

2 Jogos Educacionais

Jogos estão presentes como uma prática habitual, eles tem sido concebidos como uma atividade lúdica que é bastante motivadora no processo de ensino-aprendizado. É assim que jogos educacionais podem se transformar em uma alternativa importante neste processo. Os jogos escolhidos apresentados a seguir são uma pequena amostra de jogos educacionais para ensino na engenharia de software que têm sido integrados em atividades curriculares, assim como também temos identificado em alguns deles sua melhora ou evolução como ferramentas de ensino tradicional em leituras, provas, tarefas e simulações. A idéia central é analisar seus objetivos, atividades que realiza e métodos usados para suas modelagens. Estes jogos possuem uma semelhança com o jogo SimulES, além de serem jogos para o ensino na engenharia de software.

2.1. Visão Geral de Jogos Educacionais

Dentro dos métodos de ensino da engenharia de software em geral, se tem aulas teóricas e trabalhos práticos. Há outros métodos menos explorados como os jogos que podem completar o ensino tradicional. Em muitas áreas da tecnologia, os jogos são usados como uma ferramenta de ensino, mas isso é raro no ensino de engenharia de software. Na literatura encontramos alguns exemplos para ensinar engenharia de software que serão apresentados a seguir. Os jogos escolhidos, além de nos ajudar na contextualização de jogos educacionais, também serviram como exemplo para entendermos como jogos educacionais estão sendo modelados:

2.1.1. O Jogo Problems and Programmers (PnP)

- i. Objetivo do Jogo

É jogo de cartas educacional direcionado para o ensino de engenharia de software [1], foi desenvolvido no Departamento de Informática e Ciências da Computação da Universidade da Califórnia, Irvine (UCI). Alguns dos problemas e

conceitos de engenharia de software que são abordados neste jogo são: a) Falta de dedicar um tempo adequado para a documentação no início do projeto vai gerar problemas mais tarde, b) quando se tem uma codificação precipitada o projeto muitas vezes pode demorar mais tempo, c) quanto mais cedo os erros são encontrados seus impactos serão menos negativos para o projeto, d) inspeções de código reduzem tempo e quantidade de ciclos, além de permitir a detecção precoce de erros, e) desenvolvedores habilidosos podem causar problemas se eles são irresponsáveis ou se eles não seguem as boas práticas da engenharia de software, f) adicionar mais desenvolvedores para um projeto muitas vezes pode atrasá-lo.

ii. Objetivo no Jogo:

Seu objetivo é simular o processo de desenvolvimento de sistemas desde a concepção até a fase de entrega do software de uma maneira competitiva entre jogadores.

iii. Atividades realizadas no Jogo

No PnP os jogadores competem para terminar um projeto (carta projeto) usando cartas de conceito, evitando os perigos potenciais da engenharia de software (cartas problema). Com isso os jogadores irão aprender rapidamente que as estratégias que vão deixá-los ganhar o jogo podem ser as mesmas que irão ajudá-los no mundo real.

iv. Representações utilizadas para modelar o Jogo

Este jogo não possui implementação nem modelagem, seus artefatos, que são impressos em papel e podem ser baixados da página oficial do jogo [1], foram refinados com as retroalimentações recebidas. O PnP é importante para nós pois é a base de alguns dos jogos apresentados neste capítulo, além do jogo SimulES, nosso foco principal nesta pesquisa. Tanto o PnP como os jogos derivados dele vem sendo utilizados e alguns deles já foram implementados como apresentaremos a seguir.

2.1.2. O Jogo **SESAM**

i. Objetivo do Jogo

SESAM (*Software Engineering Simulation by Animated Models*, por sua sigla em Inglês) [2] é focado no ensino de gestão de projetos através de uma simulação na qual se pretende:

- Demonstrar e explicar como os recursos utilizados, ou a abordagem de gestão adotada em um projeto podem influenciar seus resultados globais.
- Examinar as conseqüências da mudança do processo ou os recursos.
- Permitir treinar futuros gerentes de projeto, expondo-os à realidade sobre problemas e situações típicas nos projetos.
- Permitir adaptação para ambientes de desenvolvimento específicos, a fim de apoiar o planejamento do projeto e controle do projeto.

ii. Objetivo no Jogo

A idéia básica do jogo é criar um modelo do processo de desenvolvimento de software e executá-lo usando um sistema de simulação. O modelo é complementado por um cenário do projeto inicial. Assim, projetos de software com uma determinada tarefa, ou determinados recursos ou restrições podem ser simulados. O progresso do projeto simulado se reflete quantitativamente utilizando vários processos e atributos do produto. Após a conclusão da simulação tanto o processo como também os produtos resultantes podem ser analisados.

iii. Atividades realizadas no Jogo

Na Figura 7 ilustramos a tela principal do jogo, aqui podemos ver que no menu principal do lado esquerdo estão descritas as ações possíveis dentro do jogo, que são: carregar o modelo e os estados, salvar os estados e imprimir os estados e as configurações. Tendo as informações prontas o jogador inicia a simulação.

