Especificando um Modelo de Time para o Desenvolvimento Colaborativo de Jogos Educativos

André Wilson Brotto Furtado, André Luís de Medeiros Santos, Alex Sandro Gomes

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Prof. Luís Freire, s/n, Cidade Universitária, CEP 50740-540 – Recife/PE/Brazil

{awbf,alms,asg}@cin.ufpe.br

Resumo. *Este artigo tem como objetivo propor um modelo de time para o desenvolvimento colaborativo de jogos educativos, especificando papéis para os indivíduos envolvidos no ciclo de vida deste tipo de software e estabelecendo responsabilidades bem-definidas para cada papel. Tal modelo de time visa combater limitações conhecidas de jogos educativos, como a falta de envolvimento do usuário final. Além de papéis e responsabilidades, o modelo especifica artefatos, áreas funcionais e princípios para enriquecer as atuais metodologias de desenvolvimento de software na produção de jogos educativos. Um jogo do tipo "perguntas e respostas" é apresentado, como um estudo de caso, de modo a ilustrar e validar o modelo proposto.*

Palavras-chave: jogos educativos, modelo de time, papéis, responsabilidades, processo de desenvolvimento de software, colaboração.

Abstract. *This paper presents a team model for collaborative educative game development processes. It specifies roles to the individuals involved in such kind of software lifecycle and establishes well-defined responsibilities to each one of them. The proposed model intends to avoid some already known limitations of educative games, such as the lack of involvement of the final user. Besides roles and responsibilities, the proposed team model specifies artifacts, functional areas and principles to enhance common software development methodologies in the process of creating educative games. A quiz game is also presented, as a case study, in order to illustrate and validate the proposed idea.*

Key words: educative games, team model, roles, responsibilities, software development processes, collaboration.

Especificando um Modelo de Time para o Desenvolvimento Colaborativo de Jogos Educativos

1. Introdução

Por sua necessidade intrínseca de unir diversão a aprendizado, jogos educativos constituem um desafio bastante complexo no que diz respeito à aceitação final do usuário. Enquanto projetistas e desenvolvedores de jogos não-educativos estão focados em um produto final cujo sucesso é determinado majoritariamente pelo grau de envolvimento com o usuário (seja através de gráficos realistas, sons imersivos ou um roteiro intrigante), equipes envolvidas na elaboração de jogos educativos deparam-se com um obstáculo a mais: fazer com que o tópico de aprendizagem seja assimilado com sucesso pelo jogador. Entretanto, uma bem-sucedida assimilação do conteúdo educacional pode ir de encontro a requisitos básicos de jogos de computador, como diversão, jogabilidade e *replay value* (desejo do jogador em jogar repetidas vezes o jogo). Em outras palavras, o desenvolvimento de jogos educativos requer um cuidado extra: encontrar um equilíbrio coerente entre diversão e aprendizado de modo a evitar que um prejudique o outro.

Carraher [1990] fornece os subsídios necessários à identificação de dois objetivos distintos de jogos educativos (diversão e aprendizado), ao afirmar que são dois os contextos nos quais atua a qualificação de software educativo: o contexto educacional, ou pedagógico, e o contexto técnico-computacional. O autor expressa sua preocupação em relação à ênfase dada à avaliação técnica e computacional, enquanto a mesma deveria estar subordinada à avaliação educacional e pedagógica. A dificuldade da inserção do elemento pedagógico em softwares educativos é verificada em estudos do autor, que revelam que "a produção de software de qualidade técnica e, mais ainda, de qualidade pedagógica, é mais complexa do que se imaginava há pouco tempo, de tal modo que dificilmente surgirá, de forma rápida e espontânea uma quantidade de software de qualidade.".

Dez anos após os estudos de Carraher mencionados acima, suas teorias são concretizadas por pesquisas envolvendo o público-alvo de jogos educativos. Clua, Luca e Nabais [2002] conduziram uma série de entrevistas com jogadores na faixa etária de 10 a 17 anos, com o objetivo de identificar o que eles pensam e esperam de jogos educativos. Os resultados retratam um cenário bastante pessimista em relação ao tema: nenhum dos jogadores entrevistados afirmou que existem jogos educativos "excelentes", ao passo que 68% deles consideraram todos os jogos educativos já experimentados como sendo "terríveis". Uma pequena parcela dos entrevistados (5%) se declararam indiferentes em relação ao tema em discussão e apenas 27% dos jogadores afirmaram que já jogaram pelo menos um jogo educativo que pudesse ser considerado como "bom". Os autores da pesquisa identificaram quatro motivos principais para explicar tal resultado:

- Jogos educativos possuem deficiências em apresentar desafios complexos e interessantes;
- O jogador de jogos educativos não é envolvido o suficiente por jogos deste tipo;
- A maioria dos jogos educativos é desenvolvida apenas por pedagogos, que tomam cuidado excessivo com o tópico de aprendizagem e deixam a diversão do jogo em segundo plano;
- Não há um investimento expressivo em projetos de jogos educativos, na maioria das vezes porque os *publishers* não consideram esta área um negócio rentável.

