

Uma visão do cenário Nacional do Ensino de Algoritmos e Programação: uma proposta baseada no Paradigma de Programação Visual

Romenig da Silva Ribeiro¹, Leônidas de O. Brandão¹, Anarosa A. F. Brandão²

¹Instituto de Matemática e Estatística – Universidade de São Paulo

²Escola Politécnica – Universidade de São Paulo

¹{romenig, leo}@ime.usp.br, ²anarosa.brandao@poli.usp.br

Abstract. This paper presents an overview about teaching and learning algorithms and programming considering the publications at the congresses SBIE, WIE e WEI to show how software development has been addressed in such area along recent years. Moreover, it points some difficulties faced by learners that were mentioned within papers published at these events. It also presents an approach based on Visual Programming to mitigate some of these difficulties, using iVprog, indicating that it is a good environment to overcome some difficulties.

Resumo. Este trabalho apresenta uma visão geral sobre o ensino-aprendizagem de algoritmos e de programação, considerando as publicações nos eventos SBIE, WIE e WEI, a fim de mostrar as linhas que os desenvolvedores de software tem adotado ao abordarem a questão. Este trabalho também cita algumas das dificuldades enfrentadas pelos aprendizes desse tópico, de acordo com os autores dos artigos publicados nos três eventos. O trabalho ainda apresenta dois casos de uso do sistema iVprog, que emprega o conceito de Programação Visual, indicando que este é um bom ambiente para contornar algumas dificuldades.

1.Introdução

Segundo Derek Sleeman (1986), muitos autores de sua época classificavam a programação como o novo latim do currículo escolar. Assim como Sleeman percebeu nos anos 80, estudar programação ainda não tem se mostrado uma tarefa simples para novatos. O jornal Folha de São Paulo, ao analisar os dados do Ministério da Educação, divulgou a alarmante taxa de 28%, em média, de evasão nos cursos de Matemática e Ciência da Computação [Folha de São Paulo 2009]. Existem diversos gatilhos para provocar a evasão, dentre eles as dificuldades encontradas pelos alunos nos cursos de Introdução a Computação, sejam elas relacionadas à lógica de programação ou à sintaxe da linguagem de programação associada [Mota et al. 2009].

A partir da categorização de pesquisa relacionada ao ensino de algoritmos e fundamentos de programação proposta por Júnior e Rapkiewicz (2004), são analisados

artigos publicados no SBIE, WIE e WEI entre 2004-2011. Desta análise extrai-se justificativas de uso da programação visual para atenuar dificuldades de aprendizagem relatadas na literatura, seguida de uma análise de uso do iVProg. Ademais, são apresentados os trabalhos futuros e uma possibilidade de pesquisa de Software Educacional na mesma linha da ferramenta.

2. Uma visão geral do Ensino de Algoritmos e Programação

Para dar uma visão geral do ensino de algoritmos e programação foram analisados os artigos completos do SBIE, WIE e WEI de 2004 a 2011, quando disponíveis, dentro do tema “Ensino de Fundamentos de Programação e Algoritmos”, adotando-se as categorias de produções científicas propostas por Júnior e Rapkiewicz (2004), a saber: i) ferramentas, ii) estratégias e iii) ferramentas associadas a estratégias.

Na categoria (i), as ferramentas desenvolvidas estão relacionadas principalmente a Sistemas Tutores Inteligentes (STI) utilizados em ambientes destinados ao aprendizado de programação (e.g. [Gomes et al. 2011]) ou para o gerenciamento de conteúdos das disciplinas (e.g. [Neto e Schuvartz 2007]); Robótica vista como fator motivacional desde o ensino básico (e.g. [Miranda et al. 2007]) até o ensino superior (e.g. [Pio et al. 2006]) e Computação Afetiva destinadas a personalização do nível e tipo de conteúdo a ser estudado (e.g. [Iepsen et al. 2010]) ou para o fornecimento de dicas durante a resolução de um exercício em momentos de frustração (e.g. [Iepsen et al. 2011]).

Na categoria (ii) encontram-se as seguintes tendências para o ensino de programação: Métodos de Avaliação, como o trabalho de Jesus e Raabe (2009) que traz uma interpretação da Taxonomia de Bloom dentro do contexto da Computação introdutória; e Métodos para Motivação como em [Piva Jr. e Freitas 2011].

