

Uso do Second Life em Comunidade de Prática de Programação

Micaela Esteves

Escola Superior de Tecnologia e Gestão de Leiria;

micaela@estg.ipleiria.pt

Benjamim Fonseca

CETAV – Centro de Estudos Tecnológicos do Ambiente e da Vida,

Universidade de Trás-os-Montes e Alto Douro, Vila Real;

benja@utad.pt

Leonel Morgado

CETAV GECAD – Grupo de Engenharia do Conhecimento e Apoio à Decisão,

Universidade de Trás-os-Montes e Alto Douro, Vila Real;

leonelm@utad.pt

Paulo Martins

CETAV GECAD – Grupo de Engenharia do Conhecimento e Apoio à Decisão,

Universidade de Trás-os-Montes e Alto Douro, Vila Real;

pmartins@utad.pt

Resumo

Neste artigo apresenta-se uma nova abordagem para o ensino e aprendizagem da programação, no ambiente virtual Second Life, no qual foi criado uma comunidade de prática de programação. Propôs-se aos nossos alunos a realização de projectos de programação inseridos nesta comunidade, utilizando a linguagem de scripting Linden Scripting Language (LSL) do Second Life, como forma alternativa de aprendizagem.

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 de programação. Neste artigo, dá-se conta de até que ponto este ambiente virtual permitiu motivar e despertar nos alunos o interesse pela aprendizagem da programação, contrariamente ao que acontece no ensino tradicional. Apresenta-se ainda alguns aspectos das observações realizadas, em termos das dificuldades sentidas por parte dos alunos e docentes, no desenvolvimento dos projectos, quer no acompanhamento das actividades.

Abstract

In this paper, it is presented a new way for teaching and learning computer programming. In three-dimensional virtual world Second Life was developed a programming community of practices, with the aim of teaching/learning how to program a computer. We proposed to our students to construct their own programming projects inserted in this community, using Linden Scripting Language (LSL) of Second Life as an alternative way of learning.

The learning of programming is a difficult process. To become good programmers, students have to acquire several skills far beyond the syntax and semantics of the underlying programming language. In fact, many students experience difficulties which result in high levels of failure in introductory programming courses. In this paper, we have realized how this virtual environment allowed the students' motivation and interest in programming learning, rather than in the traditional way. We also show some results of the observations done and the difficulties felt by both students and teachers in the projects development and activities.

Palavras-Chave: Comunidades de Prática, Conhecimento, Ensino-aprendizagem, Programação, Second Life. *Keywords: Practice communities, Knowledge, Teaching-Learning, Programming, Second Life.*

1. Introdução

Os alunos quando iniciam o estudo de uma linguagem de programação sentem muitas dificuldades, que se traduzem num elevado índice de reprovação a estas disciplinas. Apesar dos inúmeros estudos efectuados e das aplicações informáticas desenvolvidas para ajudar e melhorar o ensino e a aprendizagem da programação, o problema persiste. Vários são os factores apontados na literatura para este insucesso, como sejam: a falta de contextualização da aprendizagem; o ensino da programação estar demasiado focalizado na sintaxe da linguagem (Lahtinen, 2006), a falta de raciocínio lógico por parte de alguns alunos para a resolução de problemas (Dann, 2000; Gomes, 2006); e a natureza abstracta dos conceitos

básicos da programação, tais como os conceitos de variáveis e endereço de memória, entre outros (Lahtinen, 2006, Miliszewska, 2007).

A juntar a estes factores temos uma geração de alunos que nasceu e cresceu com os computadores e que não se sentem motivados para aprender a programar, não percebendo a razão pela qual têm de escrever código quando estão habituados a usar programas que com um simples mover do rato lhe gera o código e faz o que precisam. Não sentem motivação para aprender e o estereótipo “passar a noite a programar” sem contacto social faz com que esta geração evite as disciplinas que envolvam programar, assim como saídas profissionais relacionadas com a programação.

Os aspectos mencionados, bem como outros que poderiam ser referidos, originam frequentemente alguma desorientação e desinteresse por parte de muitos alunos, que embora inseridos numa comunidade de estudantes “aprendem isolados e em solidão são avaliados” (Figueiredo, 2006). A visão de ter os alunos como nós de uma rede, onde a construção do conhecimento é uma acto colectivo no qual o aluno, ao construir o seu saber, está a contribuir também, para o saber dos outros (*Ibidem*), levou-nos a utilizar o Second Life, enquanto ambiente virtual multi-utilizador, como plataforma para criarmos uma comunidade de prática para o ensino e aprendizagem da programação. A aplicação prática dos conhecimentos adquiridos na comunidade, a reflexão e a troca de conhecimentos são algumas das estratégias sugeridas por Fleury (2002) e Dillenbourg (2000).