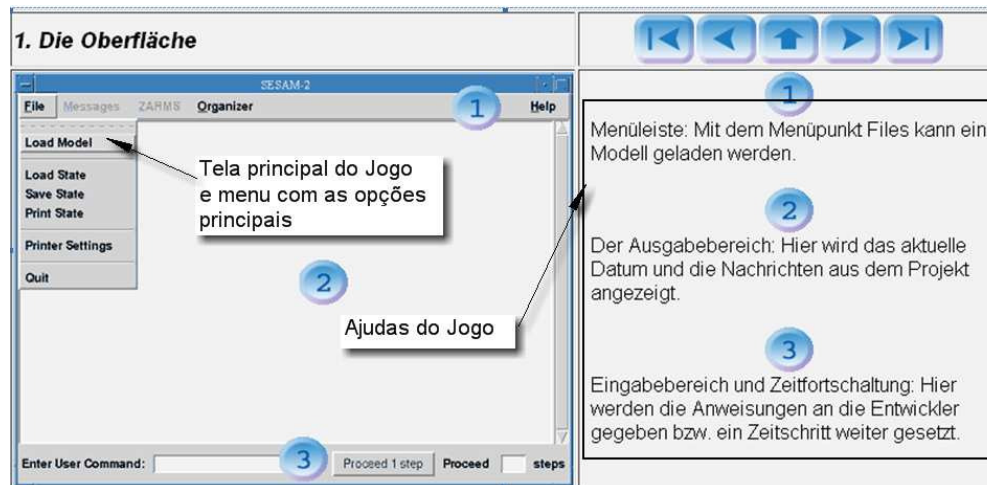


Figura 7 – Tela principal do jogo SESAM [2].

iv. Representações utilizadas para modelar o Jogo

Para sua modelagem foram criadas regras dentro da linguagem, ou seja, regras que após de ser codificadas e executadas implementarão a simulação, dividido em três partes: esquema, as condições e regras. Estudantes da Universidade de Stuttgart foram responsáveis pelo desenvolvimento do software e criação de documentos, incluindo: especificação, arquitetura, desenho e definição e implementação de ambiente de trabalho, entre outros. Este projeto foi focado em orientação a objetos e para sua implementação foi utilizado ADA95.

2.1.3. O Jogo *SimVBSE*

i. Objetivo do Jogo

Este jogo está focado no ensino do valor da engenharia de software [3], seus autores justificam-se na premissa de que grande parte da engenharia de software atual, tanto prática como de pesquisa, é feita ou baseada em um valor de definição neutra. Assim:

- Todo requisito, caso de uso, objeto, caso de teste e defeito são tratados como igualmente importantes.
- Métodos são apresentados e praticados amplamente como atividades lógicas que envolvem mapeamentos e transformações (por exemplo, desenvolvimento orientado a objeto).

- O “valor ganho” é relacionado ao custo do sistema e do cronograma, e não das partes interessadas ou do valor do negócio;
- Uma "separação de interesses" é praticada. As responsabilidades dos engenheiros de software são limitadas em transformar os requisitos do software em código verificado.

A maioria dos estudos do conceito “fator crítico de sucesso” são centrados em descobrir qual é o valor básico e o que os interessados identificam como predominante para o sucesso de um software. O objetivo deste jogo é integrar esse **valor** considerando-lo nos princípios e práticas de engenharia de software.

ii. Objetivo no Jogo

Como jogo está focado no ensino do que eles definem como **Valor** crítico de sucesso num processo de engenharia de software [3], os jogadores devem identificar os interessados no sistema com aquilo que eles entendem como fatores críticos de sucesso e/ou valores de preferência dentro de uma configuração simulada. Do mesmo modo, deve-se determinar uma estratégia para equilibrar esses valores definidos com os interessados identificados.

iii. Atividades realizadas no Jogo

Envolve principalmente identificar os interessados e seus valores de preferência; raciocinar sobre as compensações e os diferentes aspectos do projeto e do produto utilizando os valores considerados, e identificar um equilíbrio que satisfaça os valores de preferência junto com os interessados.

iv. Representações utilizadas para modelar o Jogo

Seu desenvolvimento é baseado em protótipos, as informações são extraídas de projetos, especificações e desenho de produto e se criam cenários no ambiente. Os estudantes envolvidos do projeto *SimVBSE* se encarregam de evoluir cada um dos protótipos criados.