Os autores deste artigo defendem a idéia de que as dificuldades expostas acima se originam ainda na fase de concepção, principalmente devido ao fato de não haver uma coordenação adequada entre os indivíduos envolvidos no processo de criação de desenvolvimento de jogos educativos. Tchounikine [2002], por exemplo, ressalta a dificuldade de interação entre programadores e educadores, devido à não-trivialidade de compartilhar conceitos das diferentes áreas. Mandel [1997], por sua vez, identifica problemas causados pela significativa diferença que designers, programadores e professores possuem em relação aos processos de ensino e aprendizagem.

Este artigo, portanto, tem como objetivo propor um *modelo de time* para o desenvolvimento colaborativo de jogos educativos, especificando papéis para os indivíduos envolvidos no ciclo de vida

deste tipo de software e estabelecendo responsabilidades bem-definidas para cada papel. Através de uma divisão coerente de objetivos, o modelo pretende atacar algumas das principais deficiências constatadas no processo de desenvolvimento de jogos educativos, permitindo que o time de projeto evite armadilhas comuns no desenvolvimento deste tipo de software. O modelo apresentado poderá atuar como um guia adicional para enriquecer as atuais metodologias de desenvolvimento de software, quando o produto final em questão se tratar de um jogo educativo.

Este trabalho encontra-se organizado da seguinte maneira: a seção 2 fornece os subsídios necessários para a compreensão do modelo de time proposto. Na seção 3, o modelo de time é descrito em detalhes. Na seção 4, um caso de estudo é apresentado, com o objetivo de ilustrar e validar o modelo: um jogo educativo simples do tipo "perguntas e respostas" (*quiz game*). A seção 5, por fim, realiza uma crítica conclusiva acerca do modelo proposto.

2. Considerações Iniciais para o Modelo de Time Proposto

Antes de apresentar e detalhar o modelo, faz-se necessário entender o que é um modelo de time. Posteriormente, esta seção apresenta os requisitos levados em consideração para a elaboração do modelo proposto.

2.1 Conceituação de Modelos de Time

Dado um certo contexto colaborativo, como, por exemplo, o processo de desenvolvimento de software, um modelo de time define papéis, áreas funcionais, responsabilidades, e princípios para ajudar os membros do time a gerir a concretização bem-sucedida do projeto.

Por time entende-se a equipe responsável pela realização das etapas do ciclo de vida do projeto, desde sua concepção até a sua implantação. Possivelmente, o contrato estabelecido com o cliente pode exigir manutenção e/ou atualização do produto depois da implantação. Neste caso, devem integrar a equipe indivíduos capazes de contemplar essas atividades pós-implantação, em que o projeto já está operacional.

Cada indivíduo da equipe possui objetivos específicos, portanto, a cada indivíduo é atribuído um *papel*. Cada papel define um conjunto de *responsabilidades* que devem ser realizadas por seus integrantes. No contexto específico de desenvolvimento de software, alguns exemplos de papéis seriam o Gerente do Projeto, o Desenvolvedor, o Testador e o Analista, entre outros. Como o Usuário Final e o Cliente do projeto têm parte nas decisões efetuadas durante o andamento do processo, eles também podem ser considerados papéis.

Um mesmo papel pode ser desempenhado por várias pessoas. Por exemplo, podem existir vários testadores em um mesmo projeto de software. Além disso, uma mesma pessoa pode desempenhar diferentes papéis: um indivíduo que é o analista de um projeto pode, eventualmente, participar da implementação do mesmo.

Áreas funcionais representam os campos de atuação de cada papel. Por exemplo, possíveis áreas funcionais nas quais agem o Testador de um projeto de software são o Planejamento de Testes, a Engenharia de Testes e a Reportagem do Status de Testes. Os *princípios* de um modelo de time, por sua vez, são filosofias que devem ser seguidas pela equipe. São idéias nas quais a equipe acredita e que contribuem para que as responsabilidades de cada papel sejam efetuadas de maneira correta e produtiva. Por exemplo, o princípio "Visão compartilhada da solução" contribui para que todos os integrantes de um processo de desenvolvimento de software entendam qual o objetivo final do projeto e saibam direcionar suas atividades específicas para a realização bem-sucedida desse objetivo final.

