

Combinando Diversão e Educação: Castelo dos Enigmas, um Jogo Sério para o Ensino de Algoritmos

Pasqueline Dantas Scaico, Diego Lopes Marques, Max André de Azevêdo, Jarbele Cássia da Silva, Sinval Vieira Mendes Neto¹

¹Centro de Ciências Aplicadas e Educação – Universidade Federal da Paraíba (UFPB) - Rua da Mangueira, s/n - CEP 58.297-000 – Rio Tinto – PB – Brasil

{pasqueline, diego.lopes, max.azevedo, jarbele.cassia, sinval.vieira}@dce.ufpb.br

Abstract. *This work presents a serious game that is being developed to familiarize students with the concepts of programming. The proposal of this work aims to provide a visual language to teach the structures of a programming language and the basic concepts of algorithms development.*

Resumo. *Este trabalho apresenta um jogo sério que está sendo desenvolvido para familiarizar alunos iniciantes com os conceitos de programação. A proposta objetiva fornecer uma linguagem visual para ensinar as estruturas de uma linguagem de programação e os conceitos básicos da elaboração de algoritmos, além de ser um mecanismo de avaliação que pode ser usado pelo professor.*

1. Motivação

A realidade presente nos cursos de Computação mundo afora aponta índices de abandono e retenção cada vez mais acentuados, Malmi (2006). Pesquisas têm demonstrado também que a procura por cursos na área tem caído e que a escassez de mão-de-obra qualificada para assumir futuros postos de tecnologia é uma previsão quase certa, Wilson (2010).

As disciplinas introdutórias de programação são aquelas que apresentam os piores desempenhos nos currículos universitários, Sudol (2011). Parte disso é proveniente de práticas pedagógicas antigas combinadas à existência de mecanismos suficientes para que professores acompanhem individualmente o progresso dos alunos, Simonyi (2007). Programar exige o uso de novas habilidades que nem sempre são assimiladas rapidamente. Assim, quanto mais o professor desconhece os pontos fracos dos seus alunos, maiores se tornarão as barreiras que se estabelecerão neste processo.

Outro aspecto importante para a aprendizagem é a motivação, que é necessária para que o estudante não desista frente às dificuldades. O trabalho de Koster (2004) apresenta a idéia de que as fronteiras entre aprendizagem-diversão e trabalho-jogo precisam ser desfeitas. Isto significa que o aprendizado precisa se aproximar do entretenimento para conseguir engajar os alunos. Neste contexto, os jogos sérios são uma abordagem diferente para tratar a diversão na educação. O entretenimento não deve ser tratado como um elemento estético relacionado à história, ao prazer ou a beleza da interface de um jogo educativo, mas como um mecanismo onde não existe pressão em aprender. Assim, é possível que o uso de jogos sérios no ensino de programação possa fazer com que o primeiro contato com a programação seja diferente e atraente.

Na literatura estão descritas dezenas de jogos para o ensino de algoritmos e programação. Grande parte se apresenta como ferramentas que ajudam a montar

pseudocódigos, acompanhar testes de mesa ou executar algoritmos. Contudo, poucas características de um jogo eletrônico se fazem presentes, fator este que pode desmotivar o aluno.

A existência de jogos que explorem a competição, estímulos sonoros, visuais e táteis, o uso de regras e a elaboração de estratégias juntamente com conteúdos específicos pode ser mais efetivo em disciplinas introdutórias de programação, tanto como um elemento motivacional quanto instrucional e de avaliação.

Este trabalho apresenta os resultados iniciais de um projeto inovador que tem como objetivo construir um jogo sério para o suporte ao ensino de programação, uma das atividades que vem sendo realizadas por pesquisadores do grupo de pesquisa Applied¹. O jogo está sendo desenvolvido para lidar com três tipos de situações. Primeiro, estabelecer um ambiente competitivo para que o jogador tenha interesse em voltar a jogar e avance na construção do seu conhecimento técnico. Segundo, funcionar como um mecanismo de acompanhamento para o professor, de forma que ele possa enxergar as decisões de cada aluno, fazendo com que ele perceba dificuldades individuais e seja capaz de intervir em tempo na aprendizagem, sendo de fato um agente facilitador do processo de aprender. E terceiro, ser executado também através de dispositivos que executem a plataforma Android.

O artigo está organizado a partir da seguinte estrutura: a Seção 2 apresenta os requisitos do jogo. Na Seção 3, o enredo e os desafios da versão implementada do jogo são apresentados. As seções 4 e 5 mostram as principais dificuldades enfrentadas até este ponto do projeto e o estágio atual do trabalho, respectivamente. A Seção 6 contém as considerações finais.

