

D.O.M.: UM MODELO DE JOGO DIGITAL PARA A INDÚSTRIA DE GAMES EDUCACIONAIS

William Santos¹, Lynn Alves²

^{1,2}Faculdade SENAI CIMATEC

E-mails: william.santos@fieb.org.br, lynnalves@gmail.com

RESUMO

Este trabalho tem o objetivo de socializar a produção do *game* D.O.M. (Dispositivo Oral Móvel), considerando as Técnicas de Desenvolvimento Ágil, sobretudo o Scrum, acrescido da utilização da U.M.L. (Linguagem de Modelagem Unificada) como forma de propiciar uma maior integração entre equipes multirreferenciais, principalmente aquelas que atuam na indústria de desenvolvimento de *games* com fins educacionais. Para concretizar este objetivo foi feita uma análise da utilização da U.M.L. dentro do processo do Scrum, no desenvolvimento deste *game* que é voltado para o ensino das funções quadráticas.

1. INTRODUÇÃO

Com o aumento do mercado de entretenimento virtual, o desenvolvimento de jogos digitais vem ganhando cada vez mais espaço no Brasil. Segundo o relatório do BNDES de fevereiro de 2014, estima-se que o mercado brasileiro de jogos digitais movimentou em torno de US\$ 3 bilhões. Este mesmo relatório aponta que 74% das empresas produtoras de *games* no Brasil são de pequeno e médio porte. Esse cenário vem demonstrando a necessidade de um maior investimento na área de desenvolvimento de jogos digitais, como também vem exigindo cada vez mais de quem os produz, um conhecimento técnico e um domínio nos processos de engenharia de software de forma a otimizar e dar mais qualidade no processo de desenvolvimento.

O processo de desenvolvimento de jogos digitais é diferenciado se comparado aos dos softwares corporativos. Os softwares corporativos dependem de métodos cíclicos (iterativos), enquanto os jogos digitais são desenvolvidos através de métodos de desenvolvimento ágil (pequenas equipes) ou pelo Processo de Software Pessoal (PSP) (grandes equipes).

O desenvolvimento ágil é caracterizado como um *framework* conceitual, onde a produção é feita através de ciclos curtos de vida, minimizando os riscos em sua produção, enquanto o PSP é um processo baseado em projetos individuais que busca a melhoria da capacidade de organização do indivíduo favorecendo assim a melhoria da capacidade organizacional em seu contexto macro.

A metodologia de desenvolvimento ágil favorece a utilização de uma ferramenta bastante eficaz para o processo de desenvolvimento de um jogo digital, o Scrum. O Scrum é um processo de gerenciamento e controle de desenvolvimento de softwares e produtos complexos. Segundo [1], o Scrum possui um ciclo de vida de 30 dias, chamado *sprint*, onde em cada reunião no final deste período são determinadas novas atividades a serem desenvolvidas.

