ThinkTank: Um Jogo Construtivista para Ensino de Pensamento Computacional

Gabriel Coutinho Natucci, Marcos A. F. Borges

Laboratório de Informática, Aprendizagem e Gestão - Faculdade de Tecnologia - Universidade Estadual de Campinas
marcosborges@ft.unicamp.br, gabrielnatucci@gmail.com

Resumo: Este artigo discute e apresenta o desenvolvimento do jogo Thinktank, com base construcionista, para o ensino de pensamento computacional e programação. O jogo faz uso de programação em blocos em uma mecânica de fácil entendimento para atingir públicos variados, com pouco ou nenhum conhecimento em programação. Um diferencial deste jogo é a competição multijogador em tempo real pela internet, que permite a aprendizes se desafiarem e jogarem em arenas competitivamente, estimulando-os a melhorar e aprender continuamente novos algoritmos e estratégias para ganhar a partida. É apresentado um protótipo inicial do jogo, bem como futuras direções de pesquisa e melhorias.

Palavras-chave: *construcionismo; pensamento computacional; jogos sérios*.

1. Introdução

Os algoritmos, as tecnologias de comunicação e informação, além da inteligência artificial permeiam cada vez mais a vida cotidiana. Essa ubiquidade da tecnologia resultará em uma mudança no ambiente de trabalho e na sociedade do futuro. Nesse contexto, se faz necessário preparar a população para novas demandas e oportunidades, seja oferecendo a crianças a possibilidade de ter experiências com essas tecnologias, seja capacitando adultos a trabalhar em um ambiente de trabalho permeado por elas. Esse novo cenário requer um pensamento crítico e inovador para resolver problemas cada vez mais complexos, com pessoas habilitadas a pensar computacionalmente e algoritimicamente, fluentes no uso e concepção de novas tecnologias [1].

Pensar computacionalmente é uma forma de encarar problemas análoga ao questionamento em um pensamento científico [1]. O Pensamento Computacional (PC) é uma "abordagem para resolver problemas, projetar sistemas e entender comportamentos humanos que se baseia em conceitos da ciência da computação" [2]. O PC fundamenta-se em grande parte em conceitos de programação, porém não é definido somente por isso [3].

O desenvolvimento do PC não deve se limitar ao ambiente escolar formal; ele deve ser mais pervasivo, ubíquo e não diretamente associado a uma sala de aula e outros espaços formais de ensino. Uma forma de construir esse conhecimento é através do uso ou construção de jogos digitais (JD). Em particular, o uso de jogos para a educação promove maior engajamento e imersão (*flow*) nos estudantes, o que, por sua vez, relaciona-se a um maior aprendizado, absorção de conceitos e desempenho criativo em ambientes profissionais [4].

Com base nessa necessidade do desenvolvimento do PC em diferentes ambientes, e as vantagens inerentes a jogos digitais quanto à imersão e retenção de informação, este artigo apresenta a proposta do desenvolvimento de um protótipo de jogo digital construcionista online de estratégia e ação, denominado ThinkTank, com foco no desenvolvimento de PC de forma ubíqua. O jogo traz inspirações de títulos similares como *Robobuilder* [5] apresentando diferenciais quanto à imersão de seus jogadores, bem como facilitando o desenvolvimento de conceitos de PC através de mecânicas lúdicas.

Este artigo se encontra organizado da seguinte forma: a Seção 2 descreve e fundamenta os conceitos teóricos usados. A Seção 3 descreve as práticas pedagógicas, os conceitos de PC e as mecânicas de imersão aplicados no desenvolvimento do jogo. A Seção 4 descreve a implementação do jogo e suas implicações no aprendizado de PC, bem como suas atuais limitações. Finalmente, a Seção 5 ilustra perspectivas futuras para este projeto.

2. Fundamentação Teórica

2.1 Construtivismo e Construcionismo

O construtivismo é uma teoria pedagógica proposta pelo filósofo Jean Piaget, afirmando que o ser humano não recebe conhecimento de fontes externas e o compreende/aplica diretamente; o aprendiz primeiro constrói seu próprio conhecimento [6].

Outra abordagem pedagógica é a teoria sociocultural proposta por Vygotsky. Relacionada ao construtivismo (construtivismo social), ela difere-se por ter maior ênfase à construção do conhecimento através de interações e aspectos sociais como a família, comunidades, e cultura [7].

