

## Combinando desafios e aventura em um jogo para apoiar a aprendizagem de programação em vários níveis cognitivos

Vanessa F. Dantas<sup>1</sup>, Eline R. de Macedo<sup>1</sup>, José Raul B. Andrade<sup>1</sup>, Danilo Raniery A. Coutinho<sup>1</sup>, Ahemenson F. Cavalcante<sup>1</sup>, Thiago G. Vasconcelos<sup>1</sup>, Mary Ellen de S. Pereira

<sup>1</sup>Departamento de Ciências Exatas – DCE

Universidade Federal da Paraíba (UFPB) – Campus IV – Rio Tinto, PB – Brazil

{vanessa, eline.raquel, raul.andrade, danilo.coutinho, ahemenson.cavalcante, thiago.gomes }@dce.ufpb.br , maryellen\_br@hotmail.com

***Abstract.** Many works have been proposed in order to support programming teaching and learning and minimize its various challenges. In this scenario, the production of educational games grows continually. This paper presents a game to follow the apprentice along all stages of his cognitive learning, keeping the focus on ludicity and immersion.*

***Resumo.** Muitas são as iniciativas no sentido de apoiar o ensino-aprendizagem de programação e minimizar seus diversos desafios, inclusive através de jogos. Nesse cenário, a produção de jogos educacionais vem crescendo. O presente trabalho propõe a criação de um jogo que possa acompanhar o aprendiz em todos os estágios de seu aprendizado cognitivo, preservando a ênfase no caráter lúdico e na imersão do jogador.*

### 1. Introdução

Há anos os altos índices de evasão e reprovação nas disciplinas iniciais de programação têm despertado o interesse e a preocupação de pesquisadores e educadores em todo o mundo, como relatam Delgado et al (2004), Hinterholz (2009), Prietch e Pazeto (2010) entre outros. Segundo Neto e Schuvartz (2007), acadêmicos iniciantes, ao se depararem com a disciplina, sentem-se incapazes de programar, devido ao conjunto de habilidades que a programação exige como capacidade para solucionar problemas, raciocínio lógico, habilidade matemática, capacidade de abstração, entre outras.

Em uma revisão sistemática recente, Aureliano e Tedesco (2012) evidenciam que as propostas para apoiar o ensino-aprendizagem da disciplina e combater esses problemas são bastante diversificadas, abrangendo desde ambientes para aprendizagem cooperativa e ferramentas para suporte à programação em pares, até o uso de ontologias e experiências com robótica.

A partir da análise de várias dessas abordagens inovadoras, e diante do interesse despertado pelos jogos educacionais, o presente trabalho descreve um jogo de aventura para motivar e desafiar os alunos, oferecendo oportunidades de evoluir ao longo das fases à medida que praticam seus conhecimentos de programação.

## 2. Jogos para o ensino de programação

Segundo Moratori (2003), o jogo pode ser considerado como um importante meio educacional, pois propicia um desenvolvimento integral e dinâmico nas áreas cognitiva, afetiva, lingüística, social, moral e motora, além de contribuir para a construção da autonomia, criatividade, responsabilidade e cooperação de crianças e adolescentes.

Rapkiewicz et al (2006) reforça essa ideia quando afirma que o uso de jogos de forma lúdica propicia flexibilidade e criatividade fazendo o aluno explorar, pesquisar, encorajando o pensamento criativo, saciando a curiosidade, alimentando a imaginação e estimulando a intuição, contribuindo assim para o aprendizado.

Alguns jogos recentes focam no desenvolvimento do pensamento algorítmico e da habilidade de resolver problemas. Para isso, propõem uma linguagem própria e estimulam os jogadores a usar as instruções disponíveis para solucionar desafios como fazer um robô andar ou criar uma animação 3D. São exemplos desse tipo de jogo o APIN (DIM, 2011), o ALICE (ALICE, 2010), o TALENT (MARAGOS, 2007) e o Takkou (BARBOSA, 2001).

Embora essas iniciativas se destaquem pela ludicidade e pela imersão dos alunos em seus ambientes, a necessidade de aprender uma nova linguagem conceitual tende a aumentar a curva de aprendizagem, além de poder causar confusão entre sintaxes e paradigmas. O ideal seria ter jogos em que o aluno utilizasse as linguagens de programação já conhecidas.

Nesse sentido, foram criados aplicativos como o Robocode (SANTOS, 2002), o WuCastle (EAGLE, 2008), e o Code Spell (CODE SPELL, 2013). Essas propostas combinam bem o caráter lúdico dos jogos e a prática de linguagens de programação como Java, mas nem sempre apoiam aqueles que estão iniciando seu aprendizado, uma vez que focam na competência de criar códigos.

