

Um Jogo de Estratégia de Gerência de Configuração

Karen Figueiredo, Jonatas Ferreira, Leonardo Murta, Esteban Clua

Instituto de Computação – Universidade Federal Fluminense (UFF)

{kfigueiredo, jferreira, leomurta, esteban}@ic.uff.br

***Resumo.** A utilização de jogos na educação é uma prática que vem ganhando destaque, tendo em vista seus benefícios no processo de ensino e aprendizagem. Desta forma, este artigo propõe o JEEES, que é um jogo de estratégia inovador, com intuito de transmitir conhecimentos de Gerência de Configuração. Além da descrição do JEEES, este artigo também apresenta uma avaliação preliminar do jogo em turmas de engenharia de software.*

1. Introdução

Os jogos são elementos presentes na nossa vida desde a infância, inicialmente como exercício onde a criança repete determinada situação por puro prazer. Posteriormente, com uma idade mais avançada, os jogos tendem a se tornar estratégicos e regrados [Silva 2009]. Estimulados pela corrente pedagógica do construtivismo, que ressalta a participação do aluno na construção de seu próprio conhecimento por meio de suas interações, os jogos começaram a ser utilizados no âmbito educacional [Tarouco *et al.* 2004]. De tal modo, quando elaborados com intuito educativo, os jogos podem receber diferentes nomenclaturas tais como: jogos educacionais ou educativos, jogos de aprendizagem ou jogos sérios (do inglês, *serious games*), sendo que alguns tipos de simuladores também podem ser considerados jogos educacionais [Savi 2008].

Segundo Tarouco *et al.* (2004), os jogos educacionais são importantes ferramentas instrucionais já que os mesmos divertem e motivam os alunos, facilitando assim o aprendizado e aumentando a capacidade de retenção do que foi ensinado. Além do caráter motivacional, os jogos educativos ajudam os alunos a desenvolverem uma série de habilidades e estratégias e, por isso, começam a ser tratados como importantes materiais didáticos [Gros 2003]. O fato dos jogos implicarem em uma imersão e participação ativa do usuário ressalta ainda mais o potencial de aprendizagem e fixação de conteúdo.

Dentre as várias disciplinas da Engenharia de Software, a Gerência de Configuração (GC) assume um papel fundamental de apoio ao controle da evolução [Dart 1991] durante o desenvolvimento e, em especial, durante a manutenção de software. Contudo, em 2005 somente 25% de um total de 2500 empresas brasileiras faziam uso de GC [MCT 2006]. Apesar da sua importância e da baixa adoção na indústria brasileira, raros trabalhos relatam esforços relacionados ao ensino de GC [Bendix 2001], e nenhum trabalho relata a adoção de jogos com esse propósito.

Desta forma, o objetivo deste artigo é propor um jogo educacional baseado em Engenharia de Software que visa ensinar especificamente conceitos e práticas de GC. Esse jogo, denominado JEEES (Jogo de Estratégia para o Ensino de Engenharia de Software) tem como inspiração elementos e conceitos do jogo SimulES [Figueiredo *et al.* 2006], que simula o processo de desenvolvimento de software centrado na perspecti-

va de evolução. Além do SimulES, existem outras iniciativas de construção de jogos educacionais em Engenharia de Software, como, por exemplo, em Gerência de Projetos [Kieling e Rosa 2006], Processo de Software [Navarro 2006] e Qualidade de Software [Lino 2007]. Contudo, como discutido anteriormente, nenhuma delas trata de GC. Essa foi a principal motivação deste trabalho, tendo como hipótese que iniciativas desse tipo tornariam o ensino de conceitos de GC mais efetivo e prazeroso para os alunos, atraindo mais adeptos e entusiastas para o tema. Após a elaboração do JEEES, foram executados alguns estudos experimentais em turmas de cursos de engenharia de software visando obter indícios que apoiem essa hipótese.