O modelo de time do *Microsoft Solutions Framework (MSF)*® [Microsoft® 2003], conhecido como *MSF Team Model*, é um exemplo prático que ilustra bem os conceitos apresentados acima. O MSF é um conjunto de princípios, boas práticas e guias para o planejamento, construção e entrega de soluções de tecnologia da informação, de acordo com o ponto-de-vista e a experiência da *Microsoft*®. O *MSF Team Model*, por sua vez, define uma estrutura para organizar as pessoas envolvidas no MSF em papéis e permitir o direcionamento das atividades de cada indivíduo de acordo com os objetivos específicos de seu papel.

A Tabela 1 apresenta uma caracterização resumida do *MSF Team Model*. Esta caracterização, embora informal, é útil não apenas para a solidificação dos conceitos acerca de modelos de time como também será interessante para propósitos de comparação com o modelo de time para jogos educativos proposto neste artigo.

Tabela 1. Exemplo de caracterização resumida de um modelo de time[1]

Nome	MSF Team Model
Contexto	Construção e entrega de soluções de tecnologia da informação segundo o ponto-de-vista e experiência da Microsoft®,
Papéis	Development, Testing, Program Management, Product Management, User Experience e Release Management
Algumas áreas funcionais	Treinamento e Design de interface (User Experience), Marketing (Product Management), Implementação (Development), etc.
Algumas responsabilidades	Facilitar comunicação entre o time (Program Management), Prover suporte logístico ao time (Release Management), etc.
Princípios	Vontade de aprender; Visão compartilhada da solução; Foco no cliente; Mentalidade de "defeito-zero"; Consideração integral do produto final

2.2 Requisitos do Modelo Proposto

A primeira etapa pela qual passou a especificação do modelo de time para o desenvolvimento colaborativo de jogos educativos proposto neste artigo foi um levantamento inicial de requisitos desse modelo. Em outras palavras, desejou-se saber o que seria necessário a um modelo de time para que ele oferecesse alternativas para combater limitações conhecidas de jogos educativos. A análise realizada revelou o seguinte conjunto de requisitos como resultado:

1. Os papéis definidos pelo modelo devem ser resultado de uma clara compreensão das necessidades dos indivíduos envolvidos no processo de desenvolvimento de jogos educativos;
2. O modelo deve contemplar uma solução para o dilema "diversão/aprendizado", através de uma alocação coerente de responsabilidades para os papéis envolvidos no dilema;
3. O modelo, na medida do possível, não deve permitir que o tópico de aprendizagem limite a criatividade inserida no jogo, de modo a estimular o envolvimento proporcionado pelo mesmo;
4. O modelo não deve estar atrelado a tecnologias específicas, o que dificultaria seu uso por indivíduos de menor capacitação técnica;
5. O modelo deve estimular e dar suporte à integração do trabalho dos indivíduos pertencentes aos diversos papéis definidos no mesmo;
6. O modelo deve possibilitar e estimular o reuso no desenvolvimento de jogos educativos, permitindo uma maior produtividade e um menor custo na elaboração dos mesmos.
7. O modelo não deve tornar o desenvolvimento de jogos educativos um processo burocrático. Pelo contrário, o modelo deve encorajar a popularização do processo.

Os requisitos descritos acima serão referenciados durante o detalhamento do modelo, apresentado na próxima seção.

[1] Como as áreas funcionais e responsabilidades do *MSF Team Model* são muito numerosas, optou-se por listar apenas alguns exemplos.

3. Apresentação do Modelo

O modelo proposto define quatro papéis para um time de desenvolvimento colaborativo de jogos educativos, cada um com seu conjunto de responsabilidades específicas: o *Jogador*, o *Desenvolvedor*, o *Educador* e o *Regulamentador*. Entretanto, a essência do modelo não gira em torno apenas de papéis e responsabilidades. De modo a satisfazer o requisito n° 5 do modelo (integração dos papéis definidos), foi decidido incluir *artefatos* no modelo, que representam elementos compartilhados e produzidos pelos papéis. Os artefatos são os pontos de entrada e de saída para a execução das atividades colaborativas entre o time. Os artefatos do modelo proposto são: *categorias (ou especificações) de jogos educativos*, *cenários de jogos educativos*, *entrada parametrizada para cenários de jogos educativos*, *cenários instanciados de jogos educativos* e *scripts/ferramentas auxiliares para educadores*. A articulação entre os papéis e os artefatos do modelo, cujo detalhamento é o objetivo desta seção, é ilustrada na Figura 1.

