

Introdução ao Pensamento Computacional com Scratch

Charles Madeira

Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte
Campus Universitário, s/n – Lagoa Nova, Natal – RN – Brasil

charles@imd.ufrn.br

Abstract. *Nowadays, professionals of the most several areas spend much time on computers, studying and building computational models. Nevertheless, we are not yet teaching our young students that technology can be also used to create new knowledge. Therefore, it is urgently needed to redirect efforts to stimulate computational thinking in schools. This course combines the methodologies of problem-based and digital game-based learning with visual programming techniques in order to stimulate the development of computational thinking in order to use the computer as an instrument to increase human cognitive and operational power.*

Resumo. *Atualmente, profissionais das mais diversas áreas passam grande parte do seu tempo nos computadores, estudando e construindo modelos computacionais. Apesar disso, ainda não estamos ensinando aos nossos jovens alunos que a tecnologia também serve para criar conhecimento novo. Portanto, é preciso redirecionar urgentemente os esforços para que o pensamento computacional seja estimulado nas escolas. Este minicurso une as metodologias da aprendizagem baseada em resolução de problemas e em jogos digitais com técnicas de programação visual a fim de estimular o desenvolvimento do pensamento computacional, visando o uso do computador como um instrumento de aumento do poder cognitivo e operacional humano.*

1. Introdução

A globalização da sociedade e sua informatização colocaram o conhecimento e a informação em posições privilegiadas como fonte de poder e de valor e provocaram profundas mudanças na organização do trabalho e nas formas de ensinar e aprender. Essas novas formas de produção e distribuição do conhecimento alteraram a estrutura, o planejamento das informações por meio dos diferentes meios de comunicação, surgindo um novo universo para a geração de conhecimento. Portanto, a área da Educação, ao acompanhar as mudanças e evoluções tecnológicas, deveria usar as Tecnologias da Informação e Comunicação (TIC) para proporcionar, nesse sentido, um enriquecimento dos ambientes de ensino e de aprendizagem, oportunizando, assim, espaços de convivência e, conseqüentemente, espaços de inovações.

Neste sentido, uma direção que vem se popularizando nos últimos anos e tem gerado um grande número de iniciativas de uso das TIC na educação, envolvendo atividades de programação e robótica, é a do Pensamento Computacional [Wing 2006]. Diferentes definições e enfoques para o termo são presentes na literatura e um corpo de pesquisas robusto está em desenvolvimento em diferentes lugares do mundo [Wing 2014][Zanetti *et al.* 2016][Avila *et. al.* 2016]. Mas poderíamos definir o pensamento

computacional como uma habilidade de resolução de problemas que ensina os indivíduos a pensar e analisar uma situação ou um artefato, sendo independente do uso de tecnologia, conforme afirmam De Paula *et al.* (2014).

Por outro lado, Blikstein (2008) aponta que umas das etapas fundamentais para o desenvolvimento das habilidades do pensamento computacional é aprender a programar. Entretanto, mais do que aprender a programar, queremos que crianças, jovens e adultos programem para aprender coisas novas. O objetivo não é simplesmente conhecer a sintaxe de uma linguagem de programação (C, C++, Java, Python, etc.), mas conhecer a essência de um programa, a forma como ele deverá ser construído, ou seja, as estruturas de programação que serão utilizadas para a resolução de um problema de forma eficiente.

Para isso, o desenvolvimento do raciocínio lógico e o entendimento das funções das diferentes estruturas de programação são essenciais no processo de construção dos passos necessários para a resolução de problemas que, em um segundo momento, podem vir a se tornar programas escritos em qualquer linguagem. A essência do “aprender a programar, programar para aprender” está no fato de que, para desenvolver o pensamento computacional, é exigido do indivíduo que ele lide com diversas etapas no processo de resolução tais como a análise, a decomposição, a representação, a abstração e o algoritmo a fim de buscar soluções mais eficientes e, ao mesmo tempo, enriquecer a aprendizagem daquele conteúdo que está sendo trabalhado no projeto desenvolvido.