Além desta seção, que apresenta a motivação, trabalhos relacionados, e objetivo do trabalho, o restante deste artigo está organizado da seguinte forma: na seção 2 o jogo JEEES é descrito, apresentando seu material e dinâmica; a seção 3 mostra a avaliação preliminar da primeira versão de JEEES; e na seção 4 são feitas as considerações finais.

2. JEEES: Jogo de Estratégia para o Ensino de Engenharia de Software

JEEES é um jogo de estratégia elaborado com o intuito de ensinar diversos aspectos da Engenharia de Software. Ele é constituído, em sua maior parte, por diferentes tipos de cartas e é através destas que os conteúdos são transmitidos para os jogadores. Assim, novas cartas podem ser adicionadas e cartas que se mostrarem pouco funcionais podem ser excluídas, permitindo a criação de inúmeras versões do jogo que podem ter propósitos específicos ou não. Por exemplo, a versão atual do jogo, apresentada neste artigo, é voltada especificamente para o ensino de GC, pois o deque de cartas em uso é relacionado a conceitos dessa área. Desta forma, para o ensino de outras disciplinas de Engenharia de Software, como, por exemplo, Medição e Análise ou Garantia da Qualidade, bastaria que deques de cartas voltados para essas áreas fossem produzidos. As regras do jogo e todo o material de apoio continuariam os mesmos.

2.1. O material do jogo

O jogo é constituído basicamente de um tabuleiro de projeto para cada jogador (Figura 1) e diversos tipos de cartas (Figura 2) descritos a seguir, além do material de apoio: tabuleiro para organização da área da mesa, dados de 6 faces, “dinheiro” para as transações comerciais do jogo, e marcadores. A versão atual do JEEES conta com 23 cartas com conteúdos de GC, além de 34 cartas de propósito geral (Tabela 1).

O **Tabuleiro de Projeto** é a área onde o jogador vai desenvolver o seu projeto (representado pela **Carta de Projeto**) que consiste no seu objetivo do jogo. No tabuleiro de projeto o jogador irá dispor as equipes (**Carta de Equipe**) que irão trabalhar no projeto nas respectivas áreas de desenvolvimento e de testes. Ao longo do jogo, estas equipes produzirão as funcionalidades dos componentes das *releases* do projeto. As funcionalidades produzidas poderão apresentar erros, os quais devem ser identificados pelas equipes de teste, ou defeitos, que devem ser corrigidos pelas equipes de desenvolvimento. Também são agregados ao tabuleiro de projeto os conceitos que o jogador tenha aplicado ao seu projeto (**Carta de Conceito**) e os eventos que estão em andamento que afetam o projeto (**Carta de Evento**).

A **Carta de Projeto** representa o objetivo do jogo para cada jogador, ou seja, o projeto que este jogador deve concluir para ganhar uma partida de JEEES. Ela contém:

(i) a descrição básica do projeto em questão, (ii) o número de *releases* a serem completadas e entregues para conclusão do projeto, (iii) o número de funcionalidades que devem ser produzidas para cada componente das *releases*, (iv) a qualidade do projeto, i.e., o número aceitável de defeitos que o projeto final poderá conter, e (v) a quantidade de dinheiro que será arrecadada no início do projeto e ao final de cada *release*.

