

ProGame: um jogo para o ensino de algoritmos e programação

Chrystian Gesteira Sales, Vanessa Farias Dantas

Instituto de Ensino Superior da Paraíba (IESP)

Universidade Federal da Paraíba (CCAUE-UFPA) – Campus IV

João Pessoa – PB – Brasil

{chrystian.gesteira@gmail.com, vanessa@dce.ufpb.br}

***Abstract:** This study aimed to create a prototype of an educational game to support algorithms and programming teaching and learning processes. Based on identification of the challenges involved, the game was developed to encourage skills like logical reasoning and problem resolution.*

***Resumo:** Este trabalho teve como objetivo a criação de um protótipo de jogo educacional para dar suporte ao ensino/aprendizado de algoritmos e programação. Com base na identificação dos desafios envolvidos, o jogo foi desenvolvido para estimular habilidades como raciocínio lógico e resolução de problemas.*

1. Introdução

A disciplina de Introdução à Programação, também conhecida como Algoritmos, Laboratório de Programação ou Lógica de Programação, trabalha, basicamente, com o ensino de algoritmos e linguagens de programação em cursos das áreas de Computação e Engenharia. Esta disciplina geralmente é aplicada usando o paradigma de programação estruturado, aliado a linguagens de programação como C e Pascal, e é base para alguns dos conhecimentos a serem adquiridos no decorrer dos cursos.

Entretanto, o índice de evasão e reprovação dos alunos nessa disciplina costuma ser bastante alto. Rocha (1993) explica que os cursos de Lógica de Programação iniciam com uma média de 50 alunos e, em poucos meses, constata-se que as taxas de reprovação e/ou desistência chegam a 60%. Apesar de várias iniciativas [Santos (2008); Borges (2000); Rodrigues (2004); Rosa & Rapkiewicz (2007)] já terem sido feitas para reverter esse quadro, esses números continuam se repetindo atualmente, indicando que essa ainda é uma área que merece atenção.

Gomes (2000) afirma que essa taxa de evasão/reprovação dá-se devido a dificuldades encontradas pelo aluno no momento de usar o raciocínio lógico e que esse problema pode ser intensificado por um agente externo, uma vez que esta disciplina requer muito do envolvimento do professor, pois este precisa acompanhar o trabalho dos alunos, mas muitas vezes não consegue acompanhar toda a turma. Assim, alguns desses alunos acabam ficando com dúvidas que, muitas vezes, não são repassadas para o professor, gerando desânimo, evasão e reprovação.

Pode-se inferir, a partir do exposto, que, além da evasão e reprovação causadas pela falta de motivação, um fator como o desânimo é capaz de gerar outros fatores encadeados a ele, tais como: falta de atenção, não realização de exercícios de fixação, entre outros. Estes fatores também são contribuintes para a evasão e reprovação dos alunos que ingressam em cursos de Computação e se matriculam em disciplinas de algoritmos. É preciso então motivar os alunos ao estudo da disciplina, usando ferramentas que proporcionem interatividade e feedback imediato, tornando o aprendizado dinâmico.

2. Motivação

O uso de uma ferramenta computacional no ensino de algoritmos permite aos alunos testarem suas soluções e visualizar os resultados de suas ações, o que é um grande motivador do processo de ensino aprendizagem (SANTIAGO e DAZZI 2003).

De acordo com Falkembach et. al. (2003), foi feita uma análise comparativa entre as ferramentas de apoio ao ensino de algoritmo mais usadas, levando-se em consideração os itens: conteúdo teórico, exemplos, animações, interação, testes, interface, desafios e linguagem de desenvolvimento do ambiente. A partir da análise foi possível perceber que, na aprendizagem de Algoritmos, em ambientes informatizados, podem ser utilizadas duas abordagens: a abordagem ativa e/ou a abordagem passiva.

Na abordagem ativa, o aprendiz pode testar os seus algoritmos, direcionando o seu funcionamento, e pode interagir com as animações (FALKEMBACH et. al., 2003). São exemplos de ferramentas que trabalham com essa abordagem os jogos educativos propostos por Rosa e Rapkiewicz, (2007).