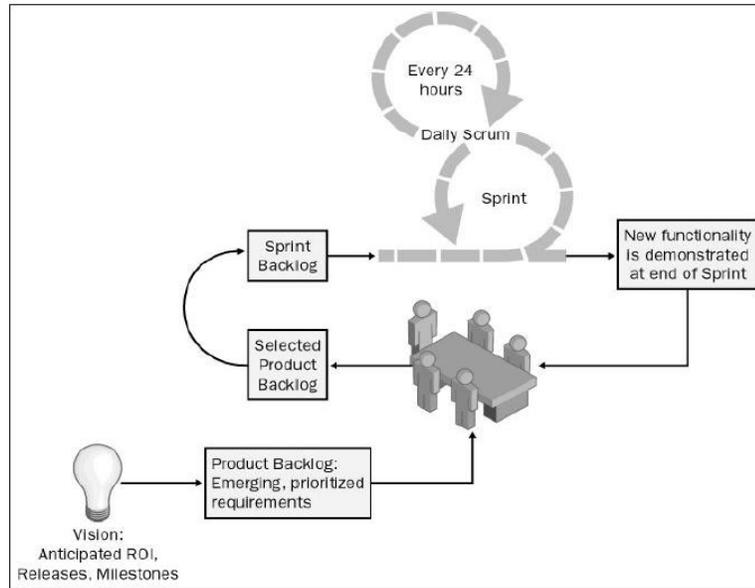


Figura 1 - Scrum - [1],

Dentro do processo de gerenciamento do Scrum há 3 momentos documentados, o *backlog* do produto, o gráfico *burndown* e o *sprint backlog*. O *backlog* do produto é o que precisa ser desenvolvido. O gráfico *burndown* representa a relação entre o que está sendo produzido e o tempo da produção, onde a meta é zerar o que precisa ser produzido até o final do período proposto. O *sprint backlog* é a definição das tarefas para próxima entrega.

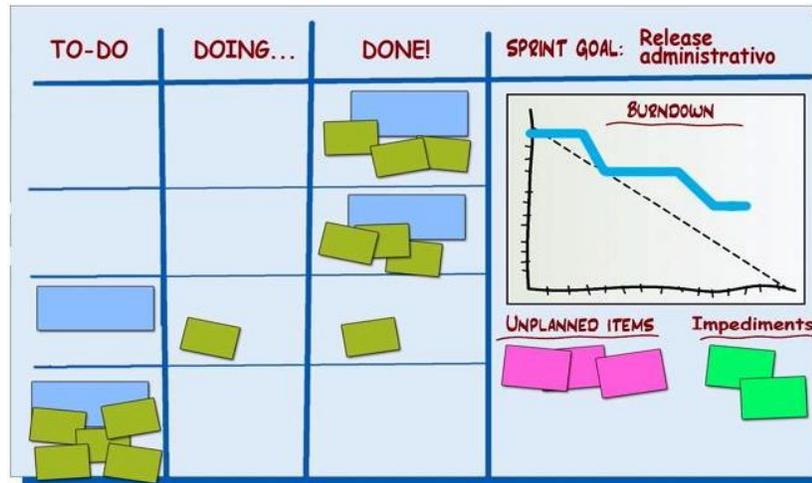


Figura 2 – Resumo do Scrum

Segundo [2], o Scrum é uma das maiores ferramentas de gerenciamento ágil, pois permite a comunicação diária a valorização do produto final e a metodologia ágil é ideal para o desenvolvimento de jogos digitais, pois provê uma abordagem incremental onde artistas, projetistas e programadores se focam tanto na diversão como no trabalho do desenvolvimento.

Dentro das empresas que desenvolvem jogos digitais, segundo o [3], cerca de 60% delas utilizam as técnicas de desenvolvimento ágil, como o Scrum, no intuito de otimizar o processo de produção, já que o desenvolvimento de jogos é uma atividade considerada complexa e dura

devido a quantidade de variáveis envolvidas nesse processo. [4], sinaliza alguns problemas no desenvolvimento de *games* com base nos *postmortems* (documento escrito no término de um projeto relatando os acontecimentos durante o processo de desenvolvimento de jogos) disponibilizados pelo site gamasutra¹, dentre eles os mais graves são:

- Falta de documentação – as fases do processo de criação não são documentados o que cria certa dificuldade caso haja uma mudança de integrantes na equipe.
- Problemas de comunicação – a comunicação entre os membros da equipe precisa ser boa a fim de evitar ambiguidades dentro do processo principalmente com a equipe técnica e equipe de arte.

Uma ferramenta que é comumente utilizada no processo de desenvolvimento de artefatos digitais e que pode solucionar esses problemas é a Unified Modelling Language (U.M.L.). Segundo [5], criadores da U.M.L., essa linguagem de modelagem proporciona uma forma-padrão na preparação de planos de arquitetura dos projetos. A U.M.L. surgiu da união de três metodologias de modelagem: o método de Booch, o método OMT (Object Modeling Technique) de Jacobson e o método OOSE (Object-Oriented Software Engineering) de Rumbaugh.