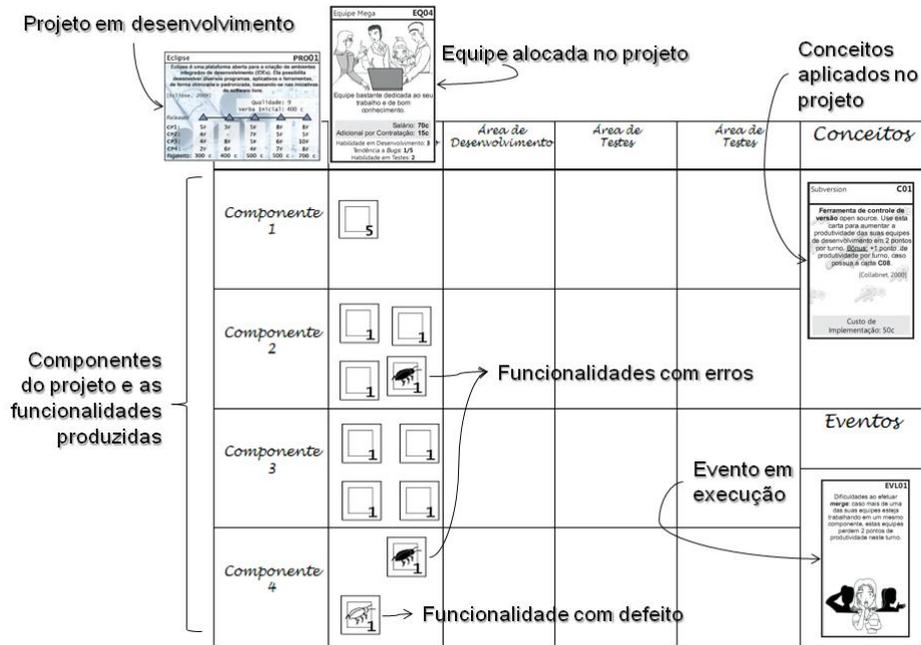


Figura 1. Tabuleiro de projeto do JEEES

Eclipse		PRO01				
Eclipse é uma plataforma aberta para a criação de ambientes integrados de desenvolvimento (IDEs). Ela possibilita desenvolver diversos programas, aplicativos e ferramentas, de forma otimizada e padronizada, baseando-se nas iniciativas de software livre.						
[Eclipse, 2009]						
		Qualidade: 9 Verba Inicial: 400 c				
Releases	▲	▲	▲	▲	▲	
CP1:	5F	3F	5F	8F	8F	
CP2:	4F	-	7F	5F	5F	
CP3:	4F	8F	5F	6F	10F	
CP4:	2F	6F	4F	7F	8F	
Pagamento:	300 c	400 c	500 c	500 c	700 c	

(a)

Equipe Mega EQ04

Equipe bastante dedicada ao seu trabalho e de bom conhecimento.

Salário: 70c
Adicional por Contratação: 15c
Habilidade em Desenvolvimento: 3
Tendência a Bugs: 1/5
Habilidade em Testes: 2

Subversion C01

Ferramenta de controle de versão open source. Use esta carta para aumentar a produtividade das suas equipes de desenvolvimento em 2 pontos por turno. **Bônus:** +1 ponto de produtividade por turno, caso possua a carta C08.

[Collabnet, 2000]

Custo de Implementação: 50c

EVL01

Dificuldades ao efetuar merge: caso mais de uma das suas equipes esteja trabalhando em um mesmo componente, estas equipes perdem 2 pontos de produtividade neste turno.

EVG01

Uma multinacional entrou no mercado e oferece ótimas oportunidades de emprego. Todos os jogadores devem ceder uma de suas equipes ou aumentar o salário de todas em 1 ponto de adicional por contratação.

(b)

(c)

(d)

(e)

Figura 2. Exemplos de cartas do jogo: carta de projeto (a), carta de equipe (b), carta de conceito (c), carta de evento local (d) e carta de evento global (e)

Tabela 1. Cartas da primeira versão de JEEES

Tipo da Carta	Quantidade	Tipo de Conteúdo
Projeto	5	Projetos <i>open-source</i> . Propósito geral.
Equipe	10	Propósito geral.
Evento	32	14 cartas com conteúdos de GC. 18 cartas de propósito geral.
Conceito	10	9 cartas com conteúdos de GC. 1 carta de propósito geral.

A **Carta de Equipe** representa a mão de obra do projeto. Quando uma equipe é alocada em desenvolvimento, ela irá produzir os componentes das *releases*, efetuando mudanças e corrigindo defeitos. Quando é alocada em testes a equipe irá detectar os erros nas *releases* candidatas e reportar às equipes de desenvolvimento. Esse tipo de carta contém os seguintes atributos: (i) salário, (ii) adicional por contratação, que é uma taxa que deve ser acrescentada ao valor do salário da equipe cada vez que for contratada por um outro jogador, (iii) a habilidade possuída pela equipe em desenvolvimento, (iv) a tendência de produção de defeitos da equipe, e (v) a habilidade que a equipe tem para efetuar testes.

