

Desenvolvimento de Jogos Educacionais para o Ensino de Engenharia de Software

Thaiana Lima Beatriz Pacheco Rodrigo Santos Cláudia Werner Fernando Limoeiro

COPPE/UFRJ – Univ. Federal do Rio de Janeiro – Programa de Eng. de Sistemas e Computação
Caixa Postal 68511 – CEP 21941-972 – Rio de Janeiro, RJ, Brasil

Resumo

Educational games represent important tools to aid the teaching and learning process because they allow simulating concepts in virtual environment and stimulating collaboration and communication among students. Since Software Engineering (SE) has been revisited in its formation process, researches in SE education emerged in the last decade, mainly associated to games. So, this paper analyzes some works related to the development of games for teaching SE in order to identify its peculiarities, that is, some characteristics, artifacts and phases to create educational games for this purpose. An experience of developing a game for requirements engineering in ubiquitous systems is presented and compared with the analysis carried out on other experiences.

Keywords: Game Development, Educational Games, Software Engineering.

Contatos dos autores:

{thaiana,biapc,rps,werner}@cos.ufrj.br
{n4ndoz@gmail.com}

Agradecimentos:

Os autores agradecem ao CNPq pelo apoio financeiro.

1. Introdução

O uso de jogos como ferramentas para auxiliar o processo de ensino e aprendizagem das diversas áreas do conhecimento tem aumentado nas escolas e universidades [Akilli 2007]. Com a acessibilidade a Web e aos dispositivos móveis, os alunos se sentem motivados com as novas tecnologias para aprenderem conteúdos acadêmicos de maneira mais fácil, rápida e divertida. Entre as tecnologias, destaca-se a realidade virtual/aumentada, TV digital/interativa, simuladores e *webquests*, entre outros [Werner *et al.* 2009]. Dessa forma, o campo dos jogos, vertente da área de Entretenimento, busca trazer motivação ao processo de aprendizagem, enquanto a simulação agrega realismo à experiência dos participantes [von Wangenheim *et al.* 2009].

Especificamente na área de Engenharia de Software (ES), os problemas e desafios do seu processo de ensino e aprendizagem motivaram a emergência de pesquisas no campo de Educação em ES [Lethbridge *et al.* 2007]. A principal motivação está no fato de que a ES é uma disciplina que envolve diversas atividades e abrange um conjunto de conceitos complexos, incluindo a própria natureza do

software. No âmbito prático do ensino de ES, o uso de jogos tem mostrado sucesso na imersão do aluno em cenários que refletem comportamentos típicos de situações reais. Trabalhos recentes têm se apoiado no uso de jogos como meio pedagógico [Dantas *et al.* 2004; Baker *et al.* 2005; Prikladnicki & von Wangenheim 2008].

No entanto, o processo de criação de jogos ainda não é muito focado no ensino da ES e seus conceitos, diferentemente de outras áreas do conhecimento [von Wangenheim *et al.* 2009]. Pensando no desenvolvimento dessas ferramentas, observa-se a necessidade da organização do processo de criação, contemplando as particularidades do contexto do jogo. No caso da ES, além dos jogos educacionais serem desenvolvidos/usados por alunos, profissionais e pesquisadores da própria área de ES, soma-se o foco em simulação, explorando abstração, trabalho cooperativo, desenvolvimento distribuído, decisões, evolução e outros [Fernandes & Werner 2009].

Assim, é oportuno dispor de meios para organizar o desenvolvimento de um jogo, que estabeleçam as atividades e artefatos a serem trabalhados e a utilização dos instrumentos adequados [Veronese 2004]. Por sua vez, um grande desafio na criação de jogos educacionais é manter o interesse do jogador e ao mesmo tempo ser didático com os conceitos. Para isso, algumas características como não linearidade do jogo, interação e desafios devem ser combinadas com sessões explanatórias no início e no fim de cada fase, desafio do jogo ou qualquer ação do jogador que esteja ligado ao assunto que o jogo quer esclarecer.

