.org/s/course3/stage/3/puzzle/1 | Curso 3 | Laços | 15 | 3/21 |
| http://studio.code.org/s/course3/stage/5/puzzle/1 | Curso 3 | Funções | 10 | 5/21 |
| http://studio.code.org/s/course3/stage/11/puzzle/1 | Curso 3 | Laços aninhados | 12 | 11/21 |
| http://studio.code.org/s/course3/stage/21/puzzle/1 | Curso 3 | Laços aninhados / Padrões geométricos | 15 | 21/21 |
| http://studio.code.org/s/course4/stage/3/puzzle/1 | Curso 4 | Laços aninhados | 14 | 3/22 |
| http://studio.code.org/s/course4/stage/6/puzzle/1 | Curso 4 | Variáveis | 16 | 6/22 |
| http://studio.code.org/s/course4/stage/10/puzzle/1 | Curso 4 | Laços for | 12 | 10/22 |
| http://studio.code.org/s/course4/stage/12/puzzle/1 | Curso 4 | Funções | 13 | 12/22 |
| http://studio.code.org/s/course4/stage/14/puzzle/1 | Curso 4 | Funções com parâmetros | 17 | 14/22 |
| http://studio.code.org/s/course4/stage/18/puzzle/1 | Curso 4 | Números binários | 11 | 18/22 |
| http://studio.code.org/s/20-hour/stage/5/puzzle/1 | Intr. acelerada | Laços | 10 | 5/20 |
| http://studio.code.org/s/20-hour/stage/7/puzzle/1 | Intr. acelerada | Laços aninhados | 11 | 7/20 |
| http://studio.code.org/s/20-hour/stage/11/puzzle/1 | Intr. acelerada | Funções | 11 | 11/20 |
| http://studio.code.org/s/20-hour/stage/15/puzzle/1 | Intr. acelerada | Funções com parâmetros | 10 | 15/20 |
| http://studio.code.org/s/20-hour/stage/19/puzzle/1 | Intr. acelerada | Laços for / Funções | 6 | 19/20 |
| https://studio.code.org/s/artist/stage/1/puzzle/1 | Hora do Código | Sequência / Laços / Funções | 10 | - |

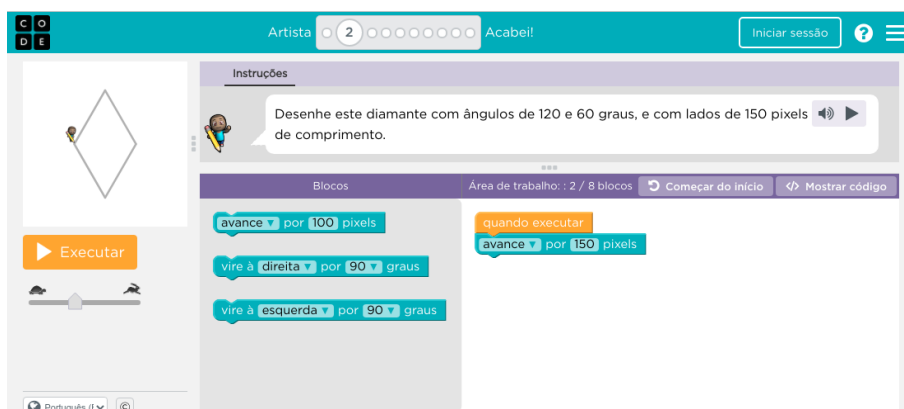
No presente trabalho, decidimos focar nossa experiência no jogo Artista da Hora do Código³, recomendado, segundo a plataforma, para um público de “4 a 104 anos”. Nesta versão do Artista, há 10 desafios propostos ao usuário, que deve usar blocos de sequência, repetição ou funções para atingir cada objetivo. A interface desta versão do Artista é apresentada na figura 1, onde pode ser vista a tela inicial e a solução do segundo desafio.

Apesar da recomendação ampla quanto à faixa etária, nota-se que o Artista não está entre as primeiras atividades sugeridas para a Hora do Código. Assume-se, então, que o usuário passe antes por outras atividades mais introdutórias. Vale lembrar que a Hora do Código foca em atividades de curta duração e sem uma sequência clara, em contraste com os cursos de maior duração, que têm uma sequência de atividades bem definida. Assim, escolhemos o tutorial Artista da Hora do Código justamente por ser uma atividade sem pré-requisitos explícitos, o que pode gerar obstáculos imprevisíveis aos usuários.