A **Carta de Conceito** descreve um elemento que pode ser agregado ao projeto. Uma carta de conceito pode ser uma ferramenta, uma técnica ou um processo. Cada carta de conceito contém sua descrição e custo de implementação. Finalmente, a **Carta de Evento** apresenta uma situação (boa ou ruim) que pode ocorrer em um projeto de software. Cada carta de evento possui a descrição do evento e causa algum efeito ao jogador que sorteou o evento (evento local) ou a todos os jogadores (evento global).

O material de JEEES possui uma série de marcadores utilizados para representar as funcionalidades, funcionalidades com erro e funcionalidades com defeito do projeto (Figura 1), e, ainda, marcadores de humor e de adicional de contratação das equipes. Além disso, o material também inclui um pequeno guia com as instruções do jogo, uma lista de todas as referências que foram utilizadas nas cartas e leituras recomendadas e um glossário com os termos marcados em negrito nas cartas para complementar o aprendizado. O material completo do jogo está disponível em (<http://sel.ic.uff.br/jeees>).

2.2. A dinâmica do jogo

O objetivo do JEEES é concluir um projeto de software. Para isto o jogador deve concluir e entregar todas as *releases* descritas em sua carta de projeto. Ganha o jogo aquele que terminar primeiro o projeto ou, em caso de empate, aquele que possuir mais dinheiro no final. O número de jogadores recomendado para esta primeira versão é de 2 a 3 jogadores devido a quantidade de cartas, mas JEEES também poderia ser jogado individualmente. A dinâmica do jogo é organizada em turnos para cada jogador, tal como descritos a seguir:

No **Turno de Contratação** o jogador pode contratar uma das equipes disponíveis no mercado, à sua escolha, pagando o salário indicado, ou tentar contratar uma equipe de outro jogador, aumentando o salário atual da equipe de acordo com o adicional por contratação da mesma. A equipe contratada deve ser alocada em uma das funções: desenvolvimento ou testes. Quaisquer realocações de função de outras equipes também devem ser realizadas nesse turno. No **Turno de Compra** o jogador deve retirar uma carta de evento e realizar as ações indicadas pela carta. Caso a carta de evento se refira a

todos os jogadores (evento global), todos os jogadores devem realizar as ações indicadas.

No **Turno de Produção** o jogador pode implementar as modificações nos componentes conforme a produção das suas equipes a fim de completá-los para a conclusão da *release* em desenvolvimento. As modificações realizadas devem respeitar o número de tendência a defeitos de cada equipe. Os defeitos reportados pela equipe de teste devem ser consertados nesse turno pelas equipes de desenvolvimento. Após o turno de produção ocorre o **Turno de Conclusão da Release Candidata** caso o jogador tenha completado o número de modificações necessárias nos componentes da *release*. Assim, ele pode anunciar a *release* candidata e entregar seus componentes para as equipes de teste. No **Turno de Testes** o jogador pode detectar os erros nos componentes da *release* candidata com a habilidade das suas equipes de teste. Depois de detectados, os erros devem ser reportados à equipe de desenvolvimento.

No **Turno de Conclusão da Release**, caso a *release* candidata tenha a sua qualidade dentro do limite aceitável pelo projeto, e se o jogador assim desejar, a conclusão da *release* pode ser anunciada e a sua entrega realizada. O jogador recebe o pagamento referente à *release*. O jogador também pode tentar entregar a *release* para o cliente mesmo se ela estiver fora dos padrões, porém, neste caso, haverá uma probabilidade de rejeição do cliente. O último turno é o **Turno de Pagamento**, onde o jogador deverá pagar o salário das suas equipes. Caso o pagamento não seja efetuado o jogador sofre algumas penalidades como a perda de produtividade das suas equipes no turno seguinte.

