

GeDAI: Um Sistema para Criar Desafios de Programação de Jogos Digitais para Promover o Pensamento Computacional

Ângelo Magno de Jesus¹, Ismar Frango Silveira²

¹Departamento de Computação, Instituto Federal de Minas Gerais, *campus* Ouro Branco, MG, Brasil.

²Universidade Presbiteriana Mackenzie, São Paulo, SP, Brasil

angelo.jesus@ifmg.edu.br, ismar.silveira@mackenzie.br

Abstract. *Developing educational activities in which students solve Computational Thinking (CT) programming problems can be challenging, since there can be a considerable cognitive load involved in these processes. In this context, this paper presents an online system whose main innovation is allowing teachers to generate CT challenges, based on mini-games programming, easily calibrating the reasoning component involved in solving these problems. In this sense, students can access the challenges through a graphical interface that provides, through visual elements, partial information for its solution, such as blocks of programming commands and their possible arrangements within the algorithm.*

Key-words: *Computational Thinking, Game Development, Problem-Based Learning, Cognitive Load.*

Resumo. *Desenvolver atividades educacionais voltadas para resolução de problemas de programação do Pensamento Computacional (PC) pode ser desafiador, dada a carga cognitiva envolvida nestes processos. Neste contexto, este artigo apresenta um sistema on-line cujo o principal diferencial é permitir que professores possam gerar desafios do PC, voltados para programação de minigames, calibrando de forma fácil o componente de raciocínio envolvido na solução destes problemas. Neste sentido, os estudantes podem acessar os desafios por meio de uma interface gráfica que disponibiliza, por meio de elementos visuais, informações parciais para sua solução, como blocos de comando de programação e suas possíveis disposições dentro do algoritmo.*

Palavras-chave: *Pensamento Computacional, Desenvolvimento de Games, Aprendizagem Baseada em Problemas, Carga Cognitiva.*

GeDAI: Um Sistema para Criar Desafios de Programação de Jogos Digitais para Promover o Pensamento Computacional

Ângelo Magno de Jesus e Ismar Frango Silveira

Introdução

O Pensamento Computacional (PC) é uma abordagem que pode contribuir para o desenvolvimento de habilidades de resolução de problemas em crianças, tornando-as mais autônomas em seu aprendizado e contribuindo para o seu sucesso em situações da vida real [1]. No entanto, os professores podem enfrentar dificuldades ao propor atividades de resolução de problemas do PC para seus estudantes. Isto pode ocorrer devido a carga cognitiva envolvida no processo de resolução destes problemas. Este artigo apresenta o sistema on-line GeDAI (Gerador de Desafios Automático e Inteligente) que visa permitir que os professores gerem desafios relacionados à programação de games e de animações interativas, para plataforma Scratch (<https://scratch.mit.edu/>), para apoiar suas aulas.

O GeDAI se baseia no framework conceitual para projeto de atividades educacionais 3C3R proposto por W. Hung [2]. Neste sentido, o sistema web permite que professores gerem problemas de programação (enviando seus próprios projetos para serem utilizadas como base ou utilizando os já disponíveis no sistema) e definam a carga cognitiva envolvida no componente raciocínio de acordo com o nível de conhecimento dos alunos em programação. O sistema também inclui uma plataforma de envio e acompanhamento das atividades. Além disso, estudantes podem utilizar o repositório de desafios para criarem seus próprios jogos digitais. A principal base inicial deste projeto foi desenvolvida nos trabalhos disponíveis em [3] e [4], portanto este texto se baseia nestas fontes. No entanto, este artigo se diferencia por apresentar o sistema como um todo, incluindo novas funcionalidades que ainda não foram apresentadas como, por exemplo, a geração de feedbacks automáticos para as soluções enviadas pelos estudantes.

Contexto Educacional. O principal público alvo do GeDAI está associado a estudantes do Ensino Fundamental II ou estudantes do primeiro ano do Ensino Médio da Educação Básica. O software se destina para realização de atividades do PC que envolvem desenvolvimento de habilidades de criatividade, resolução de problemas e de projeto de algoritmos. Neste sentido, a abordagem envolve as disciplinas de Programação de Computadores, Algoritmos ou Criação de Jogos.

Objetivo. O principal objetivo do software proposto é o de permitir que professores criem e apliquem desafios de programação cognitivamente adequados para seus estudantes, desenvolvendo a capacidade de resolução de problemas e criatividade dos mesmos.

Abordagem pedagógica. O software é focado em abordagens construtivistas e/ou construcionistas, uma vez que pode envolver atividades em que o aluno constroi seu próprio conhecimento por meio do método de aprendizagem ativa denominado Aprendizagem Baseada em Problemas (ABP). Neste sentido, o sistema dá suporte para desenvolvimento de atividades de resolução de desafios customizados, em que os estudantes, de forma colaborativa, devem lançar mão de diferentes estratégias e heurísticas para descobrir e/ou combinar comandos de programação, disponibilizados pela interface do sistema conforme definido pela carga cognitiva informada pelo professor, para gerar um *minigame*, como pode ser visto em [3].