O ProGame (DANTAS, 2011) e o Entrando Pelo Cano (SCAICO, 2012) são exemplos de jogos criados com o objetivo de propor desafios de programação em linguagens conhecidas (Python e Java), tendo como referência os diferentes níveis da Taxonomia de Bloom em relação à programação para iniciantes (DE JESUS, 2009).

No ProGame, um cientista viaja por várias partes do mundo em busca de novas tecnologias e se depara com desafios de programação, como a resolução de testes de mesa e a ordenação de instruções de um código, como condições para progredir no jogo. Os cenários apresentam inimigos e riscos, e a ferramenta usada para o desenvolvimento foi o Game Maker. Apesar de ter uma ideia original e propor desafios para dois níveis da taxonomia, a tecnologia utilizada oferece poucas possibilidades de expansão do jogo.

No Entrando Pelo Cano, o jogador precisa organizar diversas caixas de mercadorias em uma fábrica, escolhendo os tipos mais adequados para diversas variáveis conforme seus conteúdos. A aplicação é executada na web, possui um elaborado design instrucional e combina bem elementos de jogabilidade, fantasia e aprendizagem. Entretanto, apenas o primeiro nível da taxonomia é explorado, e com um conhecimento bastante pontual envolvendo tipos de dados.

Considerando os aspectos positivos e as limitações do ProGame e do Entrando Pelo Cano, o desafio de propor um novo jogo consistia em definir um enredo desafiador, escolher uma tecnologia adequada, e equilibrar bem diversão e aprendizagem contemplando os diversos níveis cognitivos.

### 3. Robotimov

Segundo Tarouco (2005), os jogos se tornam atrativos quando possuem regras bem definidas, têm metas que geram motivação, são interativos, favorecem a aprendizagem através de *feedback*, e são divertidos. Com essas diretrizes em mente, a equipe iniciou a discussão sobre um enredo que pudesse chamar a atenção do público-alvo, ou seja, dos alunos das disciplinas de programação, e que também despertasse sua curiosidade, transportando-os para um contexto de fantasia.

A proposta escolhida foi a de um parque temático com robôs em diferentes situações cotidianas, e que foi alvo de sabotagem. O jogador precisaria realizar ajustes nos códigos-fonte dos robôs para que eles voltassem a ter o comportamento esperado. Assim, a inserção dos desafios estaria diretamente relacionada ao tema do jogo, e a flexibilidade na criação de fases seria garantida, uma vez que, a cada novo nível, o jogador poderia interagir com robôs animais em uma floresta, robôs músicos em um bar, ou ainda robôs praticando esportes.

De modo a estimular a prática dos diferentes níveis cognitivos do aprendiz de programação, foram elaborados desafios de programação que pudessem ser incorporados ao Game Design da aplicação, tendo como base a interpretação da Taxonomia de Bloom feita por De Jesus e Raabe (2009). Alguns exemplos dos desafios elaborados podem ser vistos na Figura 1.

- ❖ Completar um código com os elementos de sintaxe que faltam
- ❖ Acompanhar a execução de trechos de código através do preenchimento de testes de mesa
- ❖ Identificar tipos de erro (sintático, semântico, lógico) presentes em trechos de código
- ❖ Ordenar e (tabular) um conjunto de instruções de modo a realizar uma determinada tarefa
- ❖ Escolher, dentre algumas instruções, a mais adequada para completar um trecho de código

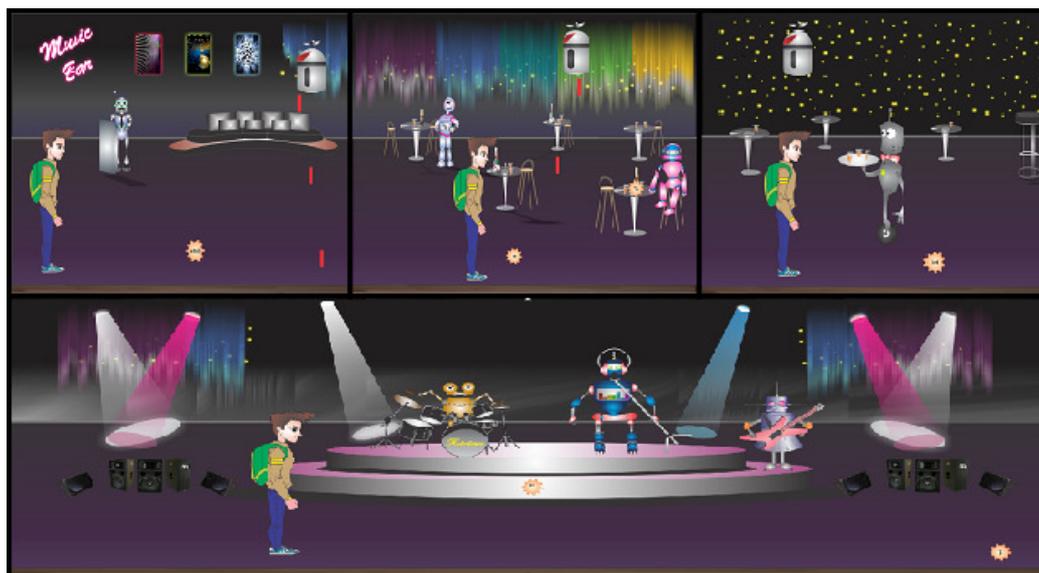
**Figura 1 – Desafios de programação para as fases do Jogo Robotimov**