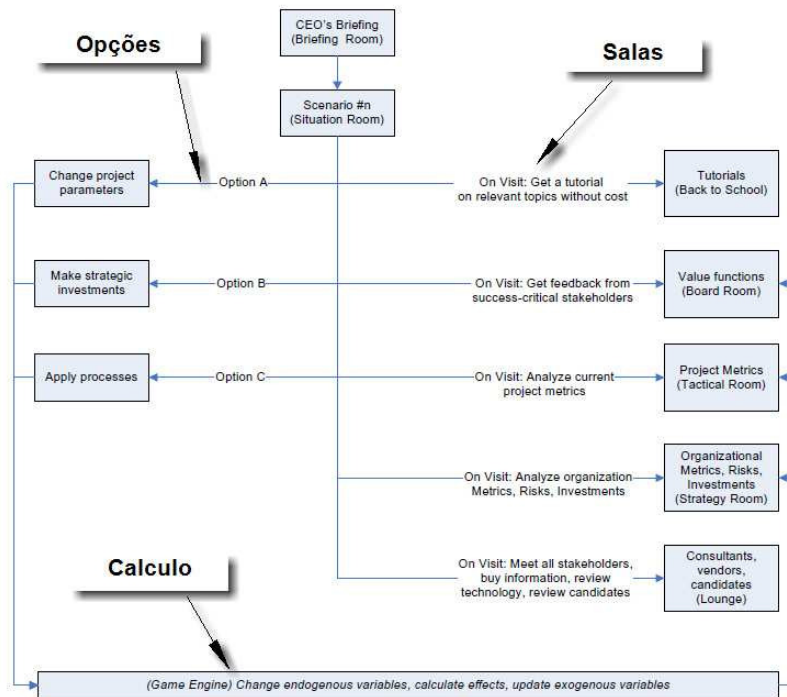


Figura 8 – Fluxo do jogo SimVBSE [3].

Na Figura 8 proporciona-nos uma ilustração de alto nível da estrutura dinâmica do jogo. Fazer um movimento no jogo envolve visitar diferentes salas que são mostradas na Figura na parte direita, e habilitar as opções que aparecem na parte esquerda da Figura. Para o jogador, visitar as salas não consome recursos, porém cada opção na esquerda pode custar uma quantia fixa de recursos.

2.1.4. O Jogo *SimSE*

i. Objetivo do Jogo

Permitir aos alunos praticar de forma “virtual” o processo de engenharia de software (ou seus sub-processos) de uma forma totalmente gráfica [13], além disso, permite saber a causa e efeito das complexas relações que permeiam os processos da engenharia de software.

ii. Objetivo no Jogo

Completar um projeto de engenharia de software, no qual o jogador assume o papel de gerente de projeto de uma equipe de desenvolvedores, O jogador tem

por objetivo concluir o processo de produção, assim, ele pode contratar e demitir engenheiros, atribuir tarefas, acompanhar seu progresso e comprar ferramentas.

iii. Atividades realizadas no Jogo:

Os alunos são orientados através da simulação e têm que lidar com o orçamento, o tempo e as dificuldades típicas de um projeto na medida em que a simulação está rodando, para isso os estudantes devem tomar decisões que podem afetar positiva ou negativamente o projeto. No progresso desta simulação o aluno pode ver os elementos do ambiente, bem como as informações sobre os engenheiros (produtividade, tarefa atual e nível de energia), artefatos (tamanho, integridade e precisão), clientes (nível de satisfação), projetos (orçamento e tempo), ferramentas (numero de usuários, fator de aumento da produtividade). A interação do jogador com os engenheiros é através de mensagem que o sistema apresenta aleatoriamente. Estas informações servem para que o jogador tome decisões e ações. No final do jogo o jogador recebe uma pontuação que indica quão bem foram realizadas as atividades. Além de toda esta informação, uma ferramenta explicativa fornece ao jogador mais informações sobre o seu jogo, incluindo regras que foram desencadeadas, rastro dos eventos, e informações de código versus tempo. Além disso, o jogador pode executar vários ramos paralelos de um jogo e retornar a qualquer ponto anterior, gerar uma nova simulação ou continuar com a simulação anterior.