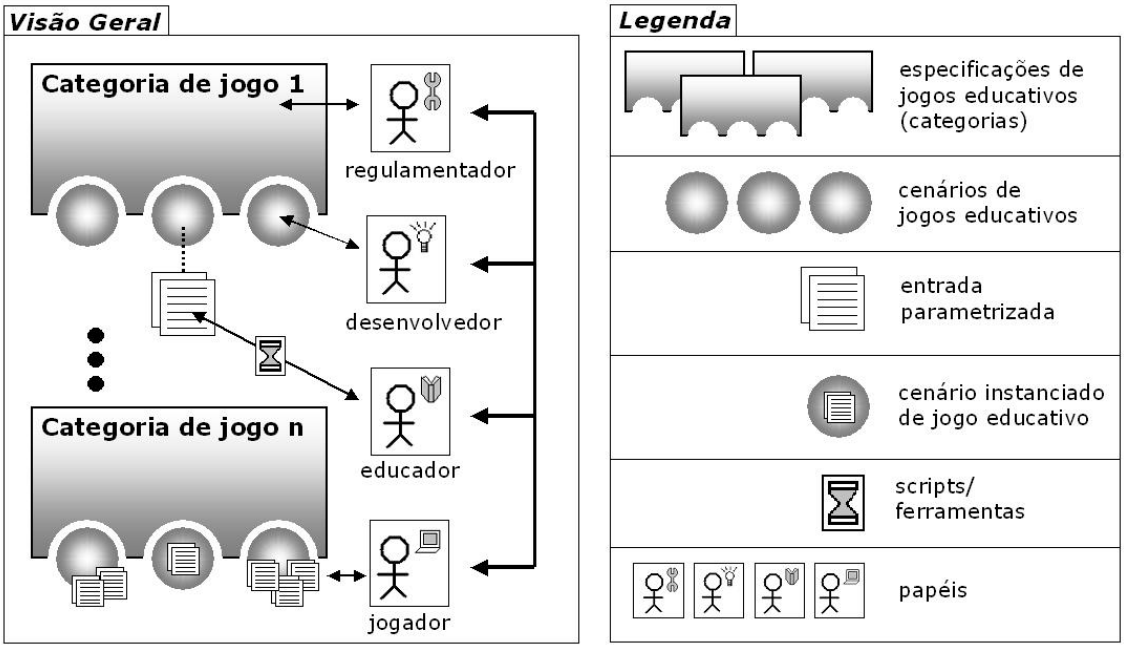

Figura 1. Visão geral do modelo de time proposto

Assim como qualquer modelo de time, o modelo proposto define *responsabilidades*, áreas *funcionais* e *princípios* a serem seguidos pelos papéis. Apesar de não serem exibidos na visão geral acima, o detalhamento do modelo revelará, gradualmente, cada um desses atributos.

A principal idéia defendida pelo modelo é que jogos educativos podem ser agrupados em diferentes *categorias*. Essas categorias não estão relacionadas a tópicos de aprendizagem, e sim ao estilo (ou tipo) do jogo. Por exemplo, jogos do tipo "pergunta e resposta" (*quizzes*) formam uma categoria, enquanto jogos do tipo "aventura" constituem outra categoria. O papel responsável por validar e alterar formalmente as categorias é o Regulamentador.O modelo requer que todo jogo educativo seja enquadrado em uma categoria. Isto se faz necessário porque uma categoria especifica um contrato entre dois papéis: o Desenvolvedor e o Educador. Este contrato será apresentado em mais detalhes a seguir.

O Desenvolvedor é o responsável pela construção de *cenários de jogos educativos*. Um cenário é, na verdade, um jogo educativo ainda incompleto. Apesar de conter gráficos, sons, enredo, gerenciamento de entrada do usuário e gerenciamento do fluxo do jogo, por exemplo, um cenário ainda não contém conteúdo educativo. Este conteúdo educativo deverá ser informado ao jogo pelo Educador, em tempo de execução, através de uma *entrada parametrizada*. É interessante observar que um cenário, por se tratar de um jogo educativo, mesmo que incompleto, deve ser enquadrado em alguma categoria de jogos educativos.

Tome-se como exemplo um jogo educativo da categoria *jogo da memória*[2]. O Desenvolvedor irá definir, ao seu desejo, se o jogador utilizará o mouse, o teclado ou outro periférico para interagir com o jogo. A existência e a configuração de sons, assim como dos gráficos do jogo também serão escolhidos pelo Desenvolvedor. O conjunto de todos esses aspectos técnicos e de implementação formará um cenário de jogo educativo. Entretanto, o conteúdo dos cartões (principal aspecto do jogo relacionado ao tópico de aprendizagem) será definido pelo Educador, e não pelo Desenvolvedor. Isto possibilita uma série de vantagens:

- Educadores de diferentes áreas poderão utilizar o mesmo cenário desenvolvido (o jogo da memória construído pelo Desenvolvedor), promovendo o reuso do jogo para diferentes tópicos de aprendizagem (requisito n° 6 do modelo). Um professor de Biologia, por exemplo, pode definir que, para cada par de cartões relacionados, o conteúdo de um dos cartões será o desenho de um órgão do corpo humano, enquanto o conteúdo do outro cartão será o nome do órgão. Um professor de Geografia, por sua vez, poderá utilizar o mesmo cenário, definindo que o conteúdo de um cartão será o nome de um país e que seu par correspondente conterá o nome de sua capital.
- A criatividade do Desenvolvedor não está limitada pelo conteúdo educativo do jogo, e sim à categoria escolhida por ele. Entretanto, caso o Desenvolvedor perceba que uma determinada categoria está restringindo seu potencial criativo, ele pode escolher outra categoria mais adequada (requisito n° 3 do modelo).
- A diversão e o aprendizado do jogo educativo são responsabilidades de papéis distintos, de modo que um não prejudique o outro (requisito n° 2 do modelo). De um lado, é responsabilidade do Desenvolvedor garantir o envolvimento e a aceitação do usuário final (o Jogador), através de sua capacidade técnica, experiência em implementação e domínio de ferramentas e bibliotecas para o desenvolvimento de jogos. De outro lado, o Educador, através de sua capacitação pedagógica e domínio do tópico de aprendizagem, é responsável por fornecer ao jogo um conteúdo educativo correto, nivelado e consistente.
- Nada impede que o Desenvolvedor crie um cenário já pensando em um tópico de aprendizagem específico. Por exemplo, um Educador da área de História pode requisitar a um Desenvolvedor a construção de um jogo da memória para o ensino da Segunda Guerra Mundial (requisito n° 5 do modelo). O Desenvolvedor, portanto, pode consultar o Educador para saber que sons e imagens seriam os mais adequados para um jogo com esse propósito educacional, de modo a inserir ainda mais o usuário final (o Jogador) no jogo.

Uma questão, entretanto, ainda não foi respondida: como garantir que a comunicação entre o Educador e o Desenvolvedor será clara, correta e produtiva, não ocorrendo de maneira *ad hoc*? O conceito de entradas parametrizadas é proposto pelo modelo para resolver tal problema. Uma entrada parametrizada, conforme introduzido anteriormente, consiste no conteúdo educativo através do qual o Educador customiza um cenário de jogo educativo. Cada categoria de jogo educativo, portanto, possui um tipo de entrada parametrizada formalmente especificada e validada pelo Regulamentador. Esta especificação estabelece um contrato entre o Desenvolvedor e o Educador: enquanto o Desenvolvedor constrói cenários capazes de interpretar uma entrada parametrizada em tempo de execução, o Educador deve criar entradas parametrizadas e alimentar tais cenários com elas. Para o modelo, um cenário alimentado com uma entrada parametrizada é chamado de *cenário instanciado de jogo educativo*. Um cenário instanciado, ao contrário de um cenário não-instanciado, é um jogo educativo completo, pronto para ser executado pelo Jogador.

[2] Jogo clássico em que é apresentado ao jogador um conjunto de cartões cujo conteúdo é desconhecido. O objetivo do jogador é agrupar, aos pares, os cartões cujos conteúdos estão relacionados, visualizando o conteúdo de apenas dois cartões por rodada.

Voltando ao exemplo da categoria "jogo da memória", o Regulamentador poderia especificar que o conteúdo dos cartões do jogo deve ser fornecido através de um arquivo XML[3] chamado "*jogoMemoriaEntrada.xml*", validado por um determinado Schema[4] ou DTD[5] previamente definidos. O arquivo "*jogoMemoriaEntrada.xml*", portanto, seria considerado a entrada parametrizada para jogos educativos da categoria "jogo da memória". Isto implica no fato de que um cenário desta categoria deve ser capaz de ler tal arquivo, interpretá-lo e utilizar adequadamente seu conteúdo.

Pode-se imaginar que o exemplo acima, utilizando a tecnologia XML, está em discordância com o requisito n° 4 do modelo: o Educador estaria sendo obrigado a dominar uma tecnologia específica para poder criar entradas parametrizadas. O modelo proposto, entretanto, atribui mais uma responsabilidade ao Regulamentador: garantir que Educadores sejam abstraídos da tecnologia utilizada na criação de entradas parametrizadas, através da validação e recomendação de ferramentas/scripts que tornem o processo transparente. A Figura 2, por exemplo, mostra um protótipo de ferramenta que auxilia Educadores na criação automática de arquivos XML como entrada parametrizada para cenários da categoria "jogo da memória", especificada acima. Uma ferramenta deste tipo permite que o Educador se concentre em **qual** será o conteúdo educativo que o jogo transmitirá ao Jogador, sem se preocupar **como** isso será realizado.