Diante da carência de jogos para o ensino de ES além dos temas gerência de projetos e processos de software, e da falta de um entendimento sobre o seu processo de criação, este artigo tem o objetivo de analisar alguns trabalhos que envolvem o desenvolvimento de jogos para o ensino de ES. Esta análise realça as atividades mais importantes das etapas presentes nos trabalhos comparados, seja seguindo algum processo definido ou de maneira *ad hoc*, assim como as características que fazem o desenvolvimento do jogo ser mais completo, e.g., sua documentação, inclusão de avaliação entre outros. Além disso, discute-se uma experiência de desenvolvimento de um jogo para ensino de engenharia de requisitos no contexto de sistemas ubíquos à luz da análise realizada. O artigo apresenta a seguinte estrutura: a Seção 2 discute e compara os

processos usados em alguns jogos para o ensino de ES; a Seção 3 apresenta uma experiência de desenvolvimento de um jogo para ensino de engenharia de requisitos no contexto de sistemas ubíquos; por fim, a Seção 4 conclui o artigo.

2. Processos de Desenvolvimento de Jogos para Ensino de ES

Na criação de qualquer jogo, seja ele educacional ou não, são seguidas algumas etapas essenciais, tais como *concepção da ideia do jogo, escolha da tecnologia de desenvolvimento, estabelecimento de metas, regras, desafios e objetivos*. No caso de jogos educacionais, existem preocupações adicionais, em geral a respeito do *conceito a ser ensinado e como será a abordagem escolhida*. Além disso, uma característica que pode deixar o jogo mais didático é que o desenvolvedor pode ser um profissional da área ou um aluno que já conheça o assunto, devido à sua experiência acadêmica e/ou profissional. Portanto, quem desenvolve um jogo para ensino de ES provavelmente entende as dificuldades dos alunos em geral, e também como apresentar o assunto de forma clara e simples para quem ainda está aprendendo.

Algumas atividades podem enriquecer o processo de desenvolvimento do jogo, assim como a qualidade do produto final. *Tutoriais, avaliações, pesquisas e documentação das etapas* seguidas na construção do jogo são exemplos de tais atividades. A partir disso, foram selecionados alguns jogos para comparação, através busca por artigos e teses cujo tema fosse voltado para ensino de ES. Foram selecionados os seguintes jogos: *Ilha de Requisitos* [Thiry et al. 2010], *The Incredible Manager* [Dantas et al. 2004], *Planager* [Prikladnicki et al. 2007] e *Software Quantum* [Knauss et al. 2008]. Os resultados da comparação realizada são apresentados na Tabela 1. Estes jogos foram escolhidos por representarem as áreas de Gerência de Projetos (GP) e Engenharia de Requisitos (ER), que são os principais campos da ES a possuírem jogos como ferramenta de ensino. Na maior parte dos jogos, a tecnologia escolhida para o desenvolvimento foi Java. Não foi encontrada uma grande diversidade de áreas da ES contempladas por jogos, o que mostra o potencial e a necessidade de explorar este tipo de ferramental no ensino de ES.

O jogo *The Incredible Manager* [Dantas et al. 2004] possui um processo de desenvolvimento do jogo bem detalhado, desde a escolha do tema do jogo abordado até o produto final gerado, passando pela tecnologia escolhida e definição de personagens, entre outros. O jogo *Software Quantum* tem o objetivo de transmitir fenômenos e *insights* a partir de uma simulação, porém não visa planejar projetos reais. O seu nome faz alusão às unidades atômicas. Estas unidades, no contexto do projeto (requisitos), são chamadas de “software quantum” [Knauss et al. 2008]. É abordado o tema ER.

Tabela 1: Comparação entre jogos para ensino de ES

Aspectos	<i>Ilha de Requisitos</i>	<i>The Incredible Manager</i>	<i>Planager</i>	<i>Software Quantum</i>
Formalização do processo	Sim	Sim	Sim	Não
Tecnologia	<i>Flash Player</i>	Java	Java	Java
Pontos fortes	Enredo criativo	Bem documentado	Simula sequência de projeto real	Bem teórico
Tópico de ensino	ER	GP	GP	ER
Plataforma	Web	Desktop	Desktop	Web
Apresenta avaliação?	Sim	Sim	Não	Sim
<i>Feedback</i> do jogador	Sim	Sim	-	Sim

