

CONTEXTUALIZAÇÃO DA APRENDIZAGEM DA PROGRAMAÇÃO: ESTUDO EXPLORATÓRIO NO SECOND LIFE®

Micaela Esteves, Ricardo Antunes

*Escola Superior de Tecnologia e Gestão de Leiria, Apartado 4163, 2411-901 Leiria, Portugal
micaela@estg.ipleiria.pt; antunes@estg.ipleiria.pt*

Leonel Morgado, Paulo Martins

*GECAD – Grupo de Engenharia do Conhecimento e Apoio à Decisão
Universidade de Trás-os-Montes e Alto Douro, Dep. Engenharias, Apartado 1013, 5001-801 Vila Real, Portugal
leonelm@utad.pt; pmartins@utad.pt*

Benjamim Fonseca

*CETAV – Centro de Estudos Tecnológicos do Ambiente e da Vida
Universidade de Trás-os-Montes e Alto Douro, Dep. Engenharias, Apartado 1013, 5001-801 Vila Real, Portugal
benjaf@utad.pt*

RESUMO

Aprender a programar é um processo difícil e ser um bom programador exige vários tipos de aptidões. A generalidade dos alunos quando começa a estudar programação sente muitas dificuldades, resultando num elevado índice de reprovações às disciplinas da área. A falta da contextualização da aprendizagem tem sido apontada como uma das razões para este insucesso. Face a esta situação pensamos utilizar o ambiente virtual Second Life para contextualizar a aprendizagem da programação.

Propusemos a dois grupos de alunos a realização de projectos de programação no Second Life, utilizando a linguagem de *scripting* LSL do Second Life, como forma alternativa de aprendizagem, para podermos observar as capacidades deste software no ensino-aprendizagem da programação. Neste artigo, damos conta de até que ponto este ambiente virtual permitiu motivar e despertar nos alunos o interesse pela aprendizagem da programação. Apresentamos ainda alguns aspectos das observações realizadas, em termos das dificuldades por parte de alunos e docentes, quer no desenvolvimento dos projectos, quer no acompanhar das actividades durante as aulas.

PALAVRAS-CHAVE

Ambientes virtuais colaborativos; aprendizagem da programação; mundos virtuais; MMOG; contexto.

1. INTRODUÇÃO

Aprender a programar é um processo difícil e exigente. Para se tornarem bons programadores os alunos têm de adquirir um conjunto de conhecimentos que vão muito além de conhecer a sintaxe e a semântica de uma linguagem de programação (Gomes, 2000; Takahashi, 1998). Os alunos necessitam de aprender a resolver problemas e elaborar algoritmos, o que dificulta a aprendizagem da programação. De facto, muitos alunos sentem dificuldades na aprendizagem da programação, o que resulta num elevado índice de insucesso às disciplinas de programação que surgem, na generalidade, no início dos cursos de Engenharia Informática e áreas afins. Na literatura encontram-se referências a vários estudos cujo objectivo é investigar as causas deste insucesso (Gray, 1993; Jenkins, 2002; Lahtinen *et al.*, 2005). Uma das causas apontadas é a falta de contextualização do conteúdo (Figueiredo, 2006). É necessário considerar a importância de se criar contextos de aprendizagem plenos de actividades, de interações e de ambientes sociais culturalmente ricos (Figueiredo, 2000).

Tendo em conta estas considerações, pensámos que a utilização do mundos virtual tridimensionais Second Life (SL), poderia contextualizar a aprendizagem da programação. O SL é um mundo com uma forte componente visual onde os utilizadores podem visualizar directamente os objectos por eles construídos e os resultados visuais dos programas a eles associados, idealizado por Philip Rosedale em 1991 e disponibilizado publicamente desde 2003, gerido pela empresa Linden Research Inc. (Rymaszewski *et al.*, 2007). É um mundo social e culturalmente rico onde os utilizadores são participantes activos dispostos a ajudar os utilizadores menos experientes e a partilhar os conhecimentos adquiridos (Bettencourt & Abade, 2007).

Para explorar a viabilidade destes ambientes, levámos a cabo durante o segundo semestre do ano lectivo 2006/2007 actividades lectivas de programação com alunos do ensino superior. Sob nossa coordenação e acompanhamento, desenvolveram-se no SL em conjunto com os alunos actividades de ensino-aprendizagem de programação (*vd.* secção 3.1), em lugar de actividades em ambientes mais clássicos, como C, C++ ou C#.

Na próxima secção descrevemos mais detalhadamente a importância da contextualização da aprendizagem. Na terceira secção expomos a experiência realizada assim como a análise dos resultados obtidos. Por último apresentamos as conclusões.