2. Requisitos do jogo

O jogo Castelo dos Enigmas é resultado de um trabalho de tutoria, que usa robótica educacional, com alunos ingressantes do curso de Ciência da Computação, UFPB. As pesquisas relacionadas que vem sido estabelecidas acerca de métodos de ensino de algoritmos, suporte ferramental para a construção de código, mecanismos de avaliação e de fatores motivacionais são os insumos para a construção do jogo, que se atém à proposta de incorporar mais entretenimento à aprendizagem dos novatos em programação através da implementação de requisitos pedagógicos, de usabilidade e de jogabilidade.

O Castelo dos Enigmas tem o intuito de fornecer desafios que explorem inicialmente a familiarização com conceitos básicos relacionados à sintaxe das principais estruturas de um algoritmo (repetição, condicional), palavras-reservadas e escopo através de um enredo, que está descrito na Seção 3, mas que também podem atingir níveis de dificuldades maiores, a exemplo da escolha correta de algumas estruturas de dados.

A especificação do jogo está pautada em vários requisitos. Eis alguns deles:

- i. Jogabilidade: deverão ser apresentados ao jogador desafios que exigirão velocidade, exploração, recompensa, tomada de decisão, pontuação e descoberta. Para estimular a competição o jogo poderá ser executado como

¹ <http://dgp.cnpq.br/buscaoperacional/detalhegrupo.jsp?grupo=0083103BLS2W5D>

- mono ou multiusuário. O jogador poderá salvar a fase da última jogada. O jogador terá um avatar;
- ii. Usabilidade: os cenários do jogo devem ser mudar a cada fase. O jogador poderá acessar menus para conferir os elementos que são conquistados nas jogadas;
 - iii. Pedagógicos: os elementos da programação devem ser introduzidos de maneira indireta ao jogador através de símbolos que lhe são apresentados e que são necessários para cumprimento dos desafios, os quais devem estar incorporados sempre ao enredo do jogo. À medida que o aluno usa os símbolos, o seu significado na programação será construído pelo próprio jogador. Em alguns desafios, ao perder vidas consecutivamente o jogador será exposto a outro desafio similar. O jogador poderá mediar a construção do seu conhecimento podendo definir o nível de dificuldade do jogo; as decisões do jogador durante o jogo serão armazenadas em um arquivo de log que será processado na forma de um relatório para que o professor tenha conhecimento do quanto o aluno evoluiu no jogo (desafios que conseguiu cumprir, número de acessos e tentativas, entre outros). Cada desafio será descrito a partir de *metadados*, que serão interpretados para que se construa uma avaliação para cada aluno no módulo do professor.
 - iv. Portabilidade: o jogo deverá ser compatível para a plataforma Android.

Existe a preocupação no projeto de que o estudante sempre esteja exposto a um desafio que requeira que ele detenha algum tipo de conhecimento técnico ou lógico. Contudo, os desafios que podem fazer com que ele perca vidas disfarçam a dificuldade que ele possa ter, não o expondo assim, de maneira negativa ao erro.

3. O jogo e seus desafios

O enredo do jogo conta a história de um aluno que tem uma prova de programação e começa a estudar apenas na noite anterior. Vencido pelo cansaço o aluno dorme e ao sonhar se vê em um castelo. Mesmo estando no sonho, há consciência de que é preciso sair daquele lugar. Ao longo da jornada, inimigos atacam o jogador e enigmas deverão ser desvendados. À medida que ele os vence, inconscientemente através do sono, ele percebe que está aprendendo. Quanto mais longe o jogador chega, mais ele estuda.

A característica marcante da primeira fase do jogo é a exploração. Em diferentes salas do castelo objetos diferentes podem ser capturados. Em uma delas há um cadeado. Em outra sala o jogador é atacado por morcegos (Figura 1). Ao destruir esses inimigos uma chave é encontrada. O jogador encontra em seguida um portão. Ao posicionar a chave na fechadura a porta não é imediatamente aberta. Uma instrução aparece na tela indicando que algo está sendo iniciado (Figura 2). O jogador perceberá com as jogadas que o uso da chave e do cadeado representa um conceito chamado escopo, que pode ser demarcado de diferentes maneiras pelas linguagens de programação e que nesta tela delimita a ação de abrir porta.

O jogador percebe com o tempo que utilizar os objetos significa ativar determinadas ações, que podem ser úteis para a solução dos enigmas e que são na verdade analogias para as estruturas que compõem os algoritmos. Outro caso é mostrado pela Figura 3, em que o jogador precisa criar uma expressão lógica para completar um teste, tendo para tanto que descobrir os objetos corretos que estão escondidos na sala.