3. Avaliação do Jogo

A fim de avaliar o jogo desenvolvido, a primeira versão de JEEES foi aplicada em três turmas de graduação de engenharia de software de duas instituições federais de ensino superior, a saber: 4º período do curso de Sistemas de Informação do Instituto Federal Fluminense (Turma A), 5º período do curso de Tecnologia em Análise e Desenvolvimento de Sistemas de Informação do IFF (Turma B) e turma de Engenharia de Software I (disciplina de 5º período) do curso de Ciências da Computação da Universidade Federal Fluminense (Turma C).

O objetivo desta avaliação preliminar foi investigar qual seria a percepção dos alunos com relação ao jogo e avaliar se eles conseguiriam apreender o conteúdo de GC que foi proposto no jogo. Para isso, foram elaborados 2 questionários de avaliação, ambos com 8 questões de múltipla escolha cada. O primeiro questionário avalia a qualidade do jogo e, além das 8 perguntas apresentadas na Tabela 2, conta com uma área para comentários, críticas e sugestões pessoais.

O segundo questionário avalia se os conteúdos propostos nas cartas desta primeira versão do jogo foram compreendidos pelos alunos e, além das opções de resposta para as perguntas de GC, cada questão possui também as seguintes opções: “Já sabia” – caso o aluno já soubesse a resposta da pergunta antes da dinâmica do jogo, “Não vi nada sobre isso no jogo” – caso o aluno não tivesse visto o conteúdo ao qual a pergunta se referia nas cartas durante o jogo, e “Não sei responder” – caso o aluno não soubesse a resposta para a questão. As perguntas desse questionário, juntamente com as suas opções de resposta, podem ser visualizadas na Tabela 3.

Tabela 2. Questionário de avaliação da qualidade do jogo

Pergunta	Opção 1	Opção 2	Opção 3
1) Após ter jogado JEEES, você diria que:	Gostou	Gostou muito	Não gostou
2) Quanto à jogabilidade, você diria que o JEEES é:	Normal	Fácil	Complicado
3) Você diria que aprendeu com o jogo?	Sim	Não	-
4) Você acha que o tempo de duração do jogo foi adequado para aprender o jogo e conhecer o assunto?	Pouco	Adequado	Muito
5) Entre o jogo e uma aula, você prefere:	Aula e jogo	Jogo	Aula
6) Uma versão digital do jogo atrairia mais sua atenção?	Sim	Não	-
7) Você gostaria de jogar JEEES novamente?	Sim	Não	-
8) O jogo despertou seu interesse por GC?	Sim	Não	-

Tabela 3. Questionário de avaliação do conteúdo de GC

Pergunta	Opções de resposta
1) Um dos objetivos principais da GC é controlar:	a) A evolução do software b) A qualidade do software c) O custo do software
2) Relacione o nome das ferramentas à sua utilização: [1] Subversion; [2] Ant; [3] Snort; [4] Trac	[] Ferramenta de controle de versões [] Ferramenta de controle de modificações [] Ferramenta de gerenciamento de construção e liberação
3) Quais práticas devem ser adotadas para que as equipes de um projeto saibam como trabalhar melhor em grupo paralelamente, evitando problemas como os de merge, por exemplo?	a) Políticas de Backup b) Políticas de Engenharia Concorrente c) Políticas de Configuração
4) Relacione o tipo de “bug” à sua definição: [1] Defeito; [2] Falha; [3] Erro	[] Problema interno em um artefato. É descoberto através da depuração/inspeção. [] Problema de percepção externa que denota que o artefato não se comportou conforme o esperado. É descoberto através de testes.
5) O documento que descreve todo o processo de GC de um projeto é chamado de:	a) Estratégia de GC b) Política de GC c) Plano de GC
6) O conjunto de técnicas que descrevem como construir um ambiente de desenvolvimento apoiado por GC para criar software mais rápido e confiável é chamado de:	a) Padrões de GC b) Treinamento de GC c) Técnicas de Desenvolvimento Ágil
7) O estado em que um sistema se encontra em um determinado momento constitui a sua:	a) Alteração b) Configuração c) Classificação
8) Quando uma nova versão de um produto de software é lançada é chamada de:	a) Merge b) Commit c) Release