2. CONTEXTUALIZAÇÃO

“Parte da actual crise da aprendizagem deve-se ao choque entre uma visão mecanicista da educação e uma civilização cada vez mais relacional e de contextos.” (Figueiredo & Afonso, 2006)

O recurso inteligente a novos *media* torna possível a contextualização do ensino-aprendizagem, a construção de saberes pelos próprios aprendentes, em ambientes activos e culturalmente ricos, ambientes que raramente existem no contexto escolar. Por exemplo, um dos alunos que desenvolveu o projecto de uma equipa de robôs no SL, descrito na secção 4, ao passear com os seus robôs por esse mundo foi abordado por outro utilizador que se interessou pelo trabalho e quis comprar os robôs. Eis uma situação impensável de acontecer quando se aprende a programar nos ambientes mais clássicos e que para o aluno é um estímulo, pois demonstra-lhe o reconhecimento e interesse externo do seu trabalho.

O SL, além de ambiente virtual tridimensional on-line, apresenta também a característica de permitir ter vários utilizadores em simultâneo no mesmo espaço que podem interagir e colaborar. Neste mundo, os alunos têm a oportunidade de aprender fazendo e ver imediatamente o resultado da sua programação, por exemplo um aluno que esteja a aprender as estruturas de repetição pode criar uma bola e desenvolver um programa que a faça mudar de cor a cada iteração do ciclo, isto é feito em poucos segundos, com um esforço mínimo, o mesmo já não se pode dizer se utilizasse um ambiente clássico. Esta transparência na representação do conhecimento permite ao aprendiz ter uma experiência não simbólica e na primeira pessoa de alguns conceitos, assim como a aquisição de saberes com aplicação visível.

3. ACTIVIDADE DESENVOLVIDA

Estas actividades exploratórias decorreram ao longo do segundo semestre do ano lectivo 2006/2007 e envolveram 9 alunos da Licenciatura em Informática da Universidade de Trás-os-Montes e Alto Douro (UTAD), nas disciplinas Laboratório I, do 1º ano curricular (LabI) e Laboratório III, do 2º ano (LabIII).

Os alunos puderam desenvolver livremente os projectos no horário e local que acharam mais adequado. No entanto, professores e alunos reuniam-se dentro do SL uma vez por semana, durante duas horas, para acompanhar o desenvolvimento dos projectos e trocarem ideias e sugestões.

3.1 Caracterização dos Alunos

Os alunos de LabI estavam numa fase muito inicial do estudo da programação: a disciplina de LabI funciona em paralelo com a disciplina de Metodologias de Programação I, a primeira do currículo integralmente dedicada à programação, embora no semestre anterior, aspectos introdutórios da programação sejam abordados em cerca de 30% de duas disciplinas (*vd.* tabela 1). Para estes alunos, o projecto em LSL constituiu o primeiro exercício de elaboração de um projecto de programação próprio.

Os alunos de LabIII estavam numa fase mais avançada da aprendizagem da programação: além do contacto com a programação de projectos no ano curricular anterior (em C), aprenderam no 1º semestre do ano 2006/2007 conceitos de programação orientada a objectos em C++ e nesse mesmo semestre desenvolveram um segundo projecto prático (em C++, *vd.* tabela 1). Estavam numa fase que, embora inicial, visava já a autonomização do uso e estudo da programação por parte dos alunos.

Tabela 1. N.º de alunos envolvidos na experiência e respectivas linguagens de programação já estudadas ou em estudo.

Disciplinas	Ano curricular	Número de alunos envolvidos	Linguagens de programação anteriormente estudadas	Linguagens de programação em estudo
LabI	1º	5		C
LabIII	2º	4	C, C++	C#

3.2 Projectos Propostos

Os projectos foram definidos previamente por escrito pelos professores e consistiram no desenvolvimento de objectos (e ligação destes para formar objectos mais complexos) utilizando o editor visual do SL, e na programação do comportamento desses objectos em LSL, a linguagem de *scripting* do SL.

O projecto atribuído aos alunos de LabI consistiu na criação de um cão que recebesse ordens do dono e as executasse, devendo também segui-lo, comportamentos que os alunos deveriam programar e atribuir ao objecto criado. Neste projecto os alunos colocam em prática conceitos de variável, de estruturas de decisão e de repetição, de funções, de evento, de estado e de comunicação entre objectos.