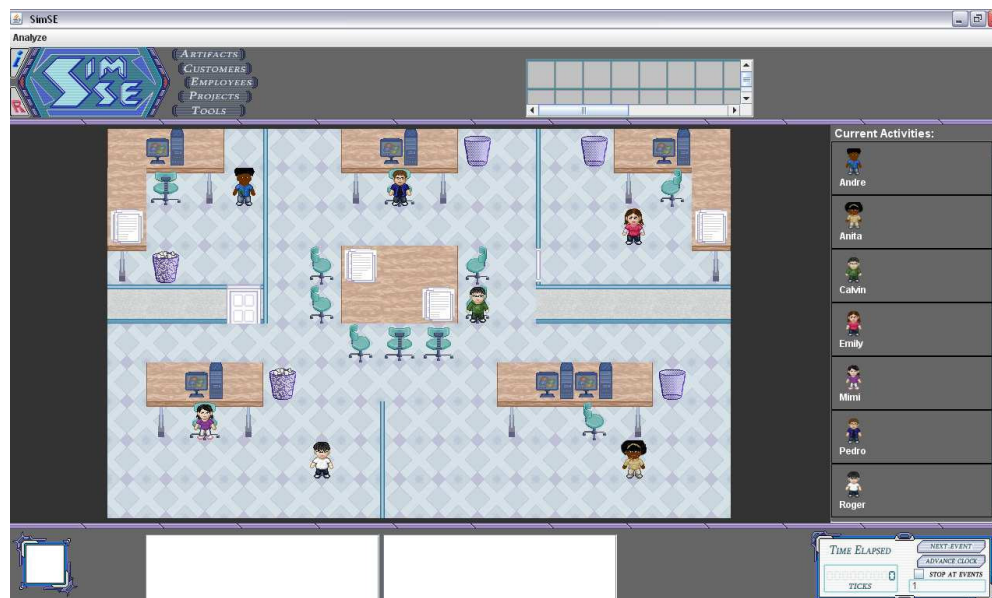


Figura 9 – Tela principal do jogo SimSE tomado de [13].

A Figura 9 mostra a tela principal do jogo SimSE, na parte direita ficam os engenheiros de software que podem ser contratados durante a simulação, na parte superior esquerda são apresentadas as condições iniciais e na parte inferior direita os elementos para executar a simulação.

iv. Representações utilizadas para modelar o Jogo

SimSE tem diferentes versões as quais estão relacionadas a diferentes processos de desenvolvimento de software, ele possui versões baseados nos processos espiral, cascata, prototipagem e RUP, ou seja, cada versão ensina um processo que também foi usado para criar seus modelos.

2.1.5. O Jogo *Planager*

i. Objetivo do Jogo

É um jogo para apoiar o ensino de conceitos de gerência de projetos [30] de uma forma prática e iterativa focado para alunos de graduação.

ii. Objetivo no Jogo

Simular alguns dos processos utilizados na gerência de projetos, principalmente processos de planejamento. Como gerenciamento do escopo e gerenciamento do tempo. Na gerência de escopo foram escolhidos os processos de definição do escopo e criação da EAP (Estrutura Analítica do Projeto). Na gerência de tempo foram escolhidos os processos de definição da atividade, seqüenciamento de atividades e desenvolvimento do cronograma (com foco no cálculo do caminho crítico).

iii. Atividades realizadas no Jogo

Segundo [30] o jogador pode usar o modulo tutorial para aprender gerencia de projetos ou pode jogar em diferentes cenários cadastrados na base de dados do jogo, estes cenários são representações de projetos e cada um tem uma descrição e cinco fases: fase escopo, fase EAP, fase definição de atividades, fase de seqüenciamento de atividades e fase de caminho crítico. Os cenários são seqüenciais onde cada cenário ou fase utiliza as informações dos cenários

anteriores. A seqüência destas fases é a mesma utilizada pelos gerentes de projetos quando planejam o escopo e tempo de um projeto.

iv. Representações utilizadas para modelar o Jogo

Para a sua modelagem foi utilizado UML. O sistema a ser construído foi representado em casos de uso, diagramas de atividades, cenários e diagrama de classes. É uma aplicação desktop de arquitetura cliente/servidor desenvolvida usando JAVA JDK 5, possui dois tipos de usuários onde o usuário administrador pode configurar novos cenários para serem executados na simulação.