O jogo *Planager* [Prikladnicki et al. 2007] se propõe a ensinar GP segundo o PMBOK¹. Além da formalização do projeto, foi elaborado um tutorial na forma de um vídeo em Flash. O desenvolvedor deste julgou que a maior dificuldade está na subjetividade presente na GP. Por este motivo, ele considera importante destacar a necessidade de descrever bem todo o cenário, visto que o jogo possui fins didáticos. O desenvolvimento do jogo *Ilha de requisitos* [Thiry et al. 2010] realizou as seguintes tarefas: concepção da ideia e definição da história, personagens, regras, desafios e maneira que o *feedback* é retornado ao jogador. Este jogo tem um enredo mais lúdico e, portanto, pode ser mais atrativo para o jogador.

Comparando a descrição do desenvolvimento contida na documentação dos jogos, todos eles cumprem as seguintes etapas: (i) estudo do tema a ser abordado; (ii) definição da ideia do jogo; (iii)

¹ <http://www.pmisp.org.br/educacao/guia-pmbok>

descrição do jogo (i.e., personagens, cenários, mecânica e outros); (iv) enredo; e (v) escolha da ferramenta de desenvolvimento. Estas etapas se mostram essenciais para a organização do processo de criação do jogo. No caso do jogo *Software Quantum*, este não descreve o processo de criação do jogo, mas descreve o jogo em si, como a mecânica, o conceito ilustrado e outras características.

Algumas etapas não foram explicitadas em todos os jogos analisados, entretanto estão presentes na maioria deles, e.g., a etapa de *avaliação*, que é importante para calibrar o efeito do jogo sobre os usuários. Algumas características também não foram encontradas em todos os jogos, e.g., o jogo retornar uma *medida do desempenho* dele, i.e., *feedback* para o usuário (o jogo mede e apresenta o avanço do jogador na busca de seus objetivos [Dantas *et al.* 2004]). O retorno ao jogador é feito por meio de *efeitos dramáticos*, que são os fatores que relacionam entretenimento ao conteúdo do aprendizado, por intermédio de sons, música, humor e histórias.

Em um trabalho sobre desenvolvimento de jogos, é proposto o seguinte processo [Veronese 2004]:

- *Descrição do jogo*: conta uma breve história sobre o contexto do jogo;
- *Modelagem da simulação*: apresenta a definição da estrutura do sistema modelado, identificando os elementos e os relacionamentos que compõem o sistema, melhor ilustrados por meio de diagramas;
- *Modelagem do enredo*: descreve aspectos da lógica e do enredo do jogo;
- *Modelagem das interações*: compreende a elaboração de um modelo que descreve os efeitos gráficos/eventos de interface do jogo, os principais conceitos abordados, imagens e animações de personagens e, por fim, objetos decorativos.

Comparando com os resultados da Tabela 1, observa-se que as *etapas de modelagem* não são, em geral, cumpridas tão rigorosamente. Todavia, a *etapa de avaliação* do jogo não está presente no processo proposto por Veronese (2004). A *elaboração de um tutorial* também é interessante em qualquer tipo de jogo, especialmente aqueles que se propõem a auxiliar no aprendizado, pois as instruções contidas no tutorial são vitais. Logo, o desenvolvimento mais completo é uma combinação dos casos discutidos.

3. Um Caso: *UbiRE*

Visando analisar pontos fortes, fracos, oportunidades e melhorias nas características, artefatos e fases do desenvolvimento de jogos, foi desenvolvido no Laboratório de Reutilização de Software da COPPE um jogo para ensino de ES, denominado *UbiRE*. Em abril de 2011, iniciou-se a *concepção da ideia do jogo* por uma *equipe* composta por uma aluna de iniciação científica do Ensino Médio, outra do curso de Engenharia de Computação e Informação e um

aluno de graduação em ES, gerenciados por um aluno de doutorado em ES. O jogo visa ensinar ER no contexto de um sistema ubíquo, usando uma casa como ambiente. O jogo *UbiRE* é baseado no conceito da Ubiquidade [Weiser, 1991], i.e., *computação em todo momento e em todo lugar*.