O paradigma construcionista foi criado por Papert como uma forma de descrever a construção do conhecimento através do computador [8]. Baseado no construtivismo, essa teoria se diferencia ao afirmar que o aprendizado ocorre ao construir um objeto; ela ainda difere-se do construtivismo social e piagetiano por ser mais situacional e pragmática [10], bem como por um maior envolvimento afetivo do aluno no aprendizado [9].

Neste trabalho, o jogo construído assemelha-se mais à prática construcionista de Papert, principalmente pelo

uso do computador como ferramenta e pela construção de robôs; o próprio jogo, neste caso, toma o papel de mediador do aprendizado.

2.2 Pensamento Computacional

O conceito de PC está intimamente ligado ao construcionismo proposto por Papert e possui muitas interpretações distintas, sendo definido de forma abrangente como uma forma de resolver problemas usando o computador como uma ferramenta para tal [1]. Desta forma, pode ser utilizado para o ensino e aprendizado das mais diversas habilidades e competências, como matemática, física e áreas relacionadas. Mesmo havendo pouco consenso entre as competências que envolvem PC, é possível entendê-las como um conjunto de habilidades a serem adquiridas. Neste trabalho, optou-se por definir PC como o seguinte conjunto de habilidades: coleta de dados (H1), análise de dados (H2), representação de dados (H3), abstração (H4), decomposição de problemas (H5), algoritmos e procedimentos (H6), Automação (H7), Simulação (H8) e paralelismo (H9) [11]. A escolha desta definição em detrimento de outras se deve a uma maior número de categorias específicas ligadas à análise de dados, bem como uma separação entre o uso de algoritmos, paralelismo e resolução de problemas.

3. Metodologia

ThinkTank é um jogo desenvolvido como uma forma de aplicar a prática construcionista para o aprendizado de PC, possuindo modos de jogo multiusuário (*multiplayer*) e monousuário (*singleplayer*). Para o desenvolvimento deste jogo, foi escolhida a *engine* de jogos Unity e a linguagem de programação C#, por serem tecnologias muito usadas em programação de jogos e possuírem uma vasta gama de tutoriais e suporte.

Em *Thinktank*, em vez de comandar um personagem diretamente através de comandos do teclado/console, o aprendiz/jogador deve codificar e construir a lógica de programação de um robô, colocado em uma arena para competir com robôs de outros jogadores; o jogador deve gerenciar, a partir de sua programação, uma reserva de energia do robô e com ela administrar sua movimentação e ações. Vence o jogador que conseguir melhor resultado sem esgotar suas reservas de energia.

O jogador programa o robô através de linguagem visual de blocos, similar à usada no software educacional *Scratch* [12], como ilustra a Figura 1. A programação visual foi escolhida por diminuir as barreiras de aprendizado de PC, além de ser mais compreendida por alunos quando comparada a uma linguagem tradicional [13]. Além disso, o uso de programação visual resulta em um aprendizado mais relevante por alunos com baixa confiança em suas próprias habilidades [14]. Finalmente, o uso dessa ferramenta aumenta a motivação e engajamento dos aprendizes [15]. Assim, essa escolha de interação traduz-se em uma forma mais igualitária, autônoma, motivadora e eficiente de aprendizado de PC.

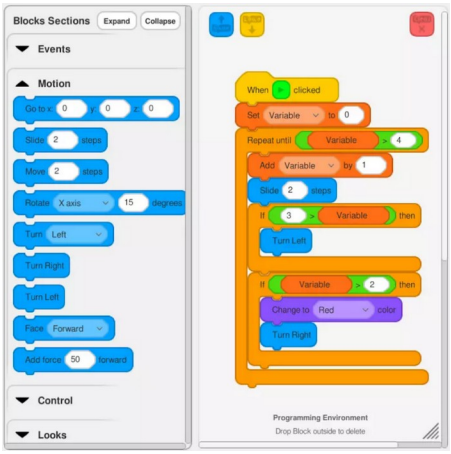

Figura 1. Programação em blocos usada no jogo [16].

A escolha do cenário e a estruturação como uma competição entre robôs já foi abordada na literatura, devido aos seguintes motivos [5]:

- facilidade na tradução entre os algoritmos abstratos e ações do robô no ambiente de jogo;
- o ato de programar robôs para executar tarefas simples é divertido e motivador em ambientes educacionais;
- o contexto (disputas de robôs) e objetivo do jogo (ganhar dos demais oponentes) são claros o suficiente para serem auto explicativos, auxiliando a prática construcionista.