Neste artigo expõem-se os resultados observados ao longo de dois anos da utilização do Second Life como plataforma de ensino e aprendizagem da programação, na qual os alunos estão inseridos numa comunidade de prática de programação. Na próxima secção dá-se uma panorâmica geral do que é uma comunidade de prática. Na segunda secção expõem-se a experiência realizada, assim como a análise dos resultados obtidos. Por último apresentam-se as conclusões.

2. Comunidades de Prática

Uma comunidade de prática (CP) é um grupo de pessoas que partilha um interesse e que se junta para criar uma prática em torno desse tópico para aprenderem, compartilharem e gerarem novos conhecimentos. Exemplos de comunidades de prática podem ser consultores que se especializaram em mudança organizacional, investigadores envolvidos na descodificação do ADN, engenheiros envolvidos na perfuração de poços de petróleo. O objectivo de participar nestas comunidades é uma necessidade autêntica de aprender com os

outros membros num ambiente de aprendizagem real, que tem por base a troca de informações de um modo síncrono ou assíncrono. Os encontros podem ser regulares ou não, em locais fixos com agendas formalizadas de trabalho ou virtuais, porém podem reunir-se pessoas que de outra forma dificilmente se encontrariam para aprenderem juntas.

Segundo Wenger (2002), há três elementos na definição de comunidades de prática. Um é o domínio: tem de haver um assunto sobre o qual a comunidade fala. O segundo é a própria comunidade que constitui uma verdadeira “fabrica” de aprendizagem, pois proporciona interações entre as pessoas e permite construir relações entre si em torno do domínio. Uma página na Web não é uma CP. Ou, se houver sessenta gestores que nunca se falam, eles não são uma CP ainda que desempenhem as mesmas funções. Tem de haver uma comunidade, uma partilha e construção de novos conhecimentos. O terceiro é a prática: tem de existir uma prática e não apenas um interesse que as pessoas partilham. Elas aprendem juntas como fazer coisas pelas quais se interessam.

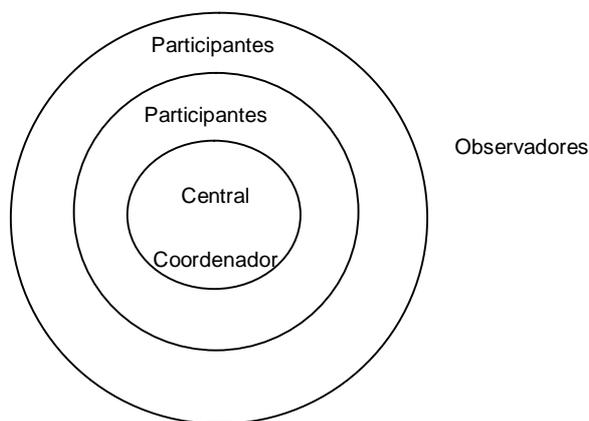
A aprendizagem envolve a participação na comunidade de prática. Essa participação não se refere somente ao envolvimento em certas actividades ou eventos com determinadas pessoas, mas sim um processo que engloba ser um participante activo na prática da comunidade, construindo uma identidade em relação a essa comunidade (Lave & Wenger, 1991). Inicialmente as pessoas juntam-se à comunidade como aprendizes na periferia, mas à medida que se tornam mais competentes movem-se para o “centro” da comunidade. A aprendizagem não é a aquisição de conhecimento individualmente, mas sim um processo de participação social.

Segundo Sílvia (2007) uma comunidade de prática é um óptimo canal para promover a aprendizagem ou boas práticas, não só porque retém o conhecimento numa forma viva e vivida, facilita a resolução de problemas, como pela própria acção da comunidade em que ocorrem os processos de interacção social. Conexões significativas podem conduzir os indivíduos a estágios de criatividade muito maior que poderiam alcançar sozinhos (Lave & Wenger, 1991).

Uma comunidade típica é constituída por diferentes níveis de participantes, como se pode observar na figura 1: central, activo e periférico.

- Central: tem a seu cargo a liderança da comunidade, condução de projectos, lançamento de novos temas e desafios.
- Activo: encontram-se regularmente e têm uma participação efectiva no fórum de discussão.
- Periférico: é composta pelos elementos novos na comunidade que vão observando e aprendendo.

