

RobotCode: um jogo educacional para auxiliar na aprendizagem de lógica de programação

Fabrizio Honda, Rafaela Melo, Fernanda Pires, Marcela Pessoa
{fhf.lic17,fpirez,msspessoa}@uea.edu.br,rmelo@icomp.ufam.edu.br
Universidade do Estado do Amazonas - Escola Superior de Tecnologia (UEA/EST)
Universidade Federal do Amazonas - Instituto de Computação (UFAM/IComp)

Historicamente, estudantes de Computação enfrentam dificuldades na compreensão dos conteúdos curriculares em disciplinas introdutórias, como linguagem de programação, ocasionando altas taxas de reprovação e evasão [2, 3]. Assim, acredita-se que é necessário desenvolver um conjunto de habilidades que possibilite resolver problemas de forma criativa, intitulado Pensamento Computacional (PC), desde a infância [12]. O PC pode ser uma ferramenta poderosa para desenvolver o raciocínio lógico [9] e pode ser estruturado em quatro pilares principais - decomposição, reconhecimento de padrões, abstração e algoritmo [1].

Diante disso, muitos pesquisadores têm utilizado de *Game-based learning* como metodologia em sala de aula. O fator lúdico dos jogos pode ser um elemento chave para despertar motivação/engajamento e, alinhado com os objetivos educacionais, pode proporcionar uma aprendizagem mais eficaz/divertida e auxiliar no desenvolvimento do Pensamento Computacional [6, 10].

Por exemplo, no trabalho de Pires et al. [10], os autores propuseram um jogo educacional intitulado “EcoLogic” para desenvolver o Pensamento Computacional (PC) e a consciência ambiental. O jogo possui formato *puzzle* e o objetivo do jogador é recolher os lixos espalhados pela fase, levando-os para as lixeiras correspondentes enquanto evita os inimigos. As fases foram estruturadas de modo a promoverem o PC, devendo o jogador utilizar de seu raciocínio lógico na elaboração de estratégias para resolvê-las.

Em Macena et al. [5], as autoras construíram “Hello Food”, um jogo educacional que possibilita praticar conteúdos introdutórios de programação, como condicionais, repetição e vetores, de forma lúdica e dinâmica. Cada fase possui dois níveis, cujos objetivos do jogador são de alocar os clientes em suas mesas respectivas, ordenando-os de forma decrescente, e preparar as refeições na cozinha através dos ingredientes, mesa de condicionais (estruturas “se” e “senão”) e fergão (laço de repetição “for”). As fases foram estruturadas levando em consideração o Pensamento Computacional, aplicado nas metas de jogo (condições para o jogador progredir).

Dessa forma, este trabalho apresenta “RobotCode”, um jogo educacional em fase de protótipo de média fidelidade. Com um público-alvo a partir de 10 anos de idade e do tipo *puzzle*, o jogo tem como objetivo estimular o raciocínio lógico, introduzir o usuário a uma experiência superficial do que seria o ato de programar na realidade e desenvolver os pilares do Pensamento Computacional.

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

A narrativa de RobotCode é contextualizada num cenário em que uma tempestade eletromagnética atingiu a Terra e removeu a programação dos robôs. O cientista Mr. Buzz é o responsável por recuperá-los e necessita da ajuda do jogador, que deverá programar os robôs “Boberto”, “Brainiac” e “Bender”. Para isso, o jogo proporciona um ambiente interativo e lúdico, em que o usuário pode testar os seus conhecimentos (Figura 1), com três fases que tratam sobre: i) movimentos básicos - mover, girar para a direita e esquerda; ii) laços de repetição e; iii) laços de repetição aninhados.



Figura 1: Tela inicial e primeira fase do jogo, respectivamente.

A mecânica se dá através da interação com blocos, que representam comandos que os robôs podem realizar. Ao apertar um bloco, o nome do comando é listado na área acima, em uma espécie de “terminal”; essa listagem é feita de cima para baixo, na ordem em que os comandos foram apertados. Quando o jogador apertar o botão verde, o robô executa os comandos na área à esquerda, simulando uma “compilação”. Para vencer as fases, o jogador deve movimentar o robô até o destino, representado por um indicador vermelho.

A concepção do jogo levou em consideração as estruturas cognitivas humanas, tendo como base a Teoria Cognitiva da Aprendizagem Multimídia [11], a Teoria dos Estágios de Jean Piaget Piaget [8] e também o Construcionismo de Seymour Papert [7], dando liberdade para o jogador construir suas próprias soluções, assim tornando-se o centro do processo de aprendizagem.

Optou-se por utilizar uma linguagem de alto nível para representar a codificação, pois o uso de uma programação visual ajuda o usuário a saber o que está acontecendo no “programa” que está executando. Ademais, o objetivo do jogo não é a aprendizagem de uma linguagem de programação, mas sim, a lógica. Dessa forma, não é necessário que o usuário escreva os códigos, visto que um dos motivos de desistência na escrita de código são os diversos erros de sintaxe que podem ser cometidos em alguma linguagem [4].

Atualmente, em versão de protótipo, o jogo precisa de elementos de *design*, implementação em uma *game engine* moderna e avaliações. Por ainda não estar completo, o *feedback* de avaliação pode ocasionar mudanças no *game design* ou incorporação de novas mecânicas, que serão futuramente implementadas.

REFERÊNCIAS

- [1] BBC. 2018. Introduction to computational thinking. (2018). <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>.
- [2] Jens Bennedsen and Michael E Caspersen. 2019. Failure rates in introductory programming: 12 years later. *ACM inroads* 10, 2 (2019), 30–36.
- [3] Yorah Bosse and Marco Aurélio Gerosa. 2015. Reprovações e trancamentos nas disciplinas de introdução à programação da Universidade de São Paulo: um estudo preliminar. In *Anais do XXIII Workshop sobre Educação em Computação*. SBC, 426–435.
- [4] Cárliston Galdino, Sebastião Neto, and Evandro Costa. 2015. KidCoder: uma proposta de ensino de programação de forma lúdica. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 26. 687.
- [5] Jeniffer Macena, Fernanda Pires, and Rafaela Melo. 2022. Hello Food: uma jornada de aprendizagem lúdica em algoritmos, programação e Pensamento Computacional. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*. SBC, 561–572.
- [6] Jean Machado and Almir Junior. 2019. Utilização de jogos como ferramenta para auxiliar o desenvolvimento do Pensamento Computacional: uma revisão sistemática. In *Anais do Workshop de Informática na Escola*, Vol. 25. 217–226.
- [7] Seymour Papert. 1980. "Mindstorms"Children. *Computers and powerful ideas* (1980).
- [8] J. Piaget. 1970. *Genetic Epistemology*. Columbia University Press. <https://books.google.com.br/books?id=LikSvgEACAAJ>
- [9] Fernanda Pires, Fabrizio F Honda, Guibson Silva, Rafaela Melo, Rosiane de Freitas, and Marcela Pessoa. 2019. A game proposal to develop computational thinking and environmental awareness. *Journal on Computational Thinking (JCTHink)* 3, 1 (2019), 111.
- [10] Fernanda Pires, Rafaela Melo, Jean Machado, Marcos S Silva, Fabrizio Franzoia, and Rosiane de Freitas. 2018. EcoLogic: um jogo de estratégia para o desenvolvimento do pensamento computacional e da consciência ambiental. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 7. 629.
- [11] John Sweller. 2011. Cognitive load theory. In *Psychology of learning and motivation*. Vol. 55. Elsevier, 37–76.
- [12] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.