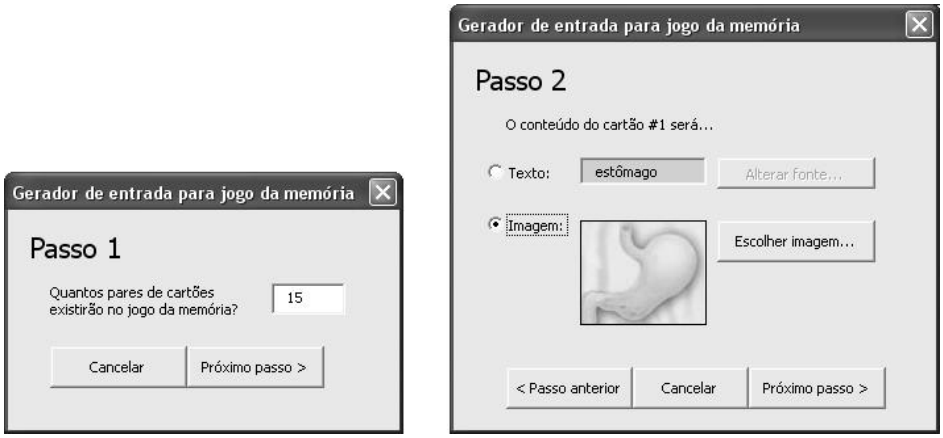

Figura 2. Protótipo de ferramenta para auxiliar Educadores na geração de entrada parametrizada para cenários da categoria *jogo da memória*

Uma vez que um cenário de jogo educativo esteja instanciado, ele está pronto para interagir com o Jogador. Este papel representa o usuário final do modelo, cuja diversão e aprendizado são os objetivos principais do processo de desenvolvimento de um jogo educativo. Segundo o modelo de time proposto, o Jogador interage com um cenário instanciado através de três maneiras: jogando, aprendendo e criticando. A crítica do jogador produz um *feedback* que pode ser de interesse de todos os outros papéis. A Figura 3 ilustra essas interações do Jogador com os artefatos e demais papéis do modelo.

Um dos princípios defendidos pelo modelo é que a interação "joga" deve envolver o Jogador ao máximo, de modo que a interação "aprende" seja executada de maneira o mais imperceptível possível pelo Jogador. Em outras palavras, o modelo parte do princípio de que o aprendizado não deve ficar limitado a metodologias puramente didáticas, sendo estimulada também a utilização do aprendizado inconsciente.

---

[3] XML é uma linguagem de marcação para documentos que contém informação estruturada. Uma linguagem de marcação é um mecanismo para identificar estruturas em um documento. A especificação XML define um padrão para adicionar marcações a documentos [O'Reilly® 2003]

[4] O W3C XML Schema é uma linguagem XML que descreve e restringe o conteúdo de documentos XML.

[5] O DTD (Document Type Definition) é uma alternativa anterior ao Schema para validação de documentos XML. Ele define os blocos válidos para a construção de um determinado documento XML.

Outro princípio defendido pelo modelo é o estímulo contínuo ao *feedback* do Jogador. Através desse *feedback*, Desenvolvedores podem melhorar a diversão oferecida pelo jogo, Educadores podem verificar se o tópico de aprendizado está sendo assimilado como desejado e o Regulamentador pode identificar se suas especificações estão provocando algum entrave no processo como um todo. O modelo proposto, entretanto, não define formalmente como a extração do *feedback* do jogador deve ser realizada, de modo a não burocratizar o processo nem restringi-lo a tecnologias ou ferramentas específicas (requisitos n° 4 e n° 7 do modelo).