Diferenciais, Potenciais Inovações e Impacto nos Processos Educacionais

Os principais diferenciais do GeDAI são: **1. Balancear o componente raciocínio:** A resolução de problemas de programação voltados para desenvolvimento de games, mesmo aqueles simples, pode ser uma tarefa extremamente complexa para estudantes iniciantes, devido à carga cognitiva requerida neste processo. **Durante**

as atividades educacionais com Scratch descritas em [3], foi observado que muitos estudantes ficavam perdidos ou distraídos com o grande número de comandos (blocos) e possibilidade de combinações entre eles. Neste sentido, baseado no Framework 3C3R, o sistema permite gerar desafios que apresentam, de acordo com a definição do professor, informações parciais para sua solução, como comandos de programação que podem ser utilizados e/ou suas possíveis disposições dentro do algoritmo. Portanto, **não se trata apenas de um repositório de situações problema, uma vez que o sistema permite que, por meio de engenharia reversa automática em projetos Scratch, o professor calibre de forma simples a carga cognitiva dos problemas, apenas definindo o percentual de blocos de programação (e suas conexões) que estará disponível para seus alunos, para que estes estudantes acessem os desafios por meio de uma interface gráfica contendo ricos recursos visuais representando os objetos disponibilizados para aliviar o componente raciocínio da atividade.**

2. Desenvolver a criatividade: Os estudantes podem criar novos jogos de acordo com a combinação de soluções de diferentes atividades disponíveis na plataforma.

3. Feedback: Ao enviar a solução (arquivo do Scratch) para o sistema, os estudantes podem ter um feedback sugerindo o que pode possivelmente estar errado em sua solução como, por exemplo, um bloco de comando faltando ou fora do lugar.

Quanto ao impacto nos processos educacionais, a abordagem para criação de desafios de programação, no modelo proposto pelo software, foi validada com estudantes do ensino fundamental em [3]. Neste sentido, os resultados descritos mostraram que a abordagem foi capaz de motivar os estudantes e exercitar habilidades do PC por meio de atividades de programação de games.

Aspectos Tecnológicos

O GeDAI apresenta funcionalidades de uma plataforma web capaz de gerenciar perfis de professores e estudantes. No entanto, este documento se concentrará em descrever a parte principal do sistema que se trata do módulo utilizado para gerar e calibrar os desafios de programação em Scratch. As Figuras 1 e 2 mostram o diagrama de classes simplificado do módulo conforme [4], seguindo o padrão GoF Bridge [5]. Cabe observar que o projeto do sistema envolveu a realização de engenharia reversa em projetos do Scratch, que podem ser enviados pelo professor, para criação de novos desafios, para que isso fosse possível, a engenharia do sistema envolveu a aplicação do padrão de projeto GoF Interpreter [5]. Desta forma, as Figuras 2 e 3 ilustram a gramática utilizada para interpretação e geração do código do desafio.

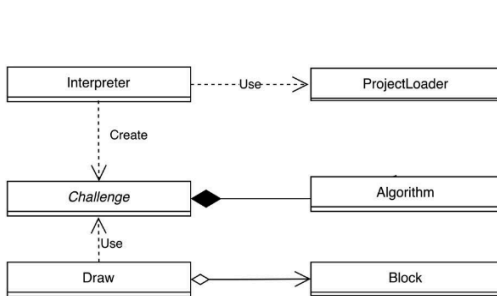


Figura 1

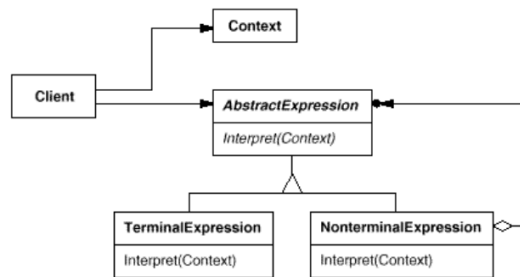


Figura 2

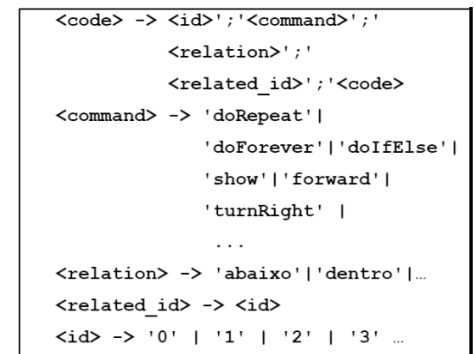


Figura 3

Apresentação do Software

Professor:

1. Visualizar uma lista de projetos Scratch pré-cadastrados por ele e por outros professores (Figura 4);
2. Enviar um projeto Scratch para o que o sistema o transforme em um novo desafio (Figura 5);
- 3.1 Gerar um desafio, primeiramente escolhendo um dos projetos na lista e selecionando a carga de informações que será disponibilizada (Figura 6);
- 3.2 Enviar o desafio para os estudantes resolverem (Figura 7);
4. Visualizar a lista de atividades de seus estudantes com os status de **Feito Corretamente**, **Feito Incorretamente** ou **Aguardando**.