Figura 1. (Sala do Castelo)



Figura 2. (Usando a chave na porta)



Figura 3. Algoritmo da balança



Figura 4. Teste com o peso das sacolas

Ainda na mesma figura, o uso da balança representa a semântica do teste de uma condição, realizado através de uma instrução *if*. Para que a condição correta seja satisfeita é preciso que o jogador descubra que ele precisa utilizar além dos objetos da sala o peso do próprio corpo para que a alavanca abra a porta e o desafio seja completado (Figura 4). Uma visualização melhor das telas do jogo podem ser encontradas no endereço <http://pibid.dce.ufpb.br/auxilio-ao-ensino-de-programacao>.

Na segunda fase do jogo uma estrutura importante é abordada, o laço de repetição. O objetivo a ser alcançado é passar por um corredor que estará totalmente apagado. Nele há cinco castiçais que deverão ser acesos para o ambiente se iluminar da seguinte maneira: ao final da fase 1 uma tocha é capturada e pode ser acesa através de uma lareira que se encontra numa pequena sala, perto do corredor. O desafio está no fato de entrar em um “laço de repetição” corretamente. Para acender as velas o jogador deverá seguir uma sequência de passos: i. acender a tocha, ii. acender a vela mais próxima que está apagada e iii. voltar para acender a tocha novamente, que se apagou neste percurso. Executando este algoritmo o jogador será capaz de chegar ao fim do corredor, quando deverá responder uma questão que lhe será feita sobre a condição de parada do laço que representa o desafio.

4. Principais dificuldades encontradas

O desenvolvimento de um jogo nos moldes de um jogo eletrônico tem imposto diferentes desafios. Inicialmente, o primeiro deles se concentrou na fase de Game Design, uma etapa que exige habilidades de desenho para especificação dos cenários e personagens. Outro aspecto é que os desafios precisam ser projetados com muita criatividade para que não haja uma separação entre o que é o entretenimento do jogo e o que é o software educativo. Assim, o processo de desenvolvimento teve de incorporar algumas fases: i. sessões de *brainstorming* (em que são levantadas as telas, documentadas a *posteriori* através de *storyboards*), ii. prototipação do jogo através do RPGMaker (que torna mais ágil a implementação dos desafios e a sua validação) e iii. os testes, que acontecem a partir de emuladores e celulares, e auxiliam o refinamento de requisitos relacionados à usabilidade e jogabilidade.

5. Estágio atual do trabalho

Está disponível (através de licença GLP), na forma de protótipo, a primeira fase do jogo, que já passou pela validação por um grupo de controle. Parte desta fase também está disponível para a versão em Android. As versões podem ser acessadas a partir do endereço eletrônico <http://pibid.dce.ufpb.br/auxilio-ao-ensino-de-programacao>.

Os requisitos recomendáveis para a execução do software através do RPGMaker são: Microsoft Windows XP ou superior, suporte à resolução 1024×768 High Color ou superior, suporte à DirectSound e DirectX 8 ou superior instalado. Para a versão existente para Android deve-se utilizar algum aparelho com suporte ao Android 2.1 ou o emulador para este sistema.

6. Considerações finais

O uso de uma linguagem visual para explicar os conceitos de uma linguagem de programação reforça a aprendizagem significativa. As primeiras fases do jogo investem na apresentação do significado de instruções de controle e repetição e ambienta o jogador no cenário de experimentação. Os usuários que avaliaram o protótipo até o momento classificaram o jogo como divertido, interessante, sendo possível perceber que alguns significados já foram assimilados pelo grupo de controle. A decisão de implementar para Android se deve ao fato de que os celulares são dispositivos acessíveis para os alunos. A distração que se espera que o jogo cause será interessante para forçar o raciocínio dos estudantes e para alimentar a base de informações do professor. Algumas decisões arquiteturais ainda estão sendo tomadas para que o processamento dos *logs* do jogo possa chegar de maneira simples para o professor e sem tanta dependência de acesso à Internet.

Referências bibliográficas

- Koster, R. (2004) Theory of fun for game design. Scottsdale: Paraglyph, p. 80-99.
- Malmi, L. "Why Students Drop Out CS1 Course? Categories and Subject Descriptors," Computer Science Education, ICER'06, September, 2006, Canterbury, pp. 97-108.
- Simonyi, C. and E. Roberts, "Encouraging Top Students in Large Undergraduate Classes," STANFORD UNIVERSITY NEWSLETTER ON TEACHING WINTER - SPEAKING OF TEACHING, vol. 8, 1997.
- Sudol, L.A. "Deepening Students' Understanding of Algorithms: Effects of Problem Context and Feedback Regarding Algorithmic Abstraction," Carnegie Mellon University, 2011.
- Wilson, C., L. A. Sudol, C. Stephenson and M. Stehlik, Running on Empty: The failure to teach K-12 computer science in the digital age, Association for Computing Machinery, 2010.