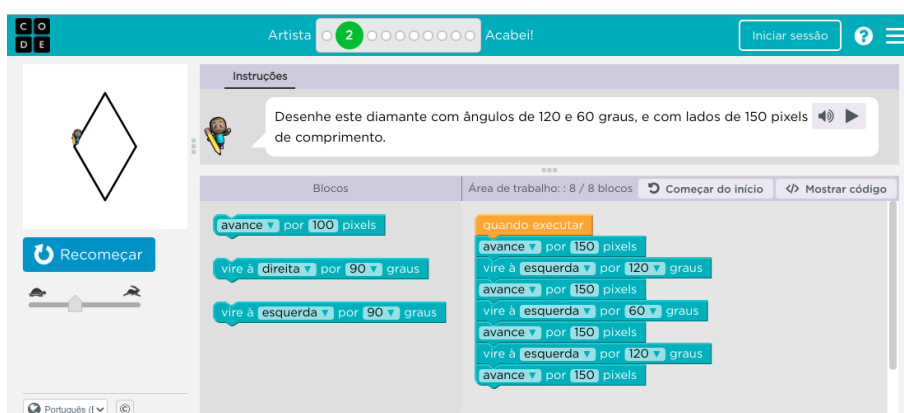
3. Realização do Percorso Cognitivo

Conforme já mencionado, o CW é uma inspeção conduzida por especialistas, sem a participação de usuários finais. Neste trabalho, o CW foi conduzido pelos autores deste

³<https://code.org/hourofcode/artist>



(a)



(b)

Figura 1. Interface do jogo tutorial Artista do Code.org: (a) tela inicial do segundo desafio e (b) solução do segundo desafio.

artigo, que possuem formações distintas, nas áreas de Computação e Psicologia. Os procedimentos adotados foram definidos em consenso entre os especialistas, passando pelas fases de preparação e avaliação. Em ambas as fases, formulários com campos e questões específicas guiaram a aplicação do método. As questões refletem um modelo cognitivo que fundamenta o método, utilizando uma terminologia padronizada.

3.1. Fase de Preparação

Na fase de preparação, os analistas devem escolher tarefas representativas, descrever tarefas, determinar sequência correta de ações, prever população de usuários e descrever objetivos iniciais dos usuários. Segundo a terminologia proposta pelo método, esta fase envolve os conceitos de objetivos, tarefas e ações. Um objetivo (*goal*) é algo que o usuário quer alcançar. Uma tarefa (*task*) é uma atividade que o usuário pode realizar. Cada tarefa pode ser composta por sub-tarefas básicas/elementares. Uma ação é uma atividade concreta que o usuário pode desempenhar (clique, seleção, etc.). Objetivos de mais alto nível se assemelham a descrições de tarefas, enquanto os de mais baixo nível descrevem ações.

Para prepararmos a aplicação do método ao jogo Artista, examinamos possibilidades na definição de objetivos, tarefas e ações. Neste ponto, percebemos que poderíamos tomar diferentes decisões dependendo da descrição dos **objetivos** do jogo. Para esta

aplicação, decidimos considerar que o formato de jogo tutorial traz naturalmente consigo uma apresentação de objetivos ao usuário: por exemplo, no primeiro desafio, o objetivo é desenhar um quadrado. A partir disso, desmembramos cada desafio em um conjunto de tarefas que foram descritas e associadas a sequências corretas de ações.

A tabela 2 apresenta um exemplo de detalhamento de ações para o segundo desafio, cujo objetivo é programar o personagem para desenhar um “diamante” (losango). Pode-se notar que as ações se relacionam à compreensão do problema proposto e aos recursos (blocos) disponíveis ao usuário para construção de um algoritmo capaz de resolvê-lo. O detalhamento de ações foi feito para todos os 10 desafios que compõem esta versão do Artista. Vale ressaltar que alguns desafios exigem mais ou menos ações para alcance do objetivo.