2.1.6. O Jogo *Scrumming*

i. Objetivo do Jogo

Ensinar, através de simulação, as práticas ágeis de gerenciamento de projetos, principalmente o SCRUM. Segundo [29] SCRUM é “um processo ágil que permite manter o foco na entrega de maior valor de negócio, no menor tempo possível. Isto permite a rápida e contínua inspeção do software em produção (em intervalos de duas a quatro semanas conhecidos como *sprints*).”

ii. Objetivo no Jogo

Durante a simulação o jogador age como se fosse um SCRUM Master, realizando tarefas como definir um sprint, monitorar o andamento do sprint através do *taskboard*, visualizar o gráfico de *burndown*, adicionar ou remover atividades do *backlog*, entre outras.

iii. Atividades realizadas no Jogo

O *Scrum Master* que se encarrega de aplicar os valores e práticas do Scrum, remover obstáculos, garantir a plena funcionalidade e produtividade da equipe, e garantir a colaboração entre os diversos papéis e funções.

iv. Representações utilizadas para modelar o Jogo

Para a sua modelagem foi utilizado UML, o sistema a ser construído foi representado em casos de uso, diagramas de atividades e modelo entidade-relacionamento. É uma aplicação desktop de arquitetura cliente/servidor

desenvolvida usando JAVA JDK 6 e base de dados MySQL 5.0. Possui dos tipos de usuários, o usuário administrador é um *Scrum Master*.

2.1.7. O Jogo X-MED v1.0

i. Objetivo do Jogo

Ensinar a medição de software [31]. Para alunos de pós-graduação em cursos de computação e/ou profissionais da Engenharia de software.

ii. Objetivo no Jogo

Simular um programa de medição voltado a gerência de projetos alinhado ao nível de maturidade 2 do CMMI-DEV [68].

Através da seleção de soluções adequadas de tarefas de medição, os alunos são incentivados a aprender como desenvolver ou selecionar objetivos de medição. O estudante assume o papel de um analista de medição e segue sequencialmente todas as tarefas de definir e aplicar um programa de medição.

iii. Atividades realizadas no Jogo

O jogo é composto das seguintes fases: fase 1 introdução ao jogo, fase 2 execução do jogo contendo os passos: passo 1 caracterização de contexto, passo 2 identificação do objetivo de medição, passo 3 desenvolvimento do plano GQM, passo 4 Desenvolvimento do plano de coleta de dados, passo 5 Verificação de dados, passo 6 Análise de dados, passo 7 interpretação de dados e por último, na fase 3, temos a finalização do jogo. Em conclusão, o aluno não compete por ganhar o jogo a idéia do jogo é a interpretação e análise dos dados gerados pela simulação.

iv. Representações utilizadas para modelar o Jogo

Para sua representação basicamente foram usadas as classes do sistema, Segundo [31] o jogo foi desenvolvido a partir de uma arquitetura em camadas, utilizando classes de limite (interface com o usuário), classes de controle (regras de negócio e fluxos de controle do programa) e classes de entidade (armazenamento dos dados). É uma aplicação desktop de arquitetura cliente/servidor desenvolvida usando JAVA JDK 6.

2.1.8.O Jogo SimulES

i. Objetivo do Jogo

Jogo de tabuleiro educacional direcionado para o ensino de engenharia de software, este jogo foi concebido a partir da evolução do jogo "*Problems and Programmers*" (PnP) [1], que é também a base de alguns dos jogos acima referidos nas seções 2.1.2, 2.1.3 e 2.1.4. Na sua evolução foram acrescentados conceitos de evolução do software, rastreabilidade e documentação de ajuda.

ii. Objetivo no Jogo

Construir um produto de software, onde o aluno assume o papel de gerente de projeto e deve lidar com o orçamento do projeto, contratação e demissão de engenheiros, e construção de diversos artefatos necessários para a culminação de um projeto de software, entre muitas outras atividades. O jogador deve criar uma estratégia que irá melhorar o seu jogo em relação à dos seus adversários, ao mesmo tempo deve fazer jogadas que desestabilizem os jogos de seus adversários. Mas eles podem responder com conceitos de engenharia de software e bloquear o ataque do jogador. O primeiro jogador a construir todos os módulos ganha a partida.

iii. Atividades realizadas no Jogo

Os jogadores executam uma série de ações chamadas “Rodadas”, estas rodadas estão dispostas sequencialmente e todos os jogadores devem participar delas de forma ordenada. O jogo possui as seguintes rodadas: início, ações, conceitos, tratamento de problemas e submeter produto. Informações mais detalhadas das rodadas estão presentes no Capítulo 3.

iv. Representações utilizadas para modelar o Jogo

Técnica de léxicos e cenários foram usadas para modelar as diferentes evoluções de SimulES. No Capítulo 4 detalharemos essas representações.