Todas as etapas necessárias nesse processo de resolução de problemas utilizado pelos cientistas da computação produzem uma mudança mental nas pessoas, ensinando-as a aprender a pensar logicamente, levando-as a adquirir habilidades específicas que precisamos desenvolver nas crianças e nos jovens alunos do século XXI, independentemente deles virem a se tornar futuros cientistas da computação. A questão é que as habilidades desenvolvidas com a programação são fundamentais pois ajudam a ver o mundo de maneira mais criativa e crítica, algo necessário para ter uma voz ativa e participar plenamente da sociedade, podendo ser aplicada no desenvolvimento de habilidades de liderança, gestão estratégica, análise de processos, inovações gerenciais e até mesmo empreendedorismo.

Portanto, esses conhecimentos de programação de computadores necessários para realizar tarefas cognitivas e de maneira automatizada deveriam ser construídos ao longo de toda a vida escolar, permitindo que as habilidades adquiridas viessem a servir como suporte ao raciocínio humano no processo de resolução de problemas. Neste sentido, este minicurso se propõe a estimular o desenvolvimento do pensamento computacional, visando o uso do computador como um instrumento de aumento do poder cognitivo e operacional humano.

2. Fundamentação Teórica

A atenção do aprendiz deve estar concentrada sobre assuntos de seu interesse para que ele venha a se engajar verdadeiramente com uma determinada tarefa [Piaget 1975]. Por esta razão, o processo de aprendizado precisa ser uma experiência agradável sob o seu ponto de vista para que com motivação torne a prática mais eficaz. Para tanto, faz-se necessário que as abordagens metodológicas de ensino-aprendizagem contemplem estratégias didático-pedagógicas que permitam a construção do próprio conhecimento pelos aprendizes, encarando-os como protagonistas dos processos de ensinar e aprender,

e ocasionando a formação de sujeitos pensantes, competentes e ágeis na resolução de problemas.

Neste sentido, uma abordagem que tem se mostrado instigante para os jovens no processo de desenvolvimento das estruturas básicas do pensamento computacional é a aquela que une as metodologias da aprendizagem baseada em resolução de problemas [Boud e Feletti 1998][Savery 2006] e da aprendizagem baseada em jogos digitais [Prensky 2007][Felicia 2014] com técnicas de programação visual [Marji 2014].

A metodologia de aprendizagem baseada em resolução de problemas é uma estratégia formativa através da qual os aprendizes são confrontados a problemas contextualizados para os quais se empenham em encontrar soluções significativas, desenvolvendo assim o raciocínio lógico, o pensamento crítico e a criatividade. Quando associada a brincadeiras e atividades lúdicas, se insere como uma ótima forma de estímulo ao aprendizado e ao desenvolvimento de novas habilidades [Mattar 2010].

A metodologia de aprendizagem baseada em jogos digitais vem sendo considerada como uma estratégia diferenciada para o aprimoramento do processo ensino-aprendizagem pois acredita-se que se os jogos forem transportados para o ambiente educacional de forma planejada e criteriosa, poderão surgir boas maneiras de ensino-aprendizagem e desenvolvimento de diversas habilidades e competências num contexto disciplinar e transdisciplinar [Chuang e Chen 2009][Whitton 2014].

Em relação ao ensino de programação, a metodologia da aprendizagem baseada em jogos desmistifica as dificuldades na medida em que a complexidade da programação passa a ser tratada de forma lúdica em formato de jogo digital. Aprender jogando é a mola mestra desta metodologia. A programação é inserida neste contexto a partir de um paradigma visual, baseado em blocos, onde ações são agrupadas para solucionar uma determinada tarefa. Por exemplo, ao criar um programa para que um robô percorra um trajeto em forma de um polígono, é essencial que o programador gere um conjunto finito de passos ordenados, ou algoritmo, indicando o que será necessário fazer para atingir o seu objetivo. Utilizando uma lógica parecida, desafios mais complexos podem ser construídos, incluindo outros tipos de jogos como, por exemplo, criar uma montanha russa (exposição dos conceitos de velocidade, altura, etc). Jogos sobre lixo, reciclagem, energia solar e outras formas sustentáveis de energia podem ser facilmente incorporados.