A dinâmica proposta teve duração de 1 hora e 30 minutos nas turmas A e B e 1 hora na Turma C, divididas respectivamente em: 20 minutos de apresentação do jogo, 50 minutos jogando e 20 minutos para aplicação dos questionários; e, 15 minutos de apresentação do jogo, 30 minutos jogando e 15 minutos para aplicação dos questionários. A duração da dinâmica foi determinada pelo tempo disponibilizado pelos professores das turmas. Na Turma A, 8 alunos participaram da dinâmica; na Turma B, 18 alunos participaram; e na Turma C, 21 alunos, totalizando 47 alunos. Estes alunos foram divididos em grupos de 4, 5 ou 6 alunos, nos quais 3 alunos jogavam e os demais alunos observavam a partida auxiliando com a distribuição das cartas e inspecionando se as

regras estavam sendo cumpridas. Os resultados gerais do primeiro e segundo questionários aplicados podem ser observados na Tabela 4.

Com relação ao primeiro questionário (Tabela 4.a), referente à qualidade do jogo, é possível observar que todos os resultados foram satisfatórios, considerando que todos os alunos gostaram do jogo e que 89% afirmaram ter aprendido com o jogo. Além disso, 95% responderam que o jogo fez despertar o interesse por GC. Ambos os indícios estão alinhados com a hipótese, levantada na seção 1. Entretanto, os alunos também apontaram o tempo curto da dinâmica, já que 78% acharam pouco o tempo de jogo. A estimativa de duração de uma partida de JEEES com 3 jogadores é de 2 horas, contudo o experimento não teve tanto tempo disponível. Este aspecto talvez justifique as dificuldades que alguns alunos demonstraram ao jogar (17% acharam o jogo complicado).

Tabela 4. Avaliações da qualidade do jogo (a) e do conteúdo de GC (b)

1º questionário (a)				2º questionário (b)					
Questão	Opção 1	Opção 2	Opção 3	Questão	Respostas certas	Respostas erradas	Não viu no jogo	Não soube responder	Já sabia
1	85,1%	14,9%	0	1	68,1%	14,9%	8,5%	6,4%	2,1%
2	63,8%	19,1%	17,1%	2	19,2%	0	48,9%	31,9%	0
3	89,4%	10,6%	-	3	14,9%	4,3%	48,9%	31,9%	0
4	78,7%	21,3%	0	4	8,5%	14,9%	29,8%	46,8%	0
5	57,4%	36,2%	6,4%	5	8,5%	10,6%	29,8%	51,1%	0
6	74,5%	25,5%	-	6	12,8%	19,2%	34,0%	34,0%	0
7	93,6%	6,4%	-	7	46,8%	4,3%	21,3%	27,6%	0
8	95,7%	4,3%	-	8	44,7%	0	4,3%	17,0%	34,0%

No segundo questionário, referente à compreensão dos conteúdos transmitidos, as questões 1, 4, 7 e 8 são relacionadas à GC de um modo geral e a elementos básicos do jogo. Com exceção da quarta questão, essas questões mostraram resultados satisfatórios, como pode ser observado na Tabela 4.b. Porém, observando os demais resultados, a taxa de acerto nas questões 2, 3, 5 e 6 foi abaixo do esperado. Estas questões se referiam a conteúdos presentes em cartas específicas do jogo, e como o tempo da dinâmica foi menor do que a duração de uma partida normal, os participantes deixaram de ver essas cartas.