Para tornar a experiência do jogo mais lúdica, foram desenvolvidos elementos de som e arte que remetem a consoles de jogos eletrônicos mais antigos e tradicionais (elementos nostálgicos): optou-se por um estilo de arte baseado em pixels (*pixel art*), em um cenário repleto de doces, no qual uma civilização altamente tecnológica, também feita de açúcar, está testando sondas robô para explorar um mundo feito de água (o principal combustível dessa civilização). Um exemplo do espaço de disputa e seus elementos de design é ilustrado na Figura 2.

Figura 2. Exemplo de cenário de disputa. Próprio(s) autor(es).

O modo monousuário foi desenvolvido de forma a executar o papel de *scaffolding* [7], ao mesmo tempo em que mantém o jogador aprendiz imerso na prática de programação. O jogo oferece uma série de robôs pré-programados, executados de forma autônoma, que servem para o jogador testar a lógica de seu próprio personagem e desafiar-se individualmente. Além disso, foi desenvolvida uma série de desafios (*puzzles*) para o

aprendiz resolver usando técnicas de PC. Esse design particular dos modos de jogo foi feito para ser usado de forma não competitiva, com os jogadores desafiando a si mesmos, disputando contra versões robóticas automatizadas, buscando ter novas experiências e ideias de melhorias nos algoritmos que estão implementando.

O modo multijogador usa a competição como forma de motivar jogadores a desenvolver algoritmos cada vez mais complexos, que possam competir satisfatoriamente com outros jogadores na Internet.

O jogo difere-se de outros na literatura, como *Robobuilder* [5] ou *Robocode* [17], por possuir modo de *multiplayer* online, enquanto essas plataformas se resumem a partidas com múltiplos jogadores de forma local, inserindo os robôs jogadores manualmente em uma partida. Em termos do modo *singleplayer*, as plataformas existentes focam somente na simulação de partidas com o uso de robôs autônomos, construídos por inteligência artificial ou disponibilizados por outros jogadores; em *Thinktank*, além de fazer uso desta mesma mecânica, foram desenvolvidas fases de tutorial e quebra-cabeças (*puzzles*) para estimular o jogador a programar de forma mais eficiente e trabalhar a sua competência em resolução de problemas. *Thinktank* traz também elementos de jogos comerciais, objetivando um maior engajamento do jogador e uma maior abrangência de público; entre esses elementos se encontra a própria estética do jogo, usando *pixel art* e uma perspectiva isométrica, diferente dos jogos analisados na literatura, que apresentam perspectiva *top-down* e pouca preocupação com a estética. O jogo ainda oferece mecânicas únicas de jogabilidade, como poderes especiais e itens de customização do robô, que permitem ao jogador adaptar seu robô com disparos mais fortes, melhor movimentação, etc; outra característica importante é a arena na qual os jogadores batalham, que possui obstáculos gerados aleatoriamente aumentando o nível de desafio da programação do robô e gerando uma diferenciação entre cada partida.

Devido à facilidade de uso e abrangência de habilidades de PC trabalhadas, é possível utilizar o jogo com alunos do ensino fundamental, médio e superior, tanto de forma autônoma quanto para ensino de áreas correlatas como ciência, tecnologia, engenharia e matemática (STEM). O jogo também pode vir a ser usado em dinâmicas de aprendizado de conceitos de física (trajetória de projéteis), matemática (usando a movimentação geral do robô), e até mesmo economia (escassez de recursos/gerenciamento de energia).

4. Conceitos de PC em *Thinktank*

O jogo ThinkTank permite a aprendizes de diversos níveis e formações desenvolverem o PC de forma lúdica e interativa. Ele trabalha pelo menos seis diferentes habilidades de PC:

- H1/H2: coleta de dados sobre o robô, como nível de energia restante, posição dos demais competidores no cenário, bem como sua posterior análise para tomada de decisão;
- H4/H5: redução da complexidade do contexto de programação para somente movimentações em uma arena pré-demarcada, permitindo ao aprendiz abstrair conceitos técnicos de forma mais simples e visual;
- H6/H7: embora não sejam trabalhadas diretamente, espera-se que os melhores jogadores proponham e testem diferentes algoritmos e procedimentos, visto que devem usar comandos visuais para realizar todas as ações de movimentação e estratégias de vitória de seu robô. A H6 em particular é praticada a todo instante pelo aprendiz jogador ao programar seu robô.