É importante destacar que a proposta do jogo não é apresentar as fases com esses desafios numa estrutura hierárquica e fixa, mas sim permitir que o jogador escolha em que sequência deseja resolver os problemas, podendo priorizar a prática dos conteúdos em que tenha maior facilidade, ou reforçar habilidades pouco desenvolvidas. Contudo, o jogo poderá apresentar sugestões para conduzir o jogador nessa escolha.

### 4. Protótipo – Fase Balada

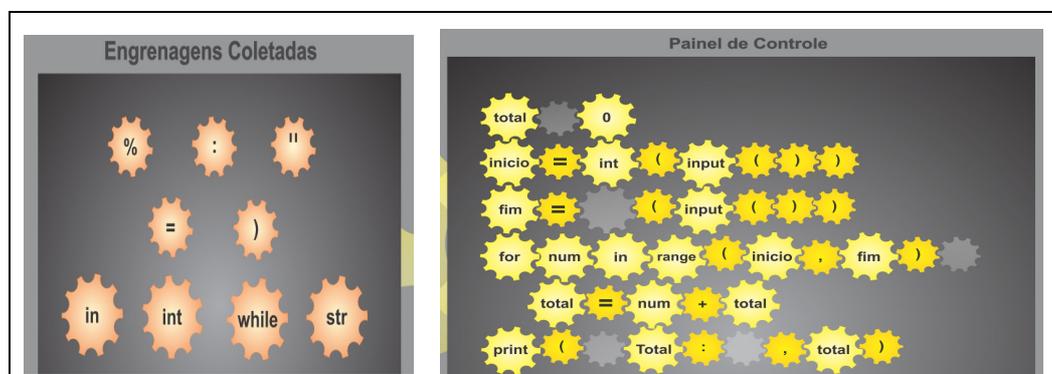
Para dar início ao desenvolvimento do jogo, foi montada uma equipe envolvendo alunos dos cursos de Design e Computação, de modo que cada um contribuísse com suas respectivas aptidões. A linguagem escolhida como objeto pedagógico foi Python, uma vez que essa é a linguagem adotada nas disciplinas iniciais de programação dos cursos oferecidos no campus.

O cenário proposto para a primeira fase foi o de um bar em que os robôs estivessem se divertindo. Além de se esquivar dos obstáculos, o herói precisa escapar dos ataques dos robôs inimigos e coletar as engrenagens perdidas, necessárias para completar o desafio de programação da fase. A Figura 2 apresenta algumas cenas dessa fase inicial.



**Figura 2 – Cenário da Fase Balada do Jogo Robotimov**

No final da fase, o jogador se depara com um painel de controle que gerencia todo o funcionamento dos robôs do ambiente. Nesse momento, ele poderá visualizar as engrenagens coletadas ao longo da fase e o código incompleto que precisa ser solucionado através do encaixe das engrenagens adequadas (Figura 3). Os tamanhos e cores diferenciados das engrenagens fornecem dicas para o jogador, e ajudam a consolidar o conhecimento sobre os elementos válidos da sintaxe Python.



**Figura 3 – Desafio de programação da fase Balada do Jogo Robotimov**

O protótipo do jogo está sendo desenvolvido usando a engine Unity, e o principal desafio encontrado pela equipe tem sido a inserção de desafios de programação nas fases de forma dinâmica, de modo que, a cada partida, o jogador seja surpreendido com um novo código.

## 5. Considerações Finais

Dentre tantas contribuições já feitas para a área de Ensino de Programação, espera-se que o Robotimov possa cumprir seus objetivos pedagógicos e lúdicos e tornar-se um valioso instrumento para apoiar o aprendizado da linguagem Python.