Estudante:

1. Visualizar a lista de desafios disponibilizados pelo professor - semelhante a tela da Figura 4;
2. Visualizar o desafio contendo as informações para sua solução - similar a Figura 7, mas um campo para enviar a solução;
3. Enviar uma proposta de solução e receber um feedback de possível problema na solução (Figura 8).



Figura 4

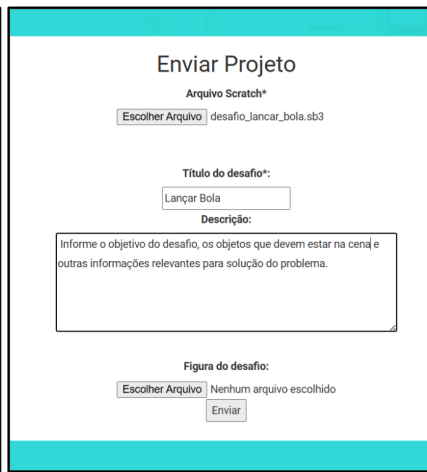


Figura 5

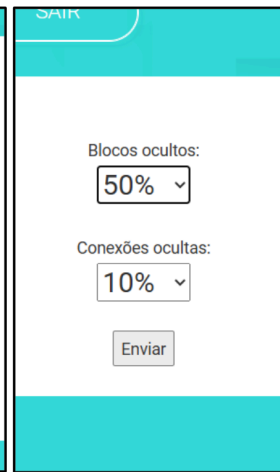


Figura 6

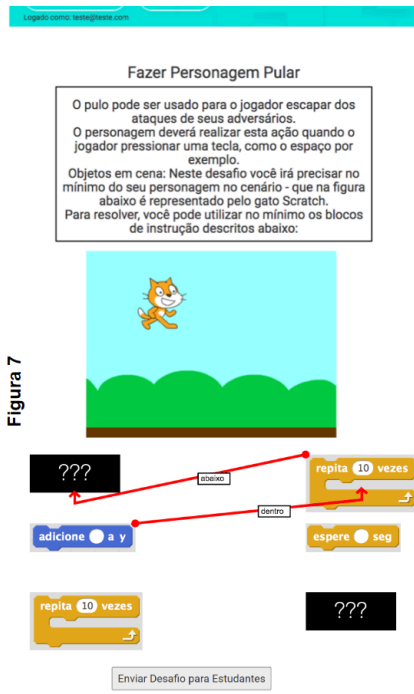


Figura 7



Figura 8

Discussão

Conclusão. Como apresentado, o GeDAI possui bases conceituais e tecnológicas que podem ser promissoras para aplicação de atividades de desenvolvimento do PC. **Trabalhos futuros.** Como trabalhos futuros, pretende-se aplicar a plataforma para diversas turmas de estudantes para avaliar sua utilização. **Limitações.** Por se tratar de um protótipo desenvolvido por uma equipe de apenas duas pessoas, o sistema web possui algumas limitações. A principal limitação pode estar associada à indisponibilidade de alguns blocos de comando muito específicos da plataforma scratch. No entanto, isso pode ser resolvido com relativa facilidade, pois novos itens podem ser adicionados facilmente por meio do padrão de projeto adotado na arquitetura. Além disso, existe a indisponibilidade de funcionalidades apuradas para gestão de contas de usuário (ex.: recuperação de senhas). Esta limitação pode ser facilmente resolvida por se tratar de um recurso muito comum em sistemas genéricos.

Referências

- [1] V. Barr e Chris Stephenson, "Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?", Acm Inroads, vol. 2, no. 1, pp. 48-54, 2011.
- [2] W. Hung, "The 3C3R model: a conceptual framework for designing problems in PBL", Interdisciplinary Journal of Problem- Based Learning", vol. 1, no. 1, pp. 55-77, 2006.
- [3] Jesus, Â. M. A Collaborative Learning Strategy for Computational Thinking Development through Game Programming. Tese de Doutorado. Programa de pós-graduação em Ensino de Ciências e Matemática. Universidade Cruzeiro do Sul. 2020.
- [4] Jesus, Â. M., & Silveira, I. F. (2020, October). A Generator of Game Programming Challenges for Computational Thinking Development. In 2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO) (pp. 1-9). IEEE.
- [5] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design patterns. Ele.