Apesar do processo de desenvolvimento de um jogo digital ser diferenciado de outros softwares convencionais pelo fato de envolver uma série de especificidades dentro das perspectivas de um *game design*, e no caso dos que são produzidos para fins educacionais, os aspectos pedagógicos, a U.M.L. além de fornecer uma documentação do processo de desenvolvimento pode possibilitar uma maior comunicação e integração a uma equipe multirreferencial².

A U.M.L. não é considerada uma linguagem de programação,

mas uma linguagem de modelagem, cujo objetivo é auxiliar os engenheiros de software a definir as características do software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. [6]

[7], sinaliza que realizar a análise da modelagem do sistema são os princípios objetivos da U.M.L.. Seus diagramas fornecem inúmeras visões de como deve ser modelado, o que facilita para o entendimento e cobertura total dos processos por meio de uma modelagem que contemple todas as áreas, de forma que os diagramas trabalhem em conjunto.

A utilização de diversos diagramas permite que falhas possam ser descobertas nos diagramas anteriores, diminuindo a possibilidade de ocorrência de erros durante a fase de desenvolvimento do software. [6]

Devido a falta de documentação como foi citado anteriormente, não há relatos da utilização da U.M.L. no desenvolvimento de *games*, tanto no processo de programação como no *game design* dos jogos digitais. Nesse intuito, o projeto aqui apresentado optou em usar esta linguagem de modelagem objetivando contribuir no processo de produção do *game* em todas as suas etapas e áreas, transpondo a ideia da utilização da U.M.L. apenas para subsidiar a etapa de programação, mas de utilizá-la também nas demais etapas do *game design*, potencializando assim o processo

¹ <http://www.gamasutra.com>. Acesso em 24 Jun 2014

² A equipe multirreferencial é aqui compreendida com a integração de distintos saberes que se relacionam, respeitando as diferentes referências que norteiam cada campo de conhecimento. Esta abordagem diferencia-se das perspectivas multi, inter e transdisciplinar que tem a ênfase na disciplina que delimita cada área.

do Scrum e servindo como referência para equipes de desenvolvimento que desejem incluir a U.M.L. em seu processo de criação.

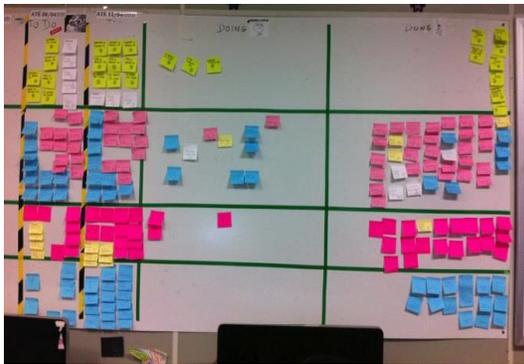
2. O DESENVOLVIMENTO DO GAME D.O.M.

Este projeto é uma ação do Grupo de Pesquisa Comunidades Virtuais que já tem no seu portfólio onze jogos desenvolvidos para distintos cenários de aprendizagem. O game D.O.M. possui gênero plataforma, em primeira pessoa (*single player*), em um ambiente gráfico 2D. O *game* conta com 4 fases (nave, deserto, floresta e caverna), *puzzles* ou *mini-games* entre as fases. É desenvolvido na linguagem de ActionScript e voltado para o ensino de funções quadráticas no 1º ano do Ensino Médio.

A equipe desenvolvimento é composta por 01 *game designer*, 01 pesquisador na área de modelagem computacional e matemática, 03 *designers* gráficos, 01 desenhista, 01 roteirista e 01 produtor de trilhas sonoras.

2.1. O Scrum no desenvolvimento do D.O.M.

A utilização do Scrum pôde possibilitar a equipe o cumprimento do cronograma de produção estabelecendo metas sobre as atividades que precisavam ser desenvolvidas por cada um da equipe como também nas etapas de retroalimentação do processo durante o *sprint backlog*. Aqui é apresentado um recorte de um dos *sprints backlogs* contendo as atividades direcionadas ao *Game Designer* para determinado período durante o processo de produção do *game*.