A proposta de uma variedade de desafios de programação para atender todas as etapas cognitivas do aprendizado, integrados diretamente com o Game Design da aplicação, evidencia uma importante iniciativa para motivar os alunos a praticarem seus conhecimentos de forma lúdica e divertida continuamente enquanto aprendem.

## Referências Bibliográficas

- ALICE. (2010) “Projeto Alice”, <http://www.alice.org>.
- AURELIANO, V. C. O.; TEDESCO, P. C. A. R., Ensino-aprendizagem de Programação para Iniciantes: uma Revisão. Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012), ISSN 2316-6533. Rio de Janeiro, 26-30 de Novembro. 2012.
- BARBOSA, L. S.; FERNANDES, T. C. B; Campos, A. M. C., Takkou: Uma Ferramenta Proposta ao Ensino de Algoritmos. WEI – XIX Workshop sobre Educação em Computação, XXXI Congresso da Sociedade Brasileira de Computação. Natal-RN, 19 a 22 jul. 2011.
- CODE SPELLS, (2013) “Projeto Code Spells” <https://sites.google.com/a/eng.ucsd.edu/codespells/home/overview>.
- DANTAS, V.; FREITAS, P.; ALENCAR, L., ProGame: Um jogo para apoiar o ensino aprendizagem de programação. In: First Workshop on Applications to Provide Learning and Teaching Support (APPLETS). Anais do XXII SBIE – XVII WIE. Aracaju, SE. 2011.
- DE JESUS, E. A.; RAABE, A. L. A., Interpretações da Taxonomia de Bloom no Contexto da Programação Introdutória. In: XX Simpósio Brasileiro de Informática na Educação. Anais do XX Simpósio Brasileiro de Informática na Educação. v. 1. Florianópolis, SC. 2009.
- DELGADO, C.; XEXEO, J. A. M.; SOUZA, I. F.; CAMPOS, M.; RAPKIEWICZ, C. E., “Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos”. *XII Workshop de Educação em Computação (WEI'2004)*. Salvador, BA, Brasil. 2004.
- DIM, C. A.; ROCHA, F. E. L. da., APIN: *Uma Ferramenta Para Aprendizagem de Lógicas e Estímulo do Raciocínio e da Habilidade de Resolução de Problemas em um Contexto Computacional no Ensino Médio*. In: XIX Workshop sobre Educação em Computação. Anais do XXI CSBC 2011. Natal, RN. 2011.
- EAGLE, M.; BARNES, T., Wu's castle: teaching arrays and loops in a game. *Proceeding ITiCSE '08 Proceedings of the 13<sup>th</sup> annual conference on Innovation and technology in computer science education*. 2008
- HINTERHOLZ Jr, O. (2009). Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. In: XVII WEI, Bento Gonçalves/RS.
- MARAGOS, K.; GRIGORIADOU, M., Designing an Educational Online Multiplayer Game for learning Programming. *Proceedings of the Informatics Education Europe II Conference*. IEEEI 2007.
- MORATORI, P., Por que utilizar jogos educacionais no processo de ensino aprendizagem? Rio de Janeiro 2003
- NETO, W. C. B.; SCHUVARTZ, A. A., Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores. In: 18º Simpósio Brasileiro de Informática na Educação. São Paulo, Brasil. 2007.
- PRIETCH, S. S. ; PAZETO, T. A. . Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias. In: Workshop de Educação em Informática Bahia-Alagoas-Sergipe (WEIBASE), Macieó/AL, 2010.
- RAPKIEWICZ, C. E.; FALKEMBACH, G.; SEIXAS, L.; ROSA, N. S.; CUNHA, V. V.; KLEMMANN, M., Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao uso de Jogos Educacionais. 2006.
- SANTOS, A.; HAMERSKI, Jr. E., Robocode: Uma maneira simples e divertida de aprender java. *Java Magazine*, 1 (3):43–45, 2002.
- SCAICO, P.; MARQUES, D. L.; MELO, L. A.; AZEVEDO, M. A.; NETO, S.V.M.; OLIVEIRA, A.; JÚNIOR, J. A.; LABANCA, M.; SCAICO, A., Um jogo para o ensino de programação em Python baseado na taxonomia de Bloom. XXXII Congresso da Sociedade Brasileira de Computação. Curitiba, PR. 2012.
- TAROUCO, L. (2005) “Jogos educativos via WWW”, disponível em <http://penta3.ufrgs.br/animacoes/JogosEducacionais>.