Os alunos de LabIII foram divididos em dois grupos, cada qual com um projecto diferente. Num caso, consistiu na criação de 4 robôs; noutra, de um comboio e respectivas estações. Em ambos os casos, pretendeu-se que os alunos desenvolvessem autonomamente um projecto mais complexo que o de LabI. Consoante as mensagens recebidas, cada objecto devia comportar-se de determinada forma. Pretendeu-se também que houvesse envio de mensagens para fora do SL através de e-mails. Os alunos teriam também de rever conceitos básicos de programação, em particularmente explorando a técnica de canais de comunicação, mas também o uso de temporizadores e de sensores virtuais, e a manipulação de listas de dados.

3.3 Análise das Observações

Os alunos, quer de LabI, quer de LabIII, sentiram algumas dificuldades na utilização do editor do SL. Por exemplo: como ligar objectos entre si, criar cópias de objectos ou alinhar objectos entre si. Os alunos de LabI, sem hábitos de desenvolvimento autónomo de programas de computador, necessitaram de um acompanhamento mais intenso da parte dos professores. Acabámos por ter de partir de exemplos simples que os alunos experimentaram, tentaram modificar de acordo com os objectivos dos seus projectos e, em caso de dificuldade, dar uma explicação sobre o que cada parte do código fazia. Observou-se que os alunos conseguiram compreender o que estes pequenos programas faziam e qual o objectivo de cada um. Quer através da nossa experiência pessoal como docentes de programação, quer através da literatura da área (*e.g.*, Mendes *et al.*, 2005), sabemos que esta situação é difícil de alcançar quando o aluno está a aprender a programar em ambientes tradicionais, como por exemplo compiladores de C para linha de comandos, em que os alunos geralmente sentem grande dificuldade em perceber o objectivo da programação.

Os alunos de LabIII não tiveram grandes dificuldades em perceber como funciona o LSL. Embora já tivessem trabalhado com eventos, nunca tinham estudado nem conheciam o conceito de máquina de estados, nem contacto com programação segundo este conceito. A maior dificuldade sentida por estes alunos, contudo, consistiu na selecção das funções a utilizar para uma determinada funcionalidade. Neste caso, o papel dos professores foi orientar e mostrar outras formas de criar o mesmo comportamento nos objectos, de modo a que o aluno reflectisse sobre qual seria a mais adequada.

Um aspecto particularmente importante no ensino da programação é a reacção dos alunos aos erros de compilação (Gomes, 2000; Esteves, 2004), inevitáveis para quem está a aprender. Neste aspecto, os alunos de LabIII conseguiram frequentemente corrigi-los sem a ajuda dos professores, enquanto que os de LabI ficaram muitas vezes sem saber o porquê do erro e sem saber como corrigi-lo.

Quando os alunos tinham alguma dificuldade em relação ao código que estavam a desenvolver, partilhavam o código com os professores e estes observavam-no, viam o que estava mal e davam indicações para que os alunos o conseguissem corrigir.

Já os erros de execução ocorriam mais vezes aos alunos de LabIII, que por sua iniciativa testavam mais programas e constatavam mais frequentemente que estes não funcionavam como esperavam. Não se observou que os alunos ficassem desmotivados por isto acontecer, antes pelo contrário: corrigiam os programas, até que os objectos em questão fizessem o pretendido.

3.4 Resultado das Observações

A primeira dificuldade sentida no acompanhamento dos alunos foi gerir a comunicação. No SL esta é feita via *chat* textual; quando estávamos numa aula com vários alunos, com pessoas a escrever em simultâneo, tornava-se difícil ler e responder a várias solicitações em simultâneo. O processo adoptado foi solicitar que nas mensagens se incluisse sempre o nome do aluno para o qual a mensagem se dirigia. Desta forma, cada mensagem tinha sempre a identificação do autor, colocada pelo SL, mas também do receptor da mensagem, evitando-se ambiguidade sobre o destinatário.

À medida que os alunos desenvolviam as actividades em aula, constatou-se que quando o professor estava a tirar dúvidas a um aluno (ou seja, focado num aluno ou grupo), perdia a noção do que os outros estavam a fazer, das dificuldades que estavam a ter, se tinham necessitado de alguma explicação, etc. Sentiu-se a necessidade de algum processo de chamada de atenção, sem ser por *chat*, para que o professor se aperceba visualmente de que um aluno ou grupo também precisa de ajuda (ainda não implementámos tal processo).

Outra dificuldade foi a disposição espacial dos grupos de alunos no mundo virtual. Cada grupo tem de estar disperso para poder construir objectos e trocar ideias entre si, pelo que os professores têm de se deslocar periodicamente para o espaço específico no SL onde cada trabalho está a ser desenvolvido.