Sprint #1
(02/10 até 14/10)

(Game Design / Produção)

- => Workflow (telas) preliminar
- => Features in game (coletáveis e interagíveis)
- => Mecânicas features in *game* (coletáveis e interagíveis)
- => Doc Design prototipagem 2.0
- => Levels Flow 1.0
- => Estrutura Narrativa por fases
- => Doc. Mecânica Movimentos personagem

Figura 3 –Scrum de Desenvolvimento do D.O.M.

A cada nova meta traçada nos *sprints backlogs* a utilização da U.M.L. servia de base para construção dos aspectos do *game design* do jogo como será discutido na próxima seção.

2.2 A U.M.L. no Desenvolvimento do D.O.M.

O uso da U.M.L. facilitou a dinâmica de desenvolvimento do *game*, na medida em que possibilitou uma comunicação a partir dos seus modelos conceituais, permitindo a construção de modelos precisos, sem ambiguidade que serviram de base para *Game Designer*, *Level Designer* (Designer de níveis ou fases), *Character Designer* (Designer de Personagens), *Lead*

Programmer (Programador Condutor), *Lead Artist* (Artista/ Ilustrador Condutor) e Consultores Pedagógicos.

Além da criação da documentação necessária para as diversas demandas da equipe de trabalho, permitiu também uma maior integração do grupo, que é composto por profissionais de diversas áreas do conhecimento possibilitando uma comunicação de fácil compreensão apesar das diferentes expertises que compõem a equipe na produção de um *game* voltado para a aprendizagem de conteúdos educacionais.

No desenvolvimento do menu inicial, por exemplo, os diagramas da U.M.L. possibilitaram a identificação dos itens que precisariam ser contemplados neste menu. Um deles priorizava a construção do menu por parte do *Lead Artist* (Ilustrador), enquanto o outro por parte do programador. Através da utilização dos diferentes diagramas, suas transposições permitem identificar erros e ambiguidades, ao serem comparados, caso apresentem alguma inconsistência, permitindo que seja revisto anteriormente a sua produção, evitando futuros problemas na programação do game.