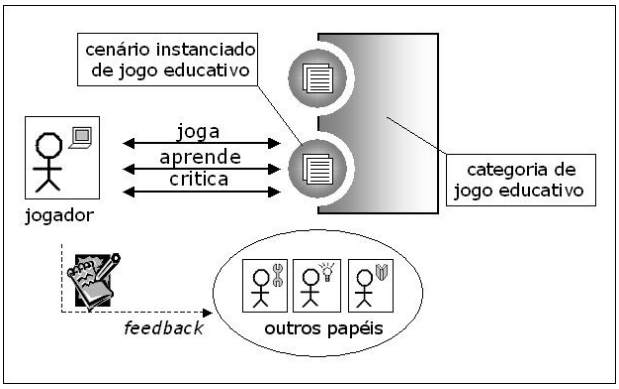

Figura 3. Interações do Jogador no modelo de time proposto

O modelo de time proposto, por fim, reconhece que o desenvolvimento de jogos educativos ainda não é um processo popularizado (requisito n° 7 do modelo). Visando suprir tal deficiência, o modelo acredita no potencial colaborativo da Internet para transformar o desenvolvimento de jogos educativos em um processo "*peer-to-peer*" [Alliance Consulting® 2003]. A arquitetura proposta pelo modelo, articulando de modo distribuído seus papéis e artefatos, viabiliza o princípio. Os autores deste trabalho acreditam na realização, através da World Wide Web, de um repositório central de categorias, ferramentas/scripts de automação, cenários e jogos educativos completos (cenários instanciados). Este repositório poderá adotar um modelo aberto, em que qualquer indivíduo será capaz de consultar ou adicionar artefatos à coleção já existente no repositório, desde que as devidas validações sejam realizadas por um Regulamentador. Adicionalmente, o repositório pode coletar *feedback* em geral para identificar possíveis deficiências no modelo ou nos artefatos disponibilizados.

O detalhamento do modelo, apresentado nesta seção, permite a construção de uma caracterização resumida do mesmo. Tal caracterização é apresentada na Tabela 2.

Tabela 2. Caracterização resumida para o modelo de time proposto

Nome	Modelo de Time para o Desenvolvimento Colaborativo de Jogos Educativos	
Contexto	Desenvolvimento colaborativo de jogos educativos visando combater limitações conhecidas dos mesmos	
Papéis	Jogador, Desenvolvedor, Educador e Regulamentador	
Áreas funcionais	Jogador	Reportagem de *feedback*
	Desenvolvedor	Projeto e Arquitetura de Implementação
		Desenvolvimento de Aplicações e de Infra-Estrutura
	Educador	Pedagogia relativa ao tópico de aprendizagem
	Regulamentador	Análise de ferramentas, scripts e tecnologias
		Gerenciamento de mudanças

		Gerência de configuração (para o repositório Web)
Responsabilidades	Jogador	Realizar críticas construtivas sobre os jogos educativos experimentados
	Desenvolvedor	Garantir a diversão e envolvimento do usuário
		Gerenciar aspectos técnicos e de implementação
		Respeitar as especificações para a categoria de jogo educativo escolhida
	Educador	Fornecer conteúdo educativo correto, nivelado e consistente.
		Respeitar as especificações para a categoria de jogo educativo escolhida
	Regulamentador	Validar especificações de categorias de jogos e scripts/ferramentas
		Viabilizar a abstração de tecnologia para o Educador
Alguns Princípios	Reuso através de customização; Abstração através de scripts e ferramentas; Aprendizado inconsciente; Estímulo contínuo ao *feedback* do Jogador; Popularização suportada por uma comunidade Web.	

4. Estudo de Caso

De modo a exemplificar o modelo, um jogo educacional foi desenvolvido de acordo com os conceitos apresentados. Apesar do jogo ser simples e não oferecer um grau de envolvimento mais do que razoável, ele atinge o propósito de ilustrar o modelo de time apresentado neste artigo.

O ciclo de vida do jogo ocorreu através das seguintes etapas:

- **Especificação da categoria de jogo**: um Regulamentador identificou e validou uma categoria de jogo educativo chamada "jogos de pergunta e resposta" (*quiz games*). A entrada parametrizada para este jogo, bastante simples, foi definida como um arquivo texto chamado "*entradaQuiz.txt*", contendo a seguinte informação (ordenadamente):
    - O número de *quizzes*, seguido do caractere de nova linha. Para cada *quiz*, deve haver:
    - Um texto descrevendo a pergunta da *quiz*, seguido do caractere de nova linha;
    - O número de respostas da *quiz*, seguido do caractere de nova linha;
    - O número da reposta correta da *quiz*, seguido do caractere de nova linha;
    - Textos descrevendo cada reposta da quiz, separados pelo caractere de nova linha.
- **Desenvolvimento do cenário**: um Desenvolvedor criou um cenário para a categoria "jogos de pergunta e resposta" em que a pergunta de cada quiz aparece no topo da tela. Cada resposta aparece em um balão, que viaja pela tela. O jogador deve atingir o balão com a resposta correta através de uma espingarda, controlada por ele no inferior da tela. Para o desenvolvimento do jogo, foi escolhida a linguagem funcional Haskell, por sua conhecida produtividade, e o motor para jogos FunGEn [Furtado e Santos 2002], útil para acelerar a etapa de implementação.
- **Instanciação do cenário:** Um Educador decidiu utilizar o cenário desenvolvido para ensinar a tabuada. Ele criou uma entrada parametrizada (arquivo *"entradaQuiz.txt"*) seguindo a especificação da categoria "jogos de pergunta e resposta". Como essa especificação é simples (arquivo texto), não foi necessário o uso de ferramentas ou scripts de automação pelo Educador. A entrada parametrizada criada, assim como o cenário instanciado a partir desta entrada, são exibidos na Figura 4.