Na abordagem passiva, o aluno só pode assistir à execução de uma série de comandos pré-definidos. Nesse caso, existe uma limitação clara: o aluno fica preso aos exemplos apresentados. Vídeo-aulas e tutoriais são exemplos de recursos que seguem essa abordagem.

Uma vez que o público-alvo das disciplinas de Algoritmos e Programação costuma ser formado majoritariamente por jovens e adultos, a abordagem ativa parece ideal para apresentar o conteúdo de forma estimulante e capaz de despertar curiosidade, prendendo sua atenção.

3. Abordagem

A inclusão do elemento lúdico no aprendizado proporciona ao indivíduo a capacidade de interagir com o mundo em sua volta. Silva (2008) diz que é preciso criar condições para a formação de indivíduos capazes de agir sobre o mundo e transformá-lo. Assim, utilizar um jogo eletrônico para auxiliar no ensino de algoritmos e programação parece uma decisão bastante indicada.

Com base em outras experiências semelhantes já realizadas (ROSA e RAPKIEWICZ, 2007; COUTINHO et. al. 2008), foi iniciado o desenvolvimento do jogo ProGame, cuja concepção gira em torno dos problemas de ensino/aprendizagem de Algoritmos e Programação, mas sem deixar de lado os aspectos lúdicos do ato de jogar, para garantir a motivação dos alunos.

A ferramenta escolhida para o desenvolvimento do jogo foi o Game Maker 7.0 por permitir um desenvolvimento rápido e por possuir recursos que garantem ao jogo uma boa usabilidade, como: áudio, vídeo e acesso em rede. Além disso, o software conta com uma linguagem de script chamada Game Maker Language que permite que cada objeto dentro do jogo seja programado individualmente, adicionando mais maleabilidade ao desenvolvimento.

A definição das habilidades abordadas pelo jogo teve como base a taxonomia de Bloom, uma estrutura conceitual criada a fim de auxiliar a definição dos objetivos de aprendizagem (JESUS e RAABE, 2009). Seguindo essa proposta, foram definidas as habilidades que poderiam ser exploradas no ProGame. As atividades escolhidas para as fases do jogo foram:

a) Montagem de algoritmos – Diante de um conjunto de instruções válidas, espera-se que o aluno reconheça e entenda cada instrução, e consiga definir a sequência correta de instruções

a ser usada para resolver um dado problema. Assim, ele estará reforçando seus conceitos sobre fluxos de execução.

b) Compreensão de mensagens – O aluno deve ser capaz de associar mensagens de erro apresentadas com possíveis problemas de sintaxe ou de lógica existentes em um programa. Nessa etapa, ele aprenderá a interpretar mensagens de erro e compreender as conseqüências de um erro no código.

c) Correção de código – Nessa etapa, o aluno irá exercitar as cinco primeiras categorias da Taxonomia de Bloom. Ele precisará entender e depurar o código, analisar o fluxo de execução do programa, e ainda elaborar as instruções necessárias para que o programa funcione conforme o esperado.

Num estágio mais avançado do jogo, espera-se que o aluno seja desafiado a elaborar um programa do início, sendo responsável não só pela construção das instruções, mas também pela estruturação delas.

4. Aplicação

A idéia principal do jogo gira em torno da criação, edição e manutenção de algoritmos pré-definidos, que testam determinadas habilidades necessárias para o aprendizado da disciplina. A ludicidade se apresenta na forma de níveis nos quais o jogador deve interagir com objetos que fazem parte do mundo da computação, como *bugs* de código, acessórios, bits e trechos de código (instruções). O jogador deve passar pelos desafios e ao chegar ao final do nível ele será confrontado com a necessidade de resolução de um algoritmo que terá ligação com um elemento do nível, como por exemplo: resolver um algoritmo para abrir a porta de uma sala do tesouro.



Figura 1: Jogo “ProGame”

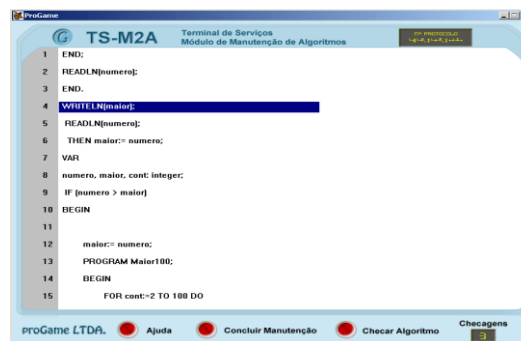


Figura 2: Resolução de Algoritmo

Na versão inicial do jogo, o professor poderá optar pela linguagem e pelos comandos que julgar mais adequados à realidade de seus alunos para editar os programas a serem verificados. Uma próxima etapa do projeto prevê a integração do jogo com diferentes compiladores para que o jogador possa interagir melhor com a linguagem que escolher.