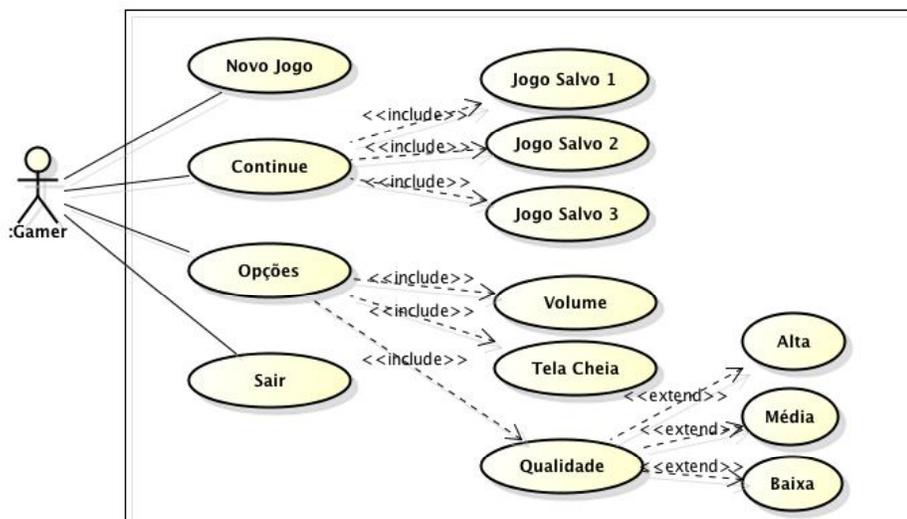


Figura 4 – Diagrama de Caso de Uso com Relações de Negócios

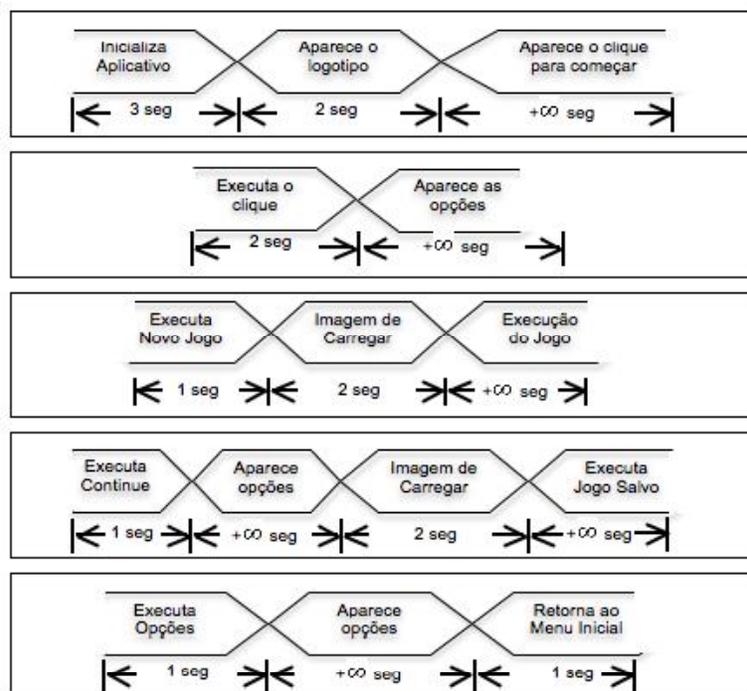


Figura 5 – Diagrama de Tempo (trocas de mensagem entre as telas)

De posse desta documentação e dos diagramas da U.M.L. foi construída as telas do menu inicial do game com o tempo das trocas de mensagem entre as telas, como pode ser visto na Figura 5.

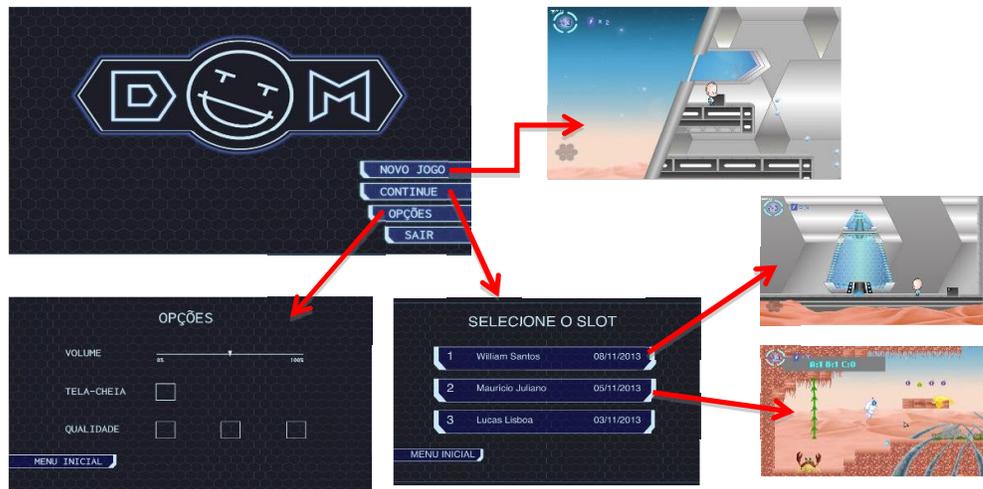


Figura 6 – Telas do Menu Inicial

Outro momento em que foi utilizado o diagrama de caso de uso foi para determinar as movimentações que o personagem teria dentro do jogo. Essa movimentação possibilitaria a construção da paleta de movimentos pelo *Lead Artist* (Ilustrador). Nesta fase, o personagem é desenhado várias vezes com pequenas mudanças em seus traços onde no momento da animação é dado a ideia do movimento do personagem.