- **Execução do jogo:** O jogo (cenário instanciado) foi apresentado a um Jogador, que interagiu com ele. O jogador foi estimulado a fornecer *feedback* aos outros papéis sobre o jogo.
- **Avaliação de feedback:** o Desenvolvedor concluiu, a partir do *feedback* do Jogador, que seu cenário deveria apresentar um maior grau de envolvimento. Os demais papéis (Educador e Regulamentador) ficaram satisfeitos com o *feedback* do Jogador.
- **Um novo ciclo é iniciado e assim por diante...**

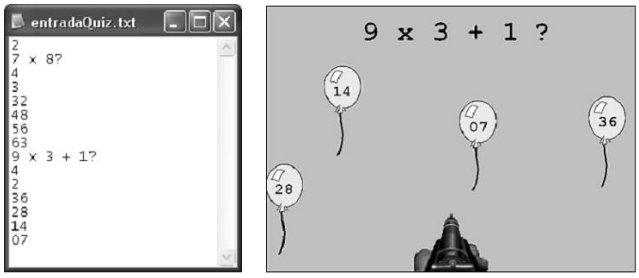

Figura 4. Entrada parametrizada e cenário instanciado do estudo de caso

5. Conclusão

Assim como ilustrado no estudo de caso, a utilização do modelo de time proposto promove impactos positivos no desenvolvimento de jogos educativos. Em primeiro lugar, o jogo desenvolvido se apresenta extensível o suficiente para que novos tópicos de aprendizagem sejam incorporados, uma vez que a customização do jogo, em tempo de execução, é trivial. Além disso, o Educador pode criar um jogo educativo totalmente diferente, mas que aproveite o conteúdo educativo (entrada parametrizada) já definido por ele: basta escolher outro cenário que pertença à mesma categoria "jogos de pergunta e resposta". Por fazer parte da mesma categoria, o novo cenário escolhido entenderá a entrada parametrizada antiga sem necessitar de qualquer modificação. É interessante observar que nenhum conhecimento de informática mais específico foi exigido do Educador, além da manipulação de arquivos texto. Adicionalmente, a utilização do modelo em conjunto com eficientes tecnologias de desenvolvimento, como é o caso da linguagem Haskell, revelou uma interessante combinação para promover reuso, concisão e modularização de código, facilitando a qualidade e a manutenção da aplicação desenvolvida. Os autores deste artigo consideram como um interessante trabalho futuro uma integração mais estreita do modelo proposto com o motor para jogos FunGEn.

O modelo, contudo, revela alguns desafios que ainda precisam ser resolvidos. Em primeiro lugar, a implantação do modelo vai exigir um esforço considerável para a identificação e especificação das várias categorias de jogos educativos atualmente existentes. Um exemplo de dificuldade em relação a isso são jogos educativos híbridos, isto é, que podem ser classificados em mais de uma categoria. Além disso, a integração do modelo com as atuais metodologias de desenvolvimento software precisa ser mais bem detalhada, de modo que sua utilização seja efetivamente produtiva. Por exemplo, o estágio atual em que se encontra o modelo não especifica os responsáveis por testar ou gerenciar o projeto.

Apesar dessas dificuldades, entretanto, os autores deste artigo acreditam que o uso do modelo contribuirá para que o processo de desenvolvimento de jogos educativos seja mais produtivo, popularizado e menos sujeito a erros, aumentando a satisfação dos educadores e dos usuários finais.

6. Referências

Alliance Consulting®, "What is Peer-to-Peer?" http://www.peertohere.com/aboutp2p/, maio 2003

Carraher, D.W. (1990) "O que esperamos do Software Educacional?" Acesso – Revista de Educação e Informática, Ano II, n° 3, jan./jun. 1990, issn 0103-0736;

Clua, E. W. G., Luca Jr., C. L., Nabais, R. J. M. "Importance and Impacts of Educational Games in Actual Society", WJogos'2002

Furtado, A. W. B., Santos, A. L. M. “FunGEn - A Game Engine for Haskell”, WJogos’2002

Mandel T., The Elements of user interface, John Wiley and Sons: New York, 1997;

Microsoft®, “The Microsoft Solutions Framework”, http://www.microsoft.com/msf, junho 2003.

O’Reilly®, “XML.com”, http://www.xml.com, maio 2003

Tchounikine P. “Pour une ingénierie des Environnements Informatiques pour l’apprentissage humain”, Information-Interaction-Intelligence, vol. 2, n. 1, 2002.