

Combinando desafios e aventura em um jogo para apoiar a aprendizagem de programação em vários níveis cognitivos

Eline Raquel de Macedo (Bolsista), José Raul B. Andrade (Bolsista), Ahemenson F. Cavalcante (Voluntário), Danilo Raniery A. Coutinho (Voluntário), Thiago G. Vasconcelos (Voluntário), Vanessa Farias Dantas (Orientadora)

1. Introdução

Há anos os altos índices de evasão e reprovação nas disciplinas iniciais de programação têm despertado o interesse e a preocupação de pesquisadores e educadores em todo o mundo, como relata Schultz (2003). Segundo Neto (2007), acadêmicos iniciantes, ao se depararem com a disciplina, sentem-se incapazes de programar, devido ao conjunto de habilidades que a programação exige, como: capacidade para solucionar problemas, habilidade matemática, capacidade de abstração, entre outras.

Uma revisão sistemática recente de Aureliano (2012) evidencia que as propostas para apoiar o ensino-aprendizagem da disciplina e combater esses problemas são bastante diversificadas, abrangendo desde ambientes para aprendizagem cooperativa (CASTRO, 2004) até experiências com robótica (RIBEIRO, 2011).

A partir da análise de várias dessas abordagens inovadoras, e diante do crescente interesse despertado pelos jogos educacionais, o presente trabalho descreve um jogo de aventura para motivar e desafiar os alunos, oferecendo oportunidades de evoluir ao longo das fases à medida que praticam seus conhecimentos de programação.

A Seção 2 descreve a concepção do jogo Robotimov em relação a enredo e aspectos pedagógicos, e a Seção 3 apresenta os resultados do desenvolvimento do protótipo de uma de suas fases. Por fim, na Seção 4, são apresentadas as considerações finais sobre o trabalho e as etapas futuras planejadas.

2. Robotimov

Segundo Tarouco (2005), os jogos se tornam atrativos quando possuem regras bem definidas, têm metas que geram motivação, são interativos, favorecem a aprendizagem através de *feedback*, e são divertidos. Com essas diretrizes em mente, a equipe iniciou a discussão sobre um enredo que pudesse causar empatia no público-alvo, mas que também despertasse sua curiosidade, transportando-os para um contexto de fantasia.

Era importante que os desafios de programação presentes em cada fase fossem de alguma forma relacionados ao contexto, proporcionando uma imersão ainda maior e evitando que houvesse uma clara distinção entre o aspecto lúdico e o aspecto instrucional. SIMÕES (2011) reforça essa ideia quando afirma que a proposta dos Jogos Sérios é que os conteúdos sejam inseridos no jogo, fazendo com que a aprendizagem se misture com o ato de jogar.

A analogia escolhida foi inspirada nas obras de ficção de Asimov (1950): diante de diversos robôs defeituosos, o jogador precisaria realizar ajustes em seus códigos-fonte para que eles voltassem a desempenhar suas funções. A definição do enredo levou em conta também a flexibilidade na criação de fases para que o jogador sempre seja surpreendido com cenários e personagens diferentes, evitando que ele fique entediado, e despertando seu desejo de evoluir no jogo. A visão completa das fases do jogo será possível através de um mapa do parque, com destaque para os ambientes cujos problemas ainda não foram solucionados.

Elaborado o enredo, o próximo passo para a criação do jogo foi elaborar seis tipos de problemas de programação que pudessem ser usados na composição das fases. Enquanto alguns desafios focam na sintaxe da linguagem de programação, outros exigem conhecimento sobre lógica, e outros ainda exercitam a depuração de erros.

As fases podem variar em tamanho do cenário e duração, e podem propor diferentes desafios de programação de um mesmo tipo mais de uma vez. Entretanto, é importante que uma fase não aborde sempre um único tipo de desafio para não confundir o jogador, nem dificultar a percepção por parte do professor sobre qual nível de aprendizagem ainda não foi alcançado.

Apesar de ser desejável que as fases sejam jogadas respeitando a evolução cognitiva dos níveis de Bloom, condicionar o acesso a uma fase à conclusão de outras demandaria bastante tempo aos alunos na hora de jogar, e poderia, inclusive, desmotivá-los por sempre terem que passar pelas fases consideradas mais simples para poderem praticar a habilidade que realmente desejam. Sendo assim, optou-se por não fixar pré-requisitos para cada fase. Contudo, no momento da escolha, o jogo apresentará uma sugestão sobre qual deveria ser a próxima fase, indicando no mapa do parque uma área emergencial.

3. Protótipo – Fase Balada

Para dar início ao desenvolvimento do jogo, foi montada uma equipe envolvendo alunos dos cursos de Design e Computação, de modo que cada um contribuísse com suas respectivas aptidões na concepção dos personagens, dos cenários e dos desafios de programação. A linguagem escolhida como objeto pedagógico foi Python, uma vez que essa é a linguagem adotada nas disciplinas iniciais de programação dos cursos da área de Informática no campus, proporcionando um público-alvo acessível para a validação do jogo.

O cenário proposto para a primeira fase foi o de um bar em que os robôs estivessem se divertindo ao som de uma banda, também composta por robôs. Com a sabotagem mencionada no enredo, alguns robôs passam a ter comportamentos estranhos. Além de se esquivar desses perigos, o herói ainda precisa escapar dos ataques dos robôs inimigos e coletar as engrenagens perdidas, necessárias para completar o desafio de programação da fase final.

A Figura 2 ilustra respectivamente o avatar correspondente ao herói do jogo, um dos robôs músicos da fase, e o robô inimigo. Já a Figura 3 apresenta algumas cenas da fase. As engrenagens com os símbolos de sintaxe e as palavras reservadas aparecem espalhadas por todo o cenário, exigindo atenção do jogador e contribuindo para a memorização da linguagem.