Figura 1 – Exemplo níveis de participantes numa comunidade de prática



Para que uma comunidade de prática tenha sucesso é necessário, segundo Wenger (1991), motivar a participação dos diferentes níveis de participantes, estabelecendo o diálogo entre as perspectivas interna e externa da comunidade, isto é, a participação na comunidade de elementos externos é extremamente útil para o desenvolvimento das suas práticas, bem como a integração da comunidade noutros grupos.

3. Actividade Desenvolvida

O objectivo principal deste estudo é verificar se o Second Life tem características que permita a sua utilização como plataforma para o ensino e aprendizagem da linguagem de programação imperativa. Para isso utilizamos a metodologia investigação-acção. Com este propósito formamos uma comunidade de prática para o ensino e aprendizagem da programação no Second Life e propusemos aos alunos da Licenciatura em Informática da Universidade de Trás-os-Montes e Alto Douro (UTAD) e do Instituto Politécnico de Leiria (ESTG), a realização de um trabalho prático neste ambiente. Estas actividades decorreram ao longo do segundo semestre do ano lectivo 2006/2007 e do primeiro e segundo semestre do corrente ano, inseridas nas disciplinas de Laboratório I, do 1º ano curricular (LabI),

Laboratório de Informática II (Lab II) e III (Lab III), do 2º ano (da UTAD); e Projecto I, (da ESTG).

Aos alunos foi-lhes proposto a realização do trabalho em grupos de dois alunos, inseridos numa comunidade de prática, na qual estes podiam trocar ideias, expor as suas dúvidas e sugestões. Ao longo do semestre os alunos puderam desenvolver livremente os projectos no horário e local que acharam mais adequado. No entanto, professores e alunos tinham reuniões, dentro do SL, uma vez por semana, durante duas horas, para acompanhar o desenvolvimento dos projectos e trocarem ideias e sugestões.

Os trabalhos 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 linguagem LSL.

O mundo virtual Second Life surge-nos como um caso concreto de um mundo virtual, que oferece aos utilizadores a possibilidade de criar conteúdo interactivo sem impor uma narrativa predefinida para as acções. Estas características, num mundo virtual, são particularmente importantes para o domínio educativo, pois permitem aos educadores desenvolver livremente os seus materiais e aos alunos a exploração e interacção, livre ou orientada, podendo expandir o desenvolvimento dos materiais fornecidos ou criar outros de raiz.

O SL tem uma linguagem de *scripting* associada, que permite definir comportamentos para os objectos criados dentro deste mundo virtual. Esta linguagem chama-se Linden Scripting Language (LSL) e é baseada em resposta a eventos, tendo uma sintaxe semelhante às das linguagens C e Java. Como aspecto particular, é baseada no conceito de máquina de estados: cada *script* possui associada a si uma máquina de estados.

Uma característica importante que este mundo possui para o caso do ensino da programação é o facto de possibilitar a partilha de objectos e código entre os elementos de um dado grupo. Deste modo os alunos podem trabalhar colaborativamente.

3.1 Estrutura da comunidade

Os professores são os coordenadores da comunidade e como tal tiveram a seu encargo a definição dos projectos a serem elaborados, incentivar e motivar a participação dos alunos que se encontravam no nível periférico, através da troca de impressões entre os membros da comunidade, assim como a partilha de experiências dos participantes activos que outrora foram periféricos.

Dos primeiros alunos que participaram nesta experiência (2ª semestre 2006/2007) 50% encontravam-se no nível periférico, portanto eram alunos com pouca experiência de programação, e os restantes 50% encontravam-se no nível activo, embora este fosse o seu primeiro contacto com o Second Life, apesar de já terem alguma experiência de programação. Coube aos professores a tarefa de dinamizar e motivar os alunos do nível periférico, para passarem rapidamente para o nível activo. O relato desta experiência pode ser consultado no artigo anteriormente publicado (Esteves, 2007). No semestre seguinte, 80% dos alunos estavam num nível periférico e os restantes num nível activo. É de realçar o facto de os alunos que participaram na comunidade no semestre anterior e tornaram-se membros activos desta, com reconhecimento interno e externo, desempenharam um papel importante na motivação e integração dos novos elementos.

A comunidade tinha reuniões de duas horas por semana no Second Life (SL), nas quais desenvolviam os trabalhos propostos, trocavam ideias e sugestões. Estas reuniões ocorriam dentro do SL num espaço reservado, constituindo reuniões virtuais como se pode observar na figura 2. Os professores encontravam-se com os alunos presencialmente uma vez por mês para trocarem impressões acerca da forma como estava a decorrer a experiência.