Tabela 2. Sequência de ações para o segundo desafio do jogo tutorial Artista da Hora do Código

| Ações | Objetivo do desafio 2: Desenhar um losango com blocos em sequência |
|-------|--|
| 1 | Ler ou ouvir as instruções: “Desenhe este diamante com ângulos de 120 e 60 graus, e com lados de 150 pixels de comprimento.” |
| 2 | Identificar a direção inicial do personagem |
| 3 | Trocar número de pixels no bloco “avance” para 150 |
| 4 | Identificar necessidade de usar bloco “vire à esquerda” |
| 5 | Identificar ângulo nos blocos “vire à esquerda” |
| 6 | Trocar ângulo para 60 ou 120 nos blocos “vire à esquerda” |
| 7 | Encadear blocos “avance” e “vire à esquerda” |
| 8 | Clicar em “Executar” |

Quanto à **população de usuários**, conforme apresentação do jogo na plataforma, assumiu-se que seriam pessoas de diferentes faixas etárias e níveis de escolaridade, já alfabetizadas, e com pelo menos uma experiência prévia de programação com blocos de sequência e repetição, como por exemplo os tutoriais “Labirinto Clássico” ou “Festa Dançante” da plataforma Code.org.

3.2. Fase de Avaliação

Na fase de avaliação, analisamos em profundidade o processo de interação de um usuário com a interface do jogo, respondendo a questões padronizadas e detalhadas para cada ação, conforme modelos disponíveis no artigo original de [Polson et al. 1992]. O modelo divide as questões em 3 seções: (1) questões sobre a relação entre os objetivos que a interface propõe e os objetivos que os usuários possam ter; (2) questões sobre possíveis problemas na realização de uma ação, dado o objetivo apropriado e (3) questões sobre a resposta da interface a ações corretas e possíveis mudanças nos objetivos do usuário.

Considerando o formato tutorial do jogo, entendemos que a primeira e terceira seções do questionário não trariam resultados úteis, pois os objetivos estão claramente apresentados para cada desafio, com instruções em texto e em áudio. Além disso, o usuário pode desfazer ações incorretas e recomeçar o desafio a qualquer tempo. A interface também oferece diferentes formas de *feedback* indicando erro ou sucesso na conclusão do desafio. Assim, na aplicação do método aos 10 desafios do Artista, respondemos somente

às 8 questões propostas por Polson et al. para identificar possíveis problemas na realização das ações. A seguir, reproduzimos estas questões, mantendo em inglês um rótulo proposto pelos autores e traduzindo o texto de cada questão:

1. *Availability*: Está óbvio que a ação correta é uma escolha possível aqui? Se não estiver, que porcentagem de usuários pode errar a ação? (% 0 25 50 75 100)
2. *Label*: Que rótulo está associado à ação correta?
3. *Link of label to action*: Se houver um rótulo associado à ação correta, ele é óbvio e está claramente vinculado à ação? Caso contrário, qual porcentagem de usuários pode ter problemas? (% 0 25 50 75 100)
4. *Link of label to goal*: Se existir um rótulo associado à ação correta, ele está obviamente conectado ao objetivo? Como? Caso contrário, qual porcentagem de usuários pode ter problemas? (% 0 25 50 75 100)
5. *No label*: Se não houver um rótulo associado à ação correta, como os usuários irão relacionar essa ação ao objetivo? Que porcentagem de usuários pode ter problemas para fazer isso? (% 0 25 50 75 100)
6. *Wrong choices*: Existem outras ações que possam parecer apropriadas para alcançar o objetivo? Em caso afirmativo, quais são elas e qual porcentagem de usuários pode fazer a escolha errada? (% 0 25 50 75 100)
7. *Time-out*: Se houver um tempo limite na interface para esta ação, este será suficiente para o usuário selecionar a ação correta? Que porcentagem de usuários pode ter problemas? (% 0 25 50 75 100)
8. *Hard to do*: Existe algo fisicamente complicado na execução da ação? Em caso afirmativo, que porcentagem de usuários pode ter problemas? (% 0 25 50 75 100)

As questões foram respondidas para cada uma das ações enumeradas para cada um dos 10 desafios do Artista, gerando uma matriz que encontra-se disponível em: <https://bit.ly/sbie2020-cw-artista>. Pode-se notar que as questões induzem uma análise objetiva de aspectos que podem oferecer obstáculos aos usuários, traduzida em estimativas sobre a população-alvo. As porcentagens foram atribuídas considerando a população descrita na fase de preparação e a experiência dos especialistas, com um cuidado para garantir coerência entre as diferentes ações e questões.

4. Resultados e Discussão

A aplicação do CW permitiu identificar pontos, em todos os desafios do Artista, em que existem prováveis descompassos entre a ação correta e a ação que o usuário pode realizar. Na tabela 3, listamos, para cada um dos 10 desafios deste jogo, as ações com maior estimativa de dificuldade, atribuída pelos especialistas durante a inspeção detalhada.