Na categoria (iii), de ferramentas associadas a métodos, temos trabalhos que vão nas direções de avaliação e acompanhamento do aprendizado como [Neto et al. 2006] e utilização de jogos como motivação [Rebouças et al. 2010].

A partir da análise desses e de outros trabalhos, encontramos apontamentos de dificuldades relacionados a leitura e interpretação dos enunciados e capacidade de abstração das informações (e.g. [Piva Jr. e Freitas 2010]) e dificuldades com a lógica matemática, sintaxe e estruturas das linguagens de programação (e.g. [Mota et al. 2009]), dentre outros. A próxima seção apresenta e justifica o por quê da ferramenta de programação visual utilizada, o iVProg.

3. Uma abordagem com Programação Visual: o iVProg

Entre as dificuldades apontadas na literatura estão aquelas relacionadas a capacidade de abstração e também aquelas ligadas as dificuldades impostas pela sintaxe e estruturas abstratas da linguagem de programação [Mota et al. 2009].

Com a intenção de diminuir a carga cognitiva dos aprendizes em relação a estas dificuldades, o iVProg permite aos alunos construir seus algoritmos interagindo com elementos visuais. Os elementos representam blocos de controle similares a estruturas do tipo “for”, “while”, “if-then-else” e variáveis. Para manipular esses objetos, os

estudantes utilizam o mouse arrastando-os e soltando-os de forma a criar um algoritmo. O iVProg é derivado do Alice [Conway e Pausch 1997], portanto a tecnologia utilizada é o Java. Assim, o iVProg funciona como um Applet e pode ser integrado ao SGC Moodle através do plugin iTarefa [Rodrigues et al. 2010]. Para viabilizar seu uso na web o iVProg não possui os recursos de criação de animações e histórias em 3D presentes no Alice. O volume de código que era de 140Mb do Alice foi reduzido para um pouco menos de 10% do valor inicial. A seguir são apresentados os trabalhos relacionados e a utilização do iVProg num ambiente formal de ensino.

3.1. Trabalhos relacionados

Existem muitas aplicações destinadas ao ensino de programação para principiantes, algumas das quais se assemelham ao iVProg. Dentre elas, cita-se Scratch [Maloney et al. 2010], Alice e pyGoGoBlocos [Barbosa et al. 2009]. Todas as ferramentas citadas utilizam blocos visuais para representarem as estruturas e detalhes das linguagens tradicionais. A manipulação destes objetos também se dá arrastando-os e soltando-os para criar o algoritmo. Tanto o Alice quanto o Scratch, apesar de possuírem recursos para a web, não são integráveis a um Sistema de Gerenciamento de Curso (SGC). Além disso os algoritmos não podem ser editados online. A princípio, as ferramentas Scratch e pyGoGoBlocos foram desenvolvidas para crianças do ensino fundamental, enquanto a ferramenta Alice foi desenvolvida para uso em cursos de Introdução a Computação.

Na seção seguinte apresenta-se a utilização do iVProg em um curso de introdução a computação no primeiro semestre de 2011.

3.2. Utilização do iVProg num ambiente formal de ensino

O iVProg foi utilizado com turmas de licenciatura em Matemática nos primeiros semestres de 2010 (curso II) e 2011 (curso III). Os cursos tiveram duração de 18 semanas e foram ministrados de maneira mista, com atividades presenciais e atividades semanais entregues pelo Moodle. As aulas deram-se num laboratório de informática com dois estudantes por computador. Para efeitos de comparação, apresentaremos os dados de um curso I ministrado em 2005 com uma abordagem de programação não visual, o iCG (<http://www.matematica.br/icg/>). Nos três cursos a linguagem C foi utilizada, sendo que no curso I e II foi utilizada a partir da semana 6 e no curso III foi utilizada desde o começo.

A ideia era verificar se o iVProg poderia melhorar a compreensão da resolução de problemas computacionais ao eliminar a preocupação com a sintaxe de uma linguagem procedimental tradicional. Além disso, era desejável verificar como a utilização do iVProg poderia impactar no aprendizado de uma linguagem procedimental tradicional tal como a linguagem C.