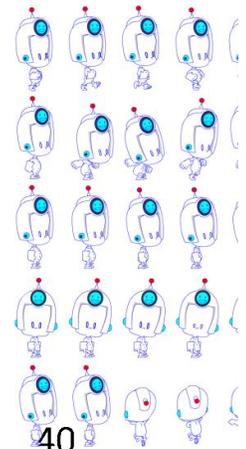
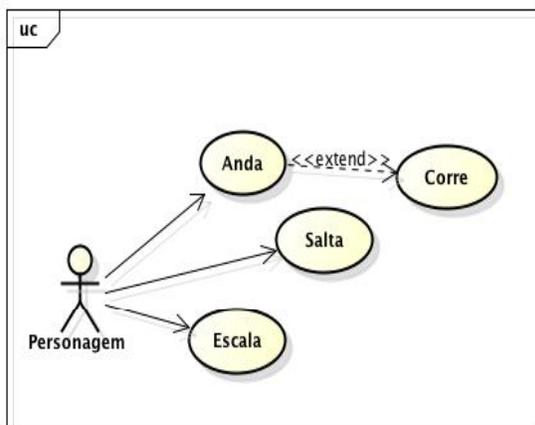


Figura 7 – Diagrama de Caso de Uso e Paleta de Movimentos do Personagem

Dentro da parte pedagógica, no que diz respeito aos conceitos e a forma que eles deveriam ser abordados no jogo durante os *puzzles*, um dos diagramas utilizados pelos consultores pedagógicos foi o diagrama de atividade. Ele permitiu a transposição da sequência e as possibilidades de resolução do puzzle que relacionava os coeficientes da função e a posição da parábola no plano cartesiano. O objetivo deste *puzzle* é movimentar a parábola apertando os botões que representam os coeficientes, encaixando a parábola no local indicada.

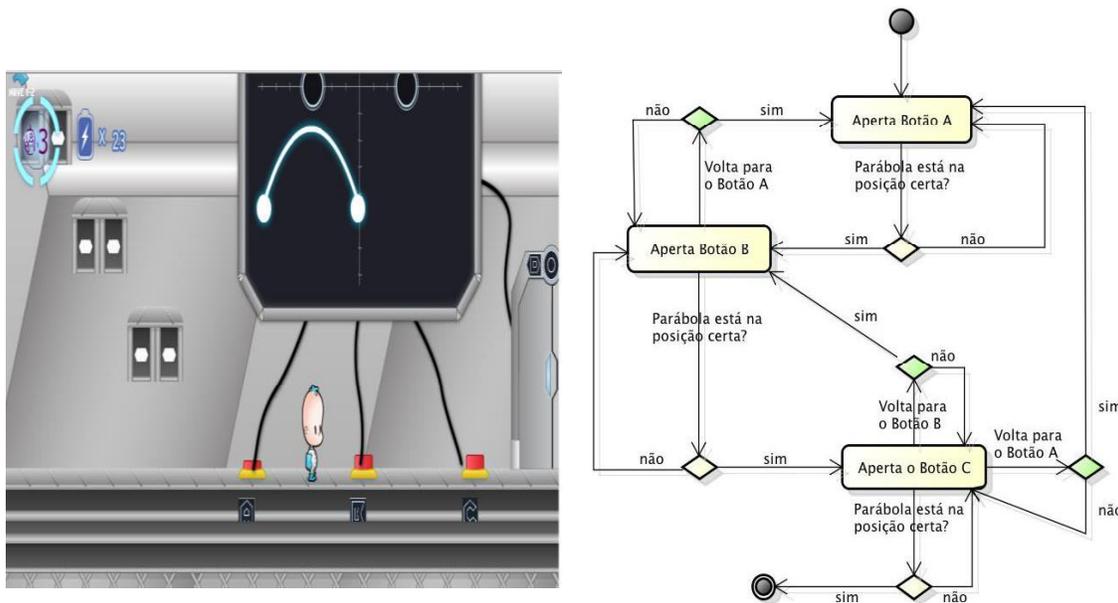


Figura 7 – *Puzzle* e Diagrama de Atividade

3. CONCLUSÃO

Na pesquisa aqui descrita, foi ratificado que a utilização do Scrum e da U.M.L. possibilitaram uma otimização do tempo de produção do *game* e na qualidade do produto final. Os *sprints backlogs* trouxeram um momento de avaliação daquilo que já havia sido construído, melhorando o controle de qualidade do produto a cada ciclo de produção, possibilitando, uma menor necessidade de revisões quanto a algum aspecto que já tivera sido desenvolvido. Quanto a U.M.