Além de ressaltar dificuldades por desafio, a matriz gerada com a aplicação do CW coloca em evidência obstáculos que estão presentes em mais de um desafio. Por exemplo, em todos os desafios, a interface do jogo menciona “ângulos”, um conceito matemático que não fazia parte do conhecimento prévio estabelecido para a população de usuários. Embora seja possível que usuários sem esse conhecimento completem os desafios, esse descompasso pode levar outros a abandonar o jogo. Neste sentido, entendemos que a recomendação do Code.org para público de “4 a 104 anos” esteja equivocada, pois estimamos que crianças entre 4 e 8 anos teriam significativa dificuldade em completar os desafios. Considerando a Base Nacional Comum Curricular brasileira, estimamos que o jogo só seja recomendável **a partir do 4º ano do Ensino Fundamental**.

Tabela 3. Ações com maior estimativa de dificuldade em cada desafio, identificadas com o CW no jogo Artista da Hora do Código

| Desafio | Objetivo / Ação com maior estimativa de dificuldade |
|---------|--|
| 1 | <i>Desenhar um quadrado com blocos em sequência / Encadear blocos “avance” e “vire”</i> |
| 2 | <i>Desenhar um losango com blocos em sequência / Trocar ângulo para 60 ou 120 nos blocos “vire”</i> |
| 3 | <i>Desenhar um hexágono com blocos em sequência / Encadear blocos “avance” e “vire”</i> |
| 4 | <i>Desenhar um hexágono com bloco de repetição / Encaixar os blocos “avance” e “vire” existentes dentro do bloco “repita”</i> |
| 5 | <i>Desenhar uma flor com repetição de losangos / Encaixar grupo de blocos dentro do bloco “repita”</i> |
| 6 | <i>Desenhar um hexágono com blocos de repetição aninhados / Encaixar bloco “vire à direita” após o bloco “repita” aninhado</i> |
| 7 | <i>Desenhar padrão com 6 hexágonos usando blocos de repetição aninhados / Encaixar bloco “vire à direita” após o bloco “repita” aninhado</i> |
| 8 | <i>Desenhar flores usando uma função / Encaixar o bloco “desenhe uma flor” dentro do bloco “repita”</i> |
| 9 | <i>Desenhar padrões com formas geométricas usando uma função / Na função, trocar 60 por 90 no bloco “vire à direita”</i> |
| 10 | <i>Desenhar livremente criando uma nova função / Usar blocos variados</i> |

Considerando o conjunto dos 10 desafios que compõem o jogo Artista da Hora do Código, a inspeção realizada indica que os desafios 6 e 7 são possivelmente os mais difíceis, devido às ações envolvendo ângulos e blocos de repetição aninhados. De fato, laços de repetição têm sido apontados por vários autores como fontes de dificuldades para iniciantes em algoritmos e programação [Qian e Lehman 2017]. O desafio 2 também tem sua dificuldade centrada na seleção dos ângulos e, mais do que isso, apresenta uma carga cognitiva significativamente maior que o primeiro desafio. Estima-se, assim, que muitos usuários precisarão de vários ciclos de tentativa e erro para concluir este desafio, o que pode ter impacto negativo na motivação ou na formulação da estratégia de resolução. Por este motivo, recomendamos que o jogo seja proposto a usuários que possam ser orientados individualmente, para que percebam os erros de forma construtiva.

Em contraste com os desafios anteriores, os desafios finais, apesar de introduzirem um novo conceito (funções), não exigem grande esforço cognitivo do usuário. De fato, a interface do jogo induz à ação correta nos desafios 8 e 9, deixando pouca margem para erros. No desafio 10, o usuário pode desenhar livremente, portanto relaxa-se a noção de solução certa ou errada. Ainda assim, estima-se que muitos usuários tenham dificuldade em utilizar toda variedade de blocos disponíveis, atendo-se somente à alteração de parâmetros.

Com a aplicação do CW, fica objetivamente caracterizado que o jogo Artista da Hora do Código alterna variados níveis de dificuldade e diferentes recursos de programação, em somente 10 desafios. Em comparação com o jogo “Labirinto Clássico”, por exemplo, que tem 20 desafios e recursos introduzidos de forma mais gradativa, o Artista se apresenta

como um tutorial mais denso e propenso a rejeição. Assim, recomendamos que este jogo seja proposto em meio a um percurso mais amplo, com outras atividades e jogos tutoriais antes e depois dele. Considerando sua utilização do conceito de “ângulo”, o Artista pode ter seu benefício potencializado se inserido em aulas de matemática que explorem este conceito. Por fim, recomendamos que este jogo seja evitado em casos de usuários que manifestem muita dificuldade em tutoriais anteriores.