2.2. Resumo das Características dos Jogos

Na tabela 2 fazemos uma síntese dos elementos mais importantes dos jogos aqui apresentados, ressaltado o jogo *PnP* por ser a base do SimULES além de outros jogos. O jogo SimSE também tem algumas semelhanças com o SimULES como o objetivo dentro do jogo e o objetivo do jogo, somente que SimSE foi projetado para um só jogador. Este jogo também possui uma documentação detalhada sobre os diferentes modelos criados.

Tabela 2 – Resumo das características dos jogos.

Nome do Jogo	Objetivo do jogo	Objetivo no Jogo	Modelagem do jogo
1. Problems and Programmers (PnP)	Ensinar engenharia de software.	Simular o processo de desenvolvimento de software em cascata.	Não contém
2. SESAM	Ensinar de gestão de projetos.	Criar um modelo do processo de desenvolvimento de software e executá-lo usando um sistema de simulação.	Documentação de especificação, arquitetura, desenho e definição e implementação de ambiente de trabalho.
3. SimVBSE	Ensinar do valor da engenharia de software.	Identificar os interessados no sistema com aquilo que eles entendem como <i>fatores críticos de sucesso e valores de preferência</i> dentro de uma configuração simulada.	Desenvolvimento baseado em protótipos.
4. SimSE	Ensinar processo de engenharia de software.	Completar um projeto de engenharia de software.	Possui diferentes modelagens conforme à versão do jogo.
5. Planager	Ensinar de conceitos de gerência de projetos.	Simular alguns dos processos utilizados na gerência de projetos, principalmente processos de planejamento.	Orientada a objetos com UML.
6. Scrumming	Ensinar através de simulação as práticas ágeis de gerenciamento de projetos, principalmente o SCRUM.	Fazer uma simulação assumindo o papel de SCRUM Master.	Orientada a objetos com UML.
7. X-MED v1.0	Ensinar a medição de software.	Simular um programa de medição voltado a gerência de projetos alinhado ao nível de maturidade 2 do CMMI-DEV.	Baseada na sua arquitetura, ou seja, foi desenvolvido a partir de uma arquitetura em camadas.

2.3. Considerações Sobre Jogos Educacionais

Acreditamos que jogos podem despertar maior interesse por parte do aluno e ajudar no aprendizado. Pesquisas cognitivas também sugerem que a percepção humana e a ação estão profundamente interconectadas [32]. Desta forma, o uso de jogos para treinar, aprender e executar atividades que imitam a realidade pode melhorar o desempenho dos alunos, pois possibilita a vivência de experiências de aprendizagem que não são fornecidas de forma teórica.

Os jogos explorados neste Capítulo nos dão uma idéia de como são os jogos propostos para serem parte das atividades curriculares, ajudando a melhorar as

ferramentas tradicionais de ensino, como leituras, provas, tarefas, simulações entre outras. Estes jogos refletem distintos modelos pedagógicos, gêneros de jogos, plataformas e usos.

Além disso, estes jogos representam de modo geral o feito até agora na tecnologia de jogos educacionais para ensino na engenharia de software e as abordagens atuais de modelagem e desenho.

Com a revisão desta literatura alguns questionamentos além dos apresentados neste capítulo surgem, como por exemplo: econômicos (qual seria o papel do software livre?), sociais (como entregaremos estes produtos, quem serão os encarregados de treinar e como os professores integraram estes de forma significativa a suas atividades curriculares?) e políticos (como serão incorporados os jogos nos ambientes de aprendizado?), em resposta a estas oportunidades, esta dissertação pretende evoluir o Jogo SimuLES, utilizando uma modelagem intencional apoiado em conceitos de transparência, e assim explorar estas perspectivas e estes questionamentos e de algum modo criar uma experiência que possa servir de base para futuras pesquisas.