Figura 2 – Exemplo de uma sessão da comunidade



3.2 Análise dos resultados

Podem-se demarcar duas fases distintas nestas actividades: a primeira consistiu na construção dos objectos (robôs, comboio, cão); e a segunda no desenvolvimento dos programas em LSL para dar comportamento (vida) aos objectos criados. Durante a primeira fase não se observaram diferenças significativas entre os alunos dos diferentes níveis de participação, tendo as dificuldades sentidas pelos alunos sido idênticas. Por exemplo: como fazer a ligação entre objectos; como criar uma cópia de um objecto; como alinhar objectos entre si.

Na segunda fase já se notaram algumas diferenças entre os alunos. Os alunos do nível activo, 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 tinham tido contacto com programação segundo este conceito. A maior dificuldade sentida por estes alunos consistiu na selecção das funções a utilizar para uma determinada funcionalidade. Neste caso, o papel dos professores era o de 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.

Em relação aos alunos do nível periférico, pouco habituados ao estudo ou desenvolvimento autónomo, têm necessidade de um acompanhamento mais intensivo por parte dos professores. Os professores têm habitualmente de partir de exemplos mais simples, dá-los aos alunos para os testarem, e depois explicar-lhes detalhadamente o código. Observou-se que os alunos conseguiram compreender o que estes pequenos programas fazem e qual o objectivo de cada um. Através da nossa experiência pessoal e da consulta de literatura especializada (Mendes *et al.*, 2005), sabemos que esta situação é difícil de alcançar quando o aluno está a aprender a programar nos 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.

Um aspecto particularmente importante no ensino da programação é a reacção dos alunos aos erros de compilação, inevitáveis para quem está a aprender (Esteves, 2004). Neste aspecto, os alunos de nível activo têm conseguido frequentemente corrigi-los sem a ajuda dos professores, enquanto que os do nível periférico, têm ficado muitas vezes sem saber o porquê do erro e sem saber como corrigi-lo. É através da partilha do programa que os professores vêem o que está errado e dão algumas indicações para que o aluno o consiga corrigir.

Já os erros de execução ocorrem mais vezes aos alunos do nível activo, alunos já com alguma iniciativa, que testam mais programas e constataam mais frequentemente que estes não funcionam como esperavam. Não se tem observado que os alunos fiquem desmotivados por isto acontecer, antes pelo contrário, eles têm vindo a corrigir os programas, até que os objectos em questão façam o pretendido.

Um dos projectos propostos à comunidade baseava-se muito na manipulação de dados e pouco na interacção gráfica, que é um dos pontos fortes do SL. Como resultado, foi bastante difícil conseguir motivar os alunos periféricos a tornarem-se activos, os líderes da comunidade tiveram que recorrer aos alunos activos da comunidade do semestre anterior e a elementos externos para motivarem e dinamizarem a comunidade.

Relativamente a aspectos práticos da intervenção dos professores, como o é o acompanhamento dos alunos, o SL possui um conjunto de mecanismos apropriados para o

efeito, através de partilha de objectos e *scripts* entre alunos, entre professores e alunos. Estes mecanismos baseiam-se num conjunto de permissões que podem ser dadas a utilizadores individuais ou grupos de utilizadores, o que possibilita o desenvolvimento colaborativo dos projectos. Contudo, constatou-se a dificuldade no ambiente SL actual de gerir a atribuição destes privilégios por parte dos docentes, levando a situações que impedem resolução imediata (por ex., um aluno que envia ao professor um objecto mas não confere ao professor permissões de acesso aos *scripts*).

Uma das dificuldades sentidas na implementação da comunidade foi a falta de um quadro no qual os vários membros pudessem escrever as suas ideias e reflexões, de forma a poderem ficar presentes durante as reuniões; um outro factor negativo encontrado foi o facto de não existirem mecanismos que permitam saber o estado de cada fase dos projectos, ou as dificuldades que os alunos estão a sentir e as várias tentativas efectuadas para ultrapassar essas dificuldades. Este último aspecto é particularmente importante quando os alunos desenvolvem o trabalho fora das sessões de acompanhamento semanal.

4. CONCLUSÃO

Neste artigo apresentamos o estudo que fizemos usando a metodologia investigação-acção com o objectivo de explorar a viabilidade de se usar o SL como plataforma de ensino e aprendizagem da linguagem de programação. Este estudo ainda não terminou mas podemos concluir que o SL pode ser usado para o ensino / aprendizagem da programação em conjunto com outra plataforma, exemplo moodle, onde o professor possa disponibilizar os materiais de estudo.