Fora das sessões de acompanhamento semanal, os alunos iam desenvolvendo os trabalhos; na aula seguinte expunham aos professores as dificuldades sentidas e a evolução. Para melhor acompanhamento, seria útil aos professores ter um mecanismo que informe, via e-mail ou outro sistema exterior ao SL, o que os alunos fizeram ao longo da semana, as dificuldades sentidas e as tentativas efectuadas para as ultrapassar.

Para os docentes testarem os trabalhos, o SL permite a partilha de objectos e *scripts* através de um conjunto de permissões que devem ser dadas a cada trabalho. Constatámos a dificuldade de gerir a atribuição destes privilégios pelos docentes, levando a situações que impedem resolução imediata (por ex., um aluno que envia ao professor um objecto sem lhe conferir permissões de acesso aos *scripts*).

4. CONCLUSÕES

As observações indiciam que a utilização do Second Life estimula o interesse dos alunos e motiva-os. Conseguimos também identificar algumas características importantes para o processo ensino-aprendizagem da programação neste ambiente. Achamos relevante efectuar mais estudos e experiências, para determinar de que forma se podem utilizar e/ou adequar este de tipo de ambiente virtual ao ensino-aprendizagem da programação. Foi notória a necessidade de desenvolvimento de diversos mecanismos tecnológicos de suporte ao desenvolvimento de actividades.

AGRADECIMENTOS

Agradecemos à Universidade de Trás-os-Montes e Alto Douro e aos professores de Laboratório I, Laboratório III da mesma Universidade por nos terem proporcionado as condições que viabilizaram a realização destas actividades de investigação. Agradecemos ainda aos alunos que participaram nesta experiência por prontamente terem aceite o desafio que lhes foi proposto num ambiente que lhes era desconhecido.

REFERÊNCIAS

- Bettencourt, Teresa; Abade, Augusto (2007). *Reflexões sobre ensinar e aprender na Second Life – um estudo preliminar*. Comunicação apresentada na workshop cef[^]sl – Comunicação, Educação e Formação no Second Life, Universidade de Aveiro, 23-25 de Maio de 2007. Disponível on-line (consulta a 21 de Julho de 2007) em http://cleobekkers.files.wordpress.com/2007/06/apresenta_1workshop_aveiro24maio07.pdf
- Centre for Research on Networked Learning and Knowledge Building (s.d), Development of Learning Theories, online “<http://www.helsinki.fi/science/networkedlearning/eng/delete.html>”, consultado em 06/06/2007.
- Esteves, M. e Mendes, A.J., A Simulation Tool to Help Learning of Object Oriented Programming Basics, in *Proceeding of 34th Frontiers in Education Conference*, Savannah, Estados Unidos, Outubro de 2004.
- Figueiredo, A. D. and Afonso, A. P., 2006. *Managing Learning in Virtual Settings: the Role of Context*, Information Science Publishing,
- Figueiredo, A. D. (2000). Web-Based Learning – Largely beyond content. In Fancisco Restivo e Lúgia Ribeiro (eds.), *WBLE 2000 Web-Based learning environments*. Porto :FEUP edições, pp. 85-88.
- Gray, W. D.; Goldberg, N.C.; Byrnes, S. A. (1993), Novices and programming: Merely a difficult subject (why?) or a means to mastering metacognitive skills? Review of the book *Studying the Novice Programmer*, *Journal of Educational Research on Computers*, pp 131-140.
- Gomes, A. J. e Mendes, A. J.(2000). Suporte à aprendizagem da programação com o ambiente SICAS, in *RIBIE*.
- Jenkins, T.(2002) On the difficulty of learning to program, in *Proceedings of the 3rd annual conference of the LTSN centre for information and computer science*, Loughborough, United Kingdom, August 27-29, 2002.
- Lahtinen, E., Mutka, K. A., and Jarvinen, H. M (2005). A Study of the difficulties of novice programmers. In *Proceedings of the 10th annual SIGSCE conference on Innovation and technology in computer science education (ITICSE 2005)*, (Monte da Caparica, Portugal, June 27-29, 2005). ACM Press, New York, NY, pp. 14-18
- Mendes, A.J., Esteves, E., Gomes, A., Marcelino, M., Bravo, C. e Redondo, M. (2005). Using Simulation and Collaboration in CS1 and CS2, *The Tenth ACM Annual Conference on Innovation and Technology in Computer Science Education*, Costa da Caparica.
- Rymaszewski, M. et al., 2007. *Second Life: The Official Guide*, Wiley: New Jersey.
- Takahashi, T. (1998). *Introdução a Programação Orientada aos Objectos*. III EBAI (Escola Brasileiro-Argentina de Informática). Curitiba, Janeiro