5. Considerações Finais

Neste trabalho, aplicamos o método do Percurso Cognitivo a um dos jogos do Code.org, que ainda não tinha sido alvo de análises minuciosas em outras pesquisas. Detalhamos, ao longo do artigo, as fases e procedimentos da aplicação do método, tal como proposto originalmente por seus autores, e as decisões tomadas ao aplicá-lo ao jogo em questão. Com base nos resultados da inspeção, apresentamos algumas recomendações que podem ser úteis a diferentes atores responsáveis por selecionar, propor ou tutorar atividades que se apoiem na plataforma Code.org para desenvolver o Pensamento Computacional.

De forma geral, pode-se dizer que o CW revelou características do jogo Artista que não se encontram descritas na plataforma Code.org ou em outras fontes. Ao simular o uso do tutorial e os processos cognitivos que o usuário utiliza para resolver as tarefas que lhe são apresentadas, o CW revela o nível de dificuldade que este jogador experimenta, que se traduz na rapidez com que entende o funcionamento do jogo e os conceitos por trás dele. Tais aspectos são de grande relevância em uma plataforma como Code.org.

Sabendo que os resultados da inspeção podem antecipar erros e descompassos de entendimento por parte dos usuários, estimamos que o CW seja um método útil e aplicável a outros jogos tutoriais do Code.org. Visando a reprodutibilidade do trabalho e sua aplicação em outros contextos semelhantes, os formulários criados e respondidos encontram-se disponíveis em <https://bit.ly/sbie2020-cw-artista>.

Referências

- Bruner, J. (1984). Vygotsky's zone of proximal development: The hidden agenda. *New Directions for Child Development*.
- Cavalcante, A., Costa, L., e Araujo, A. (2016). Um estudo de caso sobre competências do pensamento computacional desenvolvidas na programação em blocos no Code.Org. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 5(1).
- Du, J., Wimmer, H., e Rada, R. (2016). “Hour of Code”: Can it change students' attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice*, 15:53–73.
- Farrell, D. e Moffat, D. C. (2014). Adapting cognitive walkthrough to support game based learning design. *Int. J. Game-Based Learn.*, 4(3):23–34.
- Freitas, M. e Morais, P. (2019). Possibilidade de desenvolvimento do pensamento computacional por meio do Code.Org: aplicado ao ensino fundamental (anos iniciais). *Anais do Workshop de Informática na Escola*, 25(1).
- GhasemAghaei, R., Arya, A., e Biddle, R. (2017). Affective walkthroughs and heuristics: Evaluating Minecraft Hour of Code. In Zaphiris, P. e Ioannou, A., editors, *Learning and Collaboration Technologies. Technology in Education*, pages 22–40.

- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52:200 – 210.
- Lambić, D., Đorić, B., e Ivakić, S. (2020). Investigating the effect of the use of Code.org on younger elementary school students' attitudes towards programming. *Behaviour & Information Technology*, 0(0):1–12.
- Martins, R., Reis, R., e Marques, A. B. (2016). Inserção da programação no ensino fundamental uma análise do jogo Labirinto Clássico da Code.org através de um modelo de avaliação de jogos educacionais. In *XXII Workshop de Informática na Escola (WIE)*.
- Piaget, J. (1972). A evolução intelectual da adolescência à vida adulta. *Development*, 15:1–12.
- Polson, P. G., Lewis, C., Rieman, J., e Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *Int. J. Man-Mach. Stud.*, 36(5):741–773.
- Popat, S. e Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128:365 – 376.
- Qian, Y. e Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Trans. Comput. Educ.*, 18(1).
- Raabe, A., Zanchett, G., e Vahldick, A. (2015). Jogos de programar como uma abordagem para os primeiros contatos dos estudantes com a programação. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 4(1).
- Wilson, C. (2013). What's up next for Code.org? *Computer*, 46(8):95–97.
- Yáñez Gómez, R., Cascado-Caballero, D., e Sevillano, J.-L. (2017). Academic methods for usability evaluation of serious games: A systematic review. *Multimedia Tools Appl.*, 76(4):5755–5784.