Um inquérito realizado aos alunos que participaram nestas actividades revelou que 80% deles aceitaram participar uma vez que tinham tido uma má experiência na aprendizagem da programação e por isso decidiram tentar uma nova abordagem. A maioria dos alunos (90%) disseram que gostaram desta forma de aprendizagem, que se sentiram motivados principalmente pelo facto de estarem a aprender numa plataforma que permite obter uma resposta visual imediata do código processado. Este facto foi um incentivo para quererem saber mais.

Estes resultados mostram que os alunos para aprenderem é fundamental oferece-lhes oportunidades e mecanismos que suscitem o seu empenhamento. Dois eventos ocorridos fomentaram o empenhamento dos alunos: um dos alunos que desenvolveu o trabalho de uma equipa de robôs no SL, ao passear com os seus robôs por esse mundo foi abordado por outro

utilizador que se interessou pelo trabalho e quis comprá-los; outro facto foi um aluno ter recebido uma proposta para ir trabalhar para uma empresa que desenvolve aplicativos para o SL.

Para além do empenhamento, segundo Figueiredo (2006), as comunidades que criarmos devem deixar espaço para estimular a imaginação porque “quem participa sem sonhar dificilmente será criativo” (*Ibidem*). Nós concordamos inteiramente com o Figueiredo porque ao inserirmos um trabalho na comunidade que pouco partido tirava da interface gráfica do SL, resumindo-se à manipulação de dados, os alunos sentiram-se desmotivados e pouco empenhados, tal como acontece no ensino tradicional. Estes elementos levam-nos a concluir que o SL tem grandes potencialidades para ser usado no ensino da programação mas não podemos cair no erro de o usar como acontece no ensino da programação tradicional, o trabalhar os dados sem estimular a imaginação.

Referências bibliográficas

DANN, W., Cooper, S., Pausch, R. (2000). “Making the connection: programming with animated small worlds”, *In Proceedings of the 5th annual conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, July, 2000, pp. 41-44.

DILLENBOURG, Pierre (2000). “Learning In The New Millennium: Building New Education Strategies For Schools”, *Workshop on Virtual Learning Environments*, [Online] <http://tecfa.unige.ch/tecfa/publicat/dil-papers-2/Dil.7.5.18.pdf> (acedido em 12 de Julho de 2006).

ESTEVEES, M. and Mendes, A. (2004). “A Simulation Tool to Help Learning of Object Oriented Programming Basics”. *In Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference*, Savannah, Georgia, USA, October 2004, 20-23.

ESTEVEES, Micaela; Antunes, Ricardo; Morgado, Leonel; Martins, Paulo; Fonseca, Benjamim (2007). “Contextualização da aprendizagem da Programação: Estudo Exploratório no Second Life”. *In Gonçalves, Ramiro; Santoro, Flavia; Isaías, Pedro; Gutiérrez, José (eds.) & Rodrigues, Luís; Barbosa, Patrícia (ass. eds.), Actas da Conferência IADIS Ibero-Americana WWW/Internet 2007*, Vila Real, Portugal, Outubro, 7-8, 2007, ISBN 978-972-8924-45-4, pp. 253-256, Lisbon, Portugal: IADIS Press.

FIGUEIREDO, A. D. and Afonso, A. P., (2006). “Managing Learning in Virtual Settings: the Role of Context”, Information Science Publishing.

FLEURY, M. & Oliveira Junior, M.(org.), (2001). “Gestão do Conhecimento Estratégico – Integrando Aprendizagem, Conhecimento e Competências”, Editora Atlas, São Paulo.

GOMES, A. and Carmo, L. and Bigotte, E. and Mendes, A.(2006). "Mathematics and programming problem solving", *3rd E-Learning Conference – Computer Science Education*, Coimbra, September 2006.

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.

LAVE, J. & Wenger, E. (1991). “Situated Learning: Legitimate Peripheral Participation”. Cambridge: Cambridge University Press.

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.

MILISZEWSKA, I., Tan, G., “Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming”, *Journal of Issues in Informing Science & Information Technology*, Vol. 4, 2007, 277-289.

SILVIO, José (2008), “Las comunidades virtuales como conductoras del aprendizaje permanente”, [Online]:
http://funredes.org/mistica/castellano/ciberoteca/participantes/docuparti/esp_doc_31.htm (accedido em 20 de Abril de 08).

WENGER, E.C. & Snyder, W. M. & McDermott, R. (2002). “Cultivating communities of practice: a practitioner's guide to building knowledge organizations”, Harvard Business School Press Book.