Até a 6ª semana do curso I utilizou-se apenas o iCG no laboratório. As demais semanas deram-se em sala de aula e os exercícios foram resolvidos em C, em casa. No curso II, até a 6ª semana utilizou-se apenas o iVProg e depois o ambiente tradicional de programação em C, sempre no laboratório. No curso III, a linguagem C e o iVProg foram utilizados juntos, desde o começo, resolvendo primeiro no iVProg e depois em C. No total, no curso I foram resolvidos 5 exercícios em laboratório e 3 exercícios bastante

complexos, que exigiam modelagem, em casa. No curso II e III foram resolvidos 20 exercícios usando o iVProg e 33 exercícios usando a linguagem C, em sua maioria no laboratório. Houveram mais exercícios em C porque o iVProg ainda não manipula arquivos ou vetores. As avaliações dos cursos foram baseadas nos exercícios resolvidos (C e não-C) e nas provas 1 (P1) e 2 (P2). A tabela seguinte exhibe os dados referentes a frequência dos alunos nos cursos, notas nas P1 e P2 e notas nas listas de exercícios C e não-C (não-C significa usando o iCG para o curso I e o iVProg para os cursos II e III).

Tabela 1 – Comparação entre os anos de 2005, 2010 e 2011

	Curso I (2005)		Curso II (2010)		Curso III (2011)	
	Média	Variância	Média	Variância	Média	Variância
frequência	56,48%	0,109	72,92%	0,072	71,02%	0,061
P1	3,28	8,3	5,59	8,7	5,85	5,64
P2	4,33	17,02	3,47	4,92	6,36	7,29
não-C	7	20,5	9,26	0,35	9,59	0,65
C	4,28	18,7	8,51	0,43	9,08	0,74
Média final	4,06	10,44	4,59	6,31	5,24	6,19

Pode-se observar um interessante aumento na frequência as aulas dos alunos nos cursos II e III em relação ao curso I. Além disso as médias em provas e exercícios C e não-C foram maiores quando o iVProg foi utilizado, inclusive as médias finais. Esses números nos permite conjecturar que o iVProg teve impacto positivo na motivação e no aprendizado dos alunos durante os cursos II e III, incluindo em como resolver os problemas utilizando uma linguagem de programação tradicional, como a linguagem C.

4. Conclusão e Trabalhos futuros

Este trabalho apresentou uma breve visão do cenário sobre o ensino de programação e algoritmos para iniciantes dentro do contexto dos eventos SBIE, WIE e WEI. Sua principal contribuição é indicar que o modelo de programação visual, como proporcionado pelo iVProg, mostrou-se promissor na tarefa de mitigar dificuldades de aprendizagem de algoritmos. Os resultados apontam que a programação visual realmente auxilia os alunos a se concentrarem no projeto dos algoritmos.

Como trabalhos futuros pretende-se desenvolver um Tutor Inteligente e um Avaliador Automático de exercícios integrados ao *iVProg*. O Avaliador Automático (baseado em heurísticas) pode beneficiar significativamente os alunos que terão uma resposta imediata sobre suas soluções. Mais ainda, como o *iVProg* pode ser utilizado de modo integrado ao ambiente *Moodle*, os professores poderão rapidamente saber como seus alunos estão respondendo aos problemas propostos, o que por exemplo, possibilitará rápida identificação de falhas em enunciados de exercícios. Por fim, vale destacar que a questão da migração entre a programação visual e a dita *programação tradicional*, ainda faz-se necessário mais estudos. Com o iVProg os alunos puderam mais rapidamente encontrar soluções para problemas algorítmicos, entretanto migrar sua solução visual para a linguagem C mostrou-se ainda ser uma tarefa difícil.

5. Agradecimentos

Este trabalho teve apoio parcial de FAPESP (2011/10926-2), CNPq (550449/2011-6) e bolsa de mestrado Capes.