Desta forma, o jogo abrange a maior parte do que se considera como PC na literatura.

5. Conclusões e Perspectivas Futuras

Este artigo apresentou um protótipo inicial do jogo de disputa de robôs ThinkTank, com o objetivo de desenvolver o PC e conceitos de programação de forma lúdica e motivadora. O objetivo do jogo é atingir diversos públicos, podendo ser usado no ensino fundamental, médio e superior. O jogo pode ser usado individualmente, mas também possibilita a disputa entre participantes em uma forma multijogador com objetivo de tornar a atividade motivadora. É possível ainda usar o jogo como base para ensinar conceitos de STEM.

O primeiro protótipo do jogo foi desenvolvido, e algumas funcionalidades descritas neste texto ainda não estão disponíveis. Pesquisas para o desenvolvimento de modelos de avaliação do aprendizado de PC com base no jogo são importantes trabalhos futuros. Além disso, seria importante adaptar as mecânicas de programação, para permitir o uso do jogo em *smartphones* ou *tablets*, tornando-o mais responsivo. Finalmente, poder-se-ia explorar alternativas aos modos de jogo, abrindo novas possibilidades para professores usarem o jogo em sala de aula.

Agradecimentos

O trabalho foi conduzido com o apoio por parte da Secretaria de Cultura e Economia Criativa do Estado de São Paulo, através do Programa de Ação Cultural (PROAC), edital número 30/2019. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Bibliografia

[1] Shute, V. J.; Sun, C.; Asbell-Clarke, J. (2017) Demystifying computational thinking. *Educational Research Review* 22: 142–158. DOI: 10.1016/j.edurev.2017.09.003

[2] Wing, J. M. (2006) Computational thinking. *Communications of ACM* 49(3): 33-35. DOI: 10.1145/1118178.1118215

[3] Wing, J. M. (2008) Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering*

Sciences 366: 3717–3725. DOI: 10.1098/rsta.2008.0118
[4] Hamari, J. *et al.* (2016) Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior* 54: 170–179. DOI: 10.1016/j.chb.2015.07.045
[5] Weintrop, D.; Wilensky, U. (2012) RoboBuilder: A Program-to-Play Constructionist Video Game. in *Proc. of the Constructionism 2012 Conference*.
[6] Piaget, J. (1977) *The Origin of Intelligence in the Child*. (Penguin).
[7] Vygotsky, L. S. (1978) *Mind in society: the development of higher psychological processes*. (Harvard University Press).
[8] Papert, S. (1986) *Constructionism: A New Opportunity for Elementary Science Education*. (Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group).
[9] Valente, J. A. (1993) org. Computadores e conhecimento: repensando a educação. UNICAMP/NIED. Online: https://odisseu.nied.unicamp.br/wp-content/uploads/other-files/livro-computadores-e-conhecimento.pdf
[10] Ackermann, E. (2001) Piaget's Constructivism, Papert's Constructionism: What's the difference? In: Constructivism: Uses and Perspectives in Education 1 & 2 Research center in Education. p. 85-94.
[11] Borges, M., Zanetti, H. A. P.; Ricarte, I. L. M. (2016) Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira. Anais do XXVII Simp. Brasileiro de Informática na Educação (SBIE 2016), p. 21-30 DOI: 10.5753/cbie.sbie.2016.21.
[12] Maloney, J.; Resnick, M.; Rusk, N.; Silverman, B.; Eastmond, E. (2010) The Scratch Programming Language and Environment. *ACM Transactions on Computing Education* 10(4): art. 16. DOI: 10.1145/1868358.1868363
[13] Chao, P.-Y. (2016) Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education* 95, 202–215. DOI: 10.1016/j.compedu.2016.01.010
[14] Tsai, C.-Y. (2019) Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior* 95, 224–232. DOI: 10.1016/j.chb.2018.11.038
[15] Sáez-López, J.-M.; Román-González, M.; Vázquez-Cano, E. (2016) Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education* 97: 129–141. DOI: 10.1016/j.compedu.2016.03.003
[16] Play Mode Blocks Engine | Systems | Unity Asset Store. https://bit.ly/397vXPZ. Acesso em 26/04/2020.
[17] Esmail, B.; White, L. (2004) Robocode throughout the curriculum. *Journal of Computing Sciences in Colleges* 19(3): 311-313.