A casa deve ativar automaticamente os aparelhos e conexões, sem que o morador os ligue ou conecte manualmente. Para isso, o jogador é um profissional contratado para montar e configurar o sistema da casa inteligente. O jogo é dividido em quatro *fases*, representadas por cômodos, e existem três *níveis de dificuldade*, definidos pelo número de conexões feitas entre equipamentos para atender a um requisito, descrito, ainda, de forma clara ou complicada. O jogador escolhe o nível de dificuldade no início do jogo. Outro fator de dificuldade é o *número de tentativas* ser limitado. Na Figura 1, é mostrada a tela do quarto (primeira fase). Ao clicar em um objeto que pode ser conectado a outro, é aberto um *menu*. Se a opção selecionada for *conexões*, a tela mostrada na Figura 2 é aberta. Então, são mostrados os objetos do cômodo atual para que o jogador escolha.

O jogador vence quando conectar corretamente todos os objetos. A conexão está correta quando atende a, pelo menos, um dos requisitos que são mostrados no começo da fase, e que posteriormente ficam disponíveis no *menu* para serem acessados quando necessário. Por exemplo, ao clicar no aparelho de DVD, é aberta a tela de conexões (Figura 2). O jogador deve escolher o aparelho que deseja conectar entre os ícones à direita na tela. Em seguida, será pedida a configuração do objeto (i.e., ligado ou desligado, temperatura, entre outros). Um exemplo de requisito é “*O sistema deve ligar as luzes ao detectar movimento através de um sensor*”, em que a conexão correta seria a *lâmpada* conectada a um *sensor de movimento*, com a configuração da lâmpada e do sensor ambas como ‘ligado’.

No processo de criação desse jogo, a primeira decisão foi o *tema* a se ilustrar (i.e., a Ubiquidade). Contudo, os conceitos aplicados pelo jogador seriam os da ER, pois as ações principais do jogador (i.e., conexões) objetivam atender aos requisitos do sistema. Em uma etapa seguinte, foi escolhido um sistema ubíquo para representar o *ambiente do jogo*, no caso, a casa inteligente. Para a escolha da *ferramenta de desenvolvimento*, foi feita uma pesquisa e testes com alguns programas. O programa escolhido foi o GameMaker², cuja programação é facilitada pelo ambiente gráfico e por sua linguagem própria, o GML (*GameMaker Language*). Este programa é baseado em *eventos, ligados a objetos* do jogo. A imagem atribuída ao objeto é chamada *sprite*, que é separada do objeto e de seus eventos. Essa escolha foi baseada na facilidade de se aprender as funções do GameMaker, no suporte para importação

² <http://www.yoyogames.com/gamemaker>

de imagens e na possibilidade de uso de código da linguagem dele (simples). Para obtenção das *imagens do jogo*, foi utilizado o software para simulação de casas *Sweet Home 3D*³. As figuras foram importadas para o GameMaker de forma a aplicar ações do objeto ligado à imagem obtida.



Figura 1: Tela da primeira fase do jogo, o quarto



Figura 2: Tela de conexões entre os equipamentos

A maior dificuldade encontrada inicialmente foi a escolha da ferramenta adequada às necessidades do jogo. No começo do desenvolvimento, foi observado que a ferramenta anteriormente escolhida não daria suporte, de modo que um novo estudo foi realizado e o GameMaker passou a ser utilizado. A partir de então, a maior dificuldade encontrada foi o aprendizado da ferramenta e o desenvolvimento do jogo, simultaneamente. Como o objetivo do jogo é ensinar ER no contexto de um sistema ubíquo, outra dificuldade foi adequar o enredo para não confrontar esse objetivo, i.e., não deixar o morador da casa fazer nenhuma conexão, caso contrário, seria ferido o princípio da ubiquidade (i.e., transparência). Se comparado aos resultados da análise realizada na Seção 2, o jogo *UbiRE* atende a alguns aspectos como a *descrição do jogo*, *feedback para o jogador*, *avaliação parcial da usabilidade do jogo* (executada com o jogo ainda em desenvolvimento) e *descrição do enredo*. O jogo *UbiRE* funciona na plataforma *Desktop* e não houve formalização do processo, mas sim das características e elementos do jogo.