6.Referências

- Barbosa, M. R. G., Silva, F. A., Oliveira, V. M. A., Feltrim, V. D., Mirisola, L. G. B., Gonçalves, P. C. Ramos, J. J. G. e Alves, L. T. (2009) “Implementação de Compilador e Ambiente de Programação Icônica para a Linguagem Logo em um Ambiente de Robótica Pedagógica de Baixo Custo” In: Anais do XX Simpósio Brasileiro de Informática na Educação.
- Conway, M., J. e Pausch, R. (1997) "Alice: Easy to Learn Interactive 3D Graphics", Computer & Graphics, Volume 31, Issue 3, pages 58- 59.
- Folha De São Paulo, (2009) “ Matemática e ciências da computação têm alta taxa de abandono”, <http://www1.folha.uol.com.br/folha/educacao/ult305u546576.shtml> , Agosto.
- Gomes, C. C. C., Lima, D. H. S., Ribeiro, R. P., Almeida, E. S. e Brito, P. H. S. (2011) “Uma Proposta para Auxiliar Alunos e Professores no Ensino de Programação: O Ambiente AIIP” In: Anais do XIX Workshop sobre Educação em Computação (SBC)
- Iepsen, E. F., Bercht, M. e Reategui, E. (2010) “Persona-Algo: Personalização dos Exercícios de Algoritmos auxiliados por um Agente Afetivo” In: Anais do XXI Simpósio Brasileiro de Informática na Educação.
- Iepsen, E. F., Bercht, M. e Reategui, E. (2011) “Detecção e Tratamento do Estado Afetivo Frustração do Aluno na Disciplina de Algoritmos” In: Anais do XXII Simpósio Brasileiro de Informática na Educação.
- Jesus, E.A. e Raabe, A. L. A. (2009) “Interpretações da Taxonomia de Bloom no Contexto da Programação Introdutória” In: Anais do XX Simpósio Brasileiro de Informática na Educação.
- Júnior, J. C. R. P. e Rapkiewicz, C. E. (2004) “O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil” In: Anais do XII Workshop sobre Educação em Computação (SBC).
- Maloney, J., Resnick, M., Rusk, N., Silverman, B. e Eastmond, E. (2010) "The Scratch Programming Language and Environment", ACM Transactions on Computing Education, Volume 10, Issue 4.
- Miranda, L. C., Sampaio, F. F. e Borges, J. A. S. (2007) “ProgrameFácil: Ambiente de Programação Visual para o Kit de Robótica Educacional RoboFácil” In: XVIII Simpósio Brasileiro de Informática na Educação.
- Mota, M. P., Brito, S. R., Moreira, M. P e Favero, E. L. (2009) “Ambiente Integrado à Plataforma Moodle para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação” In: Anais do XX Simpósio Brasileiro de Informática na Educação.
- Neto, W. C. B. e Shuvartz, A. A. (2007) “Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores” In: XVIII Simpósio Brasileiro de Informática na Educação.
- Neto, F. A. A., Castro, T. H. C. e Júnior, A. N. C. (2006) “Utilizando o Método Clínico Piagetiano para Acompanhar a Aprendizagem de Programação” In: Anais do

XVII Simpósio Brasileiro de Informática na Educação.

- Pio, J. L. S., Castro, T. H. C. e Júnior A. N. C. (2006) “A Robótica Móvel como Instrumento de Apoio à Aprendizagem de Computação” In: Anais do XVII Simpósio Brasileiro de Informática na Educação.
- Piva Jr., D. E Freitas, R. L. (2010) “Estratégias para melhorar os processos de Abstração na disciplina de Algoritmos” In: Anais do XXI Simpósio Brasileiro de Informática na Educação.
- Rebouças, A. D. D. S, Marques, D. L., Costa, L. F. S. C. e Silva, M. A. A. (2010) “Aprendendo a Ensinar Programação Combinando Jogos e Python” In: Anais do XXI Simpósio Brasileiro de Informática na Educação.
- Rodrigues, P. A., Brandão, L. O., Brandão, A. A. F., (2010) “Interactive Assignment: a Moodle component to enrich the learning process”. In: Proceedings of Frontiers in Education, páginas T4F-1 - T4F-6.
- Sleeman, D. (1986) “The challenges of teaching computer programming”, In Communication. ACM Volume 29, Issue 9, pages 840–841.