3. Metodologia

O presente minicurso se propõe a fazer uso de uma metodologia de desenvolvimento experimental para estimular o pensamento computacional através da concepção de jogos digitais no ambiente *Scratch*¹ (ver Figura 1). Scratch é um ambiente de programação visual criado em 2007 pelo *Media Lab* do *Massachusetts Institute of Technology*, tendo em vista as experiências com a linguagem LOGO desenvolvida por Papert (1980). Scratch visa ir além dessa linguagem em três aspectos: fazer a linguagem mais suscetiva à manipulação, mais social e mais significativa. Assim a forma como os blocos podem ser manipulados lhe confere uma possibilidade de aprendizagem auto-gerida através da prática de manipulação e teste dos projetos. A versão online do ambiente permite que

¹ <https://scratch.mit.edu/>

usuários interajam entre si, critiquem e aprendam com os projetos dos outros. Além disso, por não exigir o conhecimento prévio de outras linguagens de programação, Scratch se mostra muito mais acessível para indivíduos que estão começando a programar.

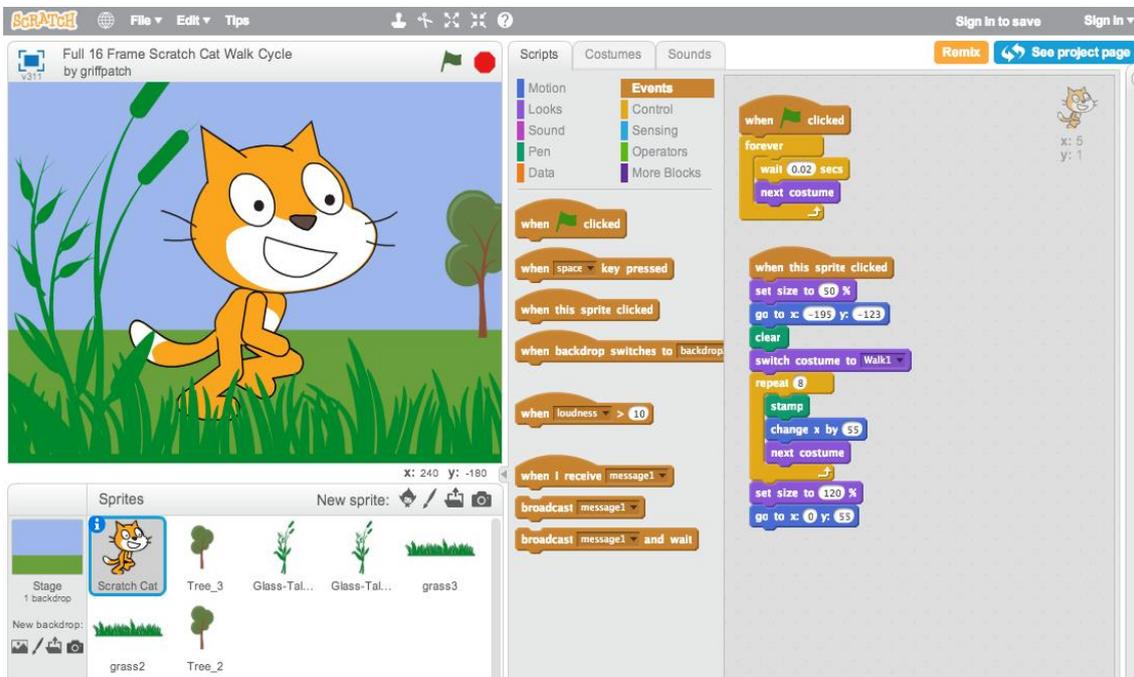


Figura 1. Tela principal do ambiente Scratch onde se encontram a janela do cenário de edição, a janela de imagens, a janela das abas de funcionalidades disponíveis e a janela de programação.

Neste contexto, serão introduzidas as bases do pensamento computacional, objetivando estimular os participantes a resolver inicialmente alguns problemas introdutórios, aproveitando ao mesmo tempo para familiarizá-los com o ambiente de programação visual e, em seguida, desenvolver um jogo digital tendo como cenário a problemática da caça ao rato (ver Figura 2) a fim de adquirir as habilidades necessárias de forma lúdica e divertida.