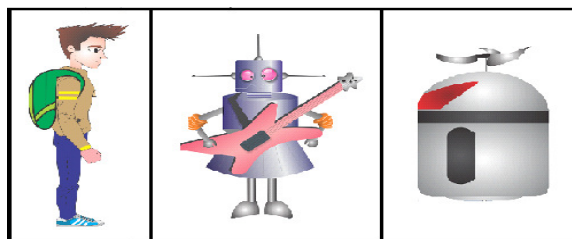


Figura 2 – Personagens da Fase Balada do Jogo Robotimov

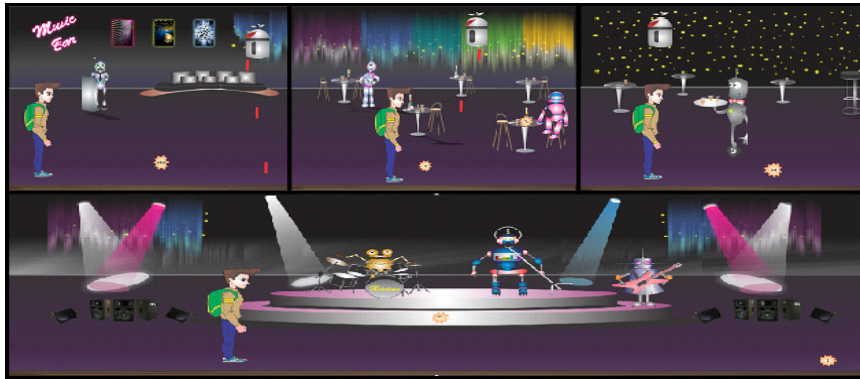


Figura 3 – Cenário da Fase Balada do Jogo Robotimov

No quadro final da fase, o jogador se depara com o painel de controle que gerencia todo o funcionamento do bar embutido no balcão. Ao se aproximar desse painel, uma nova tela aparece, exibindo as engrenagens coletadas na fase, o código incompleto que precisa ser solucionado através do encaixe das engrenagens adequadas, e os interruptores capazes de testar o posicionamento das engrenagens e o funcionamento completo do mecanismo (Figura 4).



Figura 4 – Desafio de Programação proposto para a fase Balada do Jogo Robotimov

Os tamanhos e cores diferenciados das engrenagens fornecem dicas para o jogador, e ajudam a consolidar o conhecimento sobre os elementos válidos da sintaxe Python. Caso alguma engrenagem importante fique faltando, o jogador poderá voltar à fase para

buscá-la. Também é possível que nem todas as engrenagens coletadas sejam utilizadas, forçando o jogador a escolher. Sempre que uma engrenagem for posicionada, o jogador poderá utilizar o primeiro interruptor para testar os encaixes, visualizando uma mensagem de erro caso alguma não esteja correta. Quando todas estiverem encaixadas corretamente, o jogador poderá usar o segundo interruptor para acionar o mecanismo, e ver uma pequena animação mostrando a fase em harmonia.

Para a implementação do protótipo, foram analisadas duas *engines*: Construct 2 e Unity. Apesar de ambas oferecerem uma grande variedade de recursos para a criação de jogos inclusive em 3D, observou-se que a Construct 2 demandaria uma quantidade muito elevada de eventos para propiciar todas as ações previstas para os personagens, tornando o código complexo e difícil de manter. Por isso, a equipe optou pela Unity.

4. Considerações Finais

Dentre tantas contribuições já feitas para a área de Ensino de Programação, espera-se que o Robotimov possa cumprir seus objetivos pedagógicos e lúdicos e tornar-se um valioso instrumento para apoiar o aprendizado da linguagem Python.

O projeto ainda encontra-se em fase inicial, e certamente ainda há muito a ser feito para que uma validação com alunos seja possível. Entretanto, a combinação de enredo original, visual elaborado das fases, e diversidade de desafios de programação para atender todas as etapas cognitivas do aprendizado evidencia uma importante iniciativa para motivar os alunos a praticarem seus conhecimentos de forma lúdica e divertida continuamente enquanto aprendem.

Referências

ASIMOV, I., I, Robot. Gnome Press. 1950.

AURELIANO, V. C. O.; TEDESCO, P. C. A. R., Ensino-aprendizagem de Programação para Iniciantes: uma Revisão. Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012), ISSN 2316-6533. Rio de Janeiro, 26-30 de Novembro. 2012.

CASTRO, T. H. C. de; JÚNIOR, A. N. de C.; MENEZES, C. S. de, Aprende – um Ambiente Cooperativo de Apoio à Aprendizagem de Programação. In: 15º Simpósio Brasileiro de Informática na Educação. Manaus, Brasil. 2004.

NETO, W. C. B.; SCHUVARTZ, A. A., Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores. In: 18º Simpósio Brasileiro de Informática na Educação. São Paulo, Brasil. 2007.

RIBEIRO, P. C.; MARTINS, C. B.; BERNARDINI, F. C., A Robótica como Ferramenta de Apoio ao Ensino de Disciplinas de Programação em Cursos de Computação e Engenharia. In: 17º Workshop de Informática na Escola, Aracaju, SE, Brasil. 2011.

SCHULTZ, M. R. O., “Metodologias para Ensino de Lógica de Programação de Computadores”. Monografia de Especialização (Ciência da Computação). Universidade Federal de Santa Catarina (UFSC). 69p. Florianópolis, SC, Brasil. 2003.

SIMÕES, J., Serious Games. Em: <<http://edulearning2.blogspot.com/2011/01/serious-games.html>>. Acesso em: 10 junho 2011.

Tarouco, L. (2005) “Jogos educativos via WWW”, disponível em <http://penta3.ufrgs.br/animacoes/JogosEducaionais>.