4. Conclusão

A utilização de jogos educacionais como estratégia no processo de ensino e aprendizado está em foco e, visando se beneficiar deste potencial, a área de Educação em ES vem ganhando foco, sobretudo no campo de treinamento, visto que um jogo pode

representar um ambiente de simulação de conceitos que, em aula, são difíceis de reproduzir em um nível significativo. Este artigo discutiu o desenvolvimento de jogos para o ensino de ES. Como contribuição, foram sumarizadas algumas características, fases e artefatos que devem fazer parte do processo de criação de um jogo educacional, considerando as particularidades do domínio, no caso, ensino de ES. Foi apresentado o jogo *UbiRE* como instrumento para melhorar o entendimento sobre o desenvolvimento de jogos para o ensino de ES. Apesar de não seguir um processo formal, este jogo apresenta algumas características, fases e artefatos descritos na Seção 2.

Como trabalhos futuros, pretende-se registrar os pontos fortes e fracos, bem como o processo de construção e avaliação do jogo *UbiRE*, avaliando-o como uma ferramenta para ensino de ER no contexto de sistemas ubíquos. Ainda, executar-se-á uma revisão sistemática para entender melhor os requisitos para desenvolvimento de jogos para ensino de ES em geral, a partir dos resultados obtidos com a combinação entre a análise realizada (Seção 2) e a experiência conduzida (Seção 3).

Referências

- AKILLI, G. K., 2007. Games and Simulations: A new approach in education. In: Gibson, D., Aldrich, C., Prensky, M. (eds.) *Games and simulations in online learning: research and development frameworks*. Information Science Publishing, Hershey/PA, 1-20.
- BAKER, A., NAVARRO, E. & VAN DER HOEK, A., 2005. An Experimental Card Game for Teaching Software Engineering Processes. *The Journal of Systems and Software* 75, 1-2 (February), 3-16.
- DANTAS, A., BARROS, M. & WERNER, C., 2004. Treinamento Experimental com Jogos de Simulação para Gerentes de Projeto de Software. In: *XVIII SBES*, Brasília, Brasil, 23-38.
- FERNANDES, L. & WERNER, C., 2009. Sobre o uso de Jogos Digitais para o Ensino de Engenharia de Software. In: *II FEES, SBES 2009*, Fortaleza, Brasil, 17-24.
- KNAUSS, E., SCHNEIDER, K. & STAPEL, K., 2008. *A Game for Taking Requirements Engineering More Seriously*. Leibniz Universität Hannover, Germany.
- LETHBRIDGE, T. C. ET AL., 2007. Improving Software Practice through Education: Challenges and Future Trends. In: *Conference on The Future of Software Engineering*, 29th ICSE, Minneapolis, USA, 12-28.
- PRIKLADNICKI, R. & VON WANGENHEIM, C. G., 2008. O Uso de Jogos Educacionais para o Ensino de Gerência de Projetos de Software. In: *I FEES, XXII SBES*, Campinas, Brasil, 37-45.
- PRIKLADNICKI, R., ROSA, R. & KIELING, E., 2007. Ensino de Gerência de Projetos de Software com o Planager In: *XVIII SBIE*, São Paulo, Brasil, 11-20.
- THIRY, M.; ZOUCA, A. & GONÇALVES, R., 2010. Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo. In: *XXI SBIE*, J. Pessoa, 1-10.
- VERONESE, G., 2004. *Sistematização do Desenvolvimento de Jogos de Simulação para Treinamento*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 133f.
- VON WANGENHEIM, C., KOCHANSKI, D. & SAVI, R., 2009. Revisão Sistemática sobre Avaliação de Jogos Voltados para Aprendizagem de Engenharia de Software no Brasil. In: *II FEES, SBES 2009*, Fortaleza, Brasil, 41-48.
- WEISER, M., 1991. The Computer for the 21st Century. *Scientific American* 265, 3, 94-104.
- WERNER, C. ET AL., 2009. Projeto Tec3ES: Tecnologias e Estratégias para Educação em Engenharia de Software. In: *17th CIESC, CLEI 2009*, Pelotas, Brasil, 1-2.

³ <http://www.sweethome3d.com/index.jsp>