4. Conclusão

Este artigo propôs o jogo JEEES, criado para ensinar conceitos de Engenharia de Software com foco em GC. Durante a concepção do JEEES houve a preocupação de aprimorar algumas características herdadas do jogo Simules, o qual nos serviu de inspiração, por exemplo: (i) alterar o sistema de compras de cartas e aumentar a quantidade de cartas para que as cartas do jogo sejam suficientes para completar uma partida, (ii) renovar o *design* das cartas para que a disposição do conteúdo da carta e seu efeito constituam um único texto, evitando que o jogador leia somente o efeito da carta e ignore o conteúdo que está sendo informado, (iii) retirar a possibilidade de jogadores lançarem problemas para outros jogadores, (iv) implementar o pagamento de salário das equipes por turno. Além disso, como dito anteriormente, o foco do jogo foi no ensino da disciplina de GC. Dentro do conhecimento dos autores, esse é o primeiro jogo com esse intuito.

Outro ponto relevante foi a avaliação, ainda que preliminar, da versão atual do jogo. O estudo mostrou que os alunos receberam bem o jogo, inspirando o aprimoramento deste trabalho.

Trabalhos futuros consistem na aplicação de novos testes para avaliação do jogo e na criação de novos conjuntos de cartas para o ensino de outras áreas da Engenharia de Software, além do refinamento do conjunto de cartas de GC desenvolvido, de forma que englobe conteúdos de todas as funções da GC. Com um deque maior de cartas, será possível aplicar o jogo com mais participantes, podendo inclusive estender para jogadores em rede. Finalmente, a criação de uma versão digital do jogo JEEES está em andamento, já que 74% dos alunos participantes citaram esse interesse.

Agradecimentos

Gostaríamos de agradecer às professoras Aline Vasconcelos e Viviane Torres, que viabilizaram os estudos experimentais, e ao CNPq pelo apoio financeiro.

Referências

- Bendix, L. (2001) "Experience from Teaching Configuration Management" In: International Workshop on Software Configuration Management (SCM), Toronto, Canada.
- Dart, S. (1991) "Concepts in Configuration Management Systems" In: International Workshop on SCM, Trondheim, Norway: ACM Press, 1-18 p.
- Figueiredo, E., Lobato, C., Dias, K., Leite, J. e Lucena, C. (2006) "Simules: Um Jogo para o Ensino de Engenharia de Software". Relat. Técnico 34/06, Depto de Informática, PUC-Rio.
- Gros, B. (2003) "The Impact of Digital Games in Education", http://www.firstmonday.org/issues/issue8_7/xyzgros/index.html, Acesso em: 18 de Nov de 2009.
- Kieling, E. e Rosa, R. (2006) "Planager - Um Jogo para Apoio ao Ensino de Conceitos de Gerência de Projetos de Software", Monografia de Trabalho de Conclusão de Curso de Graduação, Ciência da Computação, FACIN, PUCRS, Porto Alegre.
- Lino, J. (2007) "Proposta de um Jogo Educacional para Medição e Análise de Software", Trabalho de Conclusão de Curso, Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis.
- MCT (2006) "Qualidade e Produtividade no Setor de Software Brasileiro". Brasília, DF, Ministério de Ciência e Tecnologia, Secretaria de Política de Informática.
- Navarro, E. (2006) "SimSE: A Software Engineering Simulation Environment for Software Process Education", Doctoral Dissertation, Donald Bren School of Information and Computer Sciences, University of California, Irvine.
- Savi, R. (2008) "Jogos Digitais Educacionais: Benefícios e Desafios", In: Revista de Novas Tecnologias na Educação, UFRGS, Porto Alegre, V. 6 N° 2.
- Silva, L. (2009) "O Jogo na Educação", In: Quaderns Digitals nº59, Departamento de Educação e Psicologia, Universidade de Trás-dos-Montes e Alto Douro, Portugal.
- Tarouco, L., Roland, L., Fabre, M. e Konrath M. (2004) "Jogos Educacionais", In: Revista de Novas Tecnologias na Educação, UFRGS, Porto Alegre, V. 2 N° 1.