5. Conclusão e trabalhos futuros

As dificuldades de aprendizado dos alunos em relação à disciplina de Algoritmos e Programação são um ponto crítico no ensino dos cursos da área de Computação e Engenharia. Os alunos têm dificuldades em entender o código devido a fatores como falta de feedback, inibição para fazer questionamentos e/ou falta de motivação. Atualmente, os jogos eletrônicos têm sido uma solução adotada para ensinar diferentes conteúdos, como lógica matemática, estruturas de dados e raciocínio lógico. Entretanto, os professores de Algoritmos

e Programação ainda carecem de um jogo específico para apoiar o aprendizado de linguagens de programação procedurais.

A partir da análise da Taxonomia de Bloom, que propõe a divisão dos objetivos do domínio cognitivo em categorias e verbos que dizem respeito a estas categorias, foi possível conceber um jogo de aventura educativo, com etapas que exercitam as diferentes competências necessárias para um bom programador.

Atualmente, o jogo está em fase de homologação, sendo testado por alunos das disciplinas de Algoritmos e Programação em diferentes instituições de ensino. Espera-se que as opiniões coletadas possam servir para melhorar a qualidade do jogo e para identificar novos aspectos da aprendizagem que possam ser trabalhados pela aplicação.

Referências

BORGES, M. Avaliação de uma metodologia alternativa para a aprendizagem de programação. VIII Workshop de Educação em Computação – WEI 2000. Curitiba, 2000. Homepage:

<http://200.169.53.89/download/CD%20congressos/2000/SBC%202000/eventos/wei/wei006.pdf>

FALKEMBACH, G. A. M.; AMORETTI, M. S. M.; TAROUCO, L. R.; Aprendizagem de Algoritmos: Uso da estratégia ascendente de resolução de problemas. Chile:TISE, 2003

GOMES, A. J. Ambiente de suporte à aprendizagem de conceitos básicos de programação, Dissertação (Mestrado)–Universidade de Coimbra, 2000.

JESUS, E. A. de; RAABE, A. L. A. Interpretações da taxonomia de Bloom no contexto da programação introdutória. Florianópolis: SBIE, 2009.

ROCHA, H. V.; Representações Computacionais Auxiliares ao Entendimento de Conceitos de Programação. In: “Computadores e Conhecimento: Repensando a Educação”. Livro organizado por Valente, J. A. Editora Unicamp. 1993.

SANTIAGO, R.; DAZZI, R. L. S. Ferramentas que auxiliam o desenvolvimento da lógica de programação. Blumenau: FURB, 2003.

SILVA, F. M.; COSTA, F. P. D.; SANTOS, C. L. Concepção e realização de um jogo educativo no contexto da aprendizagem colaborativa. Belo Horizonte: SBGames, 2008.

SANTOS, R. et al. Uma proposta de cenário para ensino de algoritmos e programação com contribuições de cooperação, colaboração e coordenação. XVI Workshop sobre Educação em Computação, Belém. Anais do XXVIII Congresso da Sociedade Brasileira de Computação, 2008. v. 1. p. 218-227. Homepage:

<http://www.prodepa.gov.br/sbc2008/anais/pdf/arq0197.pdf>

RODRIGUES, M. Experiências positivas para o ensino de algoritmos. IV Escola Regional de Computação Bahia-Sergipe. Feira de Santana, 2004. Homepage:

<http://www.uefs.br/erbase2004/documentos/weibase/Weibase2004Artigo001.pdf>

ROSA, N.; RAPKIEWICZ, C. Ensinando princípios básicos de programação utilizando jogos educativos em um programa de inclusão digital. VI Brazilian Symposium on Computer Games and Digital Entertainment. São Leopoldo, 2007. Homepage:

<http://www.inf.unisinos.br/~sbgames/anais/shortpapers/35482.pdf>