Figura 2. Problemática da caça ao rato. O rato se move em direção ao queijo com o intuito de comê-lo e o gato se move em direção ao rato com o intuito de capturá-lo.

A caça ao rato consiste em um cenário no qual existem três objetos que estão presentes: um gato, um rato e um pedaço de queijo. O rato tem como objetivo de se mover em direção ao queijo com o intuito de comê-lo e o gato tem como objetivo se mover em direção ao rato com o intuito de capturá-lo para não deixá-lo chegar até o queijo. Para isso, é preciso que todo o processo de execução das tarefas desses personagens seja idealizado e programado. Por exemplo, se considerarmos que o rato deve ser um personagem automatizado (controlado pelo computador) e o gato deve ser controlado pelo jogador, precisaremos definir os requisitos para o desenvolvimento da solução e, em seguida, implementá-la. Para isso, eventos deverão ser controlados através de estruturas de condição e de estruturas de repetição de forma que possa existir uma evolução de cada personagem presente no ambiente simulado. É o caso quando o jogador pressionar determinadas teclas do teclado que deverão ter como consequência a movimentação do gato no ambiente. Neste caso, estruturas condicionais serão necessárias para indicar a funcionalidade de cada uma das teclas. Este tipo de estímulo ao pensamento computacional será o foco deste minicurso, colocando em prática estruturas básicas de lógica de programação tais como laços, condicionais, variáveis e, eventualmente, funções.

4. Considerações Finais

Neste minicurso, espera-se proporcionar a base necessária para que os participantes ganhem interesse em carregar a bandeira do desenvolvimento do pensamento computacional dentro da sala de aula, sendo capacitados para tal fim através da experiência no emprego de metodologias lúdicas. Além disso, visa-se demonstrar a simplicidade e a viabilidade do emprego das metodologias de aprendizagem baseadas em resolução de problemas e em jogos digitais quando associadas à programação visual.

Referências

- Avila, C., Bordini, A., Marques, M., Cavalheiro, S. e Foss, L. (2016). Desdobramentos do Pensamento Computacional no Brasil. Anais do Simpósio Brasileiro de Informática na Educação (SBIE-2016), p.200-209.
- Blikstein, P. (2008). O Pensamento Computacional e a Reinvenção do Computador na Educação. Disponível em: <<http://bit.ly/1IXlbNn>>.
- Boud, D. and Feletti, G. (1998). The Challenge of Problem-Based Learning, Kongan Page.
- Chuang, T.-Y., Chen, W.-F. (2009) “Effect of Computer-Based Video Games on Children: An Experimental Study”. Educational Technology & Society, 12(2):1-10.
- De Paula, B., Valente, J. e Burn, A. (2014). O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra. Currículo sem Fronteiras, v. 14, n. 3, p.46-71.
- Felicia, P. (2014). Game-based Learning: Challenges and Opportunities. Cambridge Scholars Publishing.
- Marji, M. (2014). Learn to Program with Scratch: A Visual Introduction to Programming with Games, Art, Science, and Math. No Starch Press.

- Mattar, J. (2010). Games em educação: como os nativos digitais aprendem. Pearson Prentice Hall.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.
- Piaget, J. (1975). A equilibração das estruturas cognitivas. Zahar.
- Prensky, M. (2007). Digital Game-Based Learning. McGraw-Hill.
- Savery, J. (2006). Overview of Problem-Based Learning: Definitions and Distinctions. In The Interdisciplinary Journal of Problem-Based Learning, 1(2).
- Whitton, N. (2014) “Digital Games and Learning: Research and Theory”. Routledge.
- Wing, J. (2006). Computational Thinking. Communications of the ACM, 49(3):33-36.
- Wing, J. (2014). Computational Thinking Benefits Society. 40th Anniversary Blog of Social Issues in Computing. Disponível em: <<http://bit.ly/2d9PrKn>>.
- Zanetti, H., Borges, M. e Ricarte, I. (2016). Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira. Anais do Simpósio Brasileiro de Informática na Educação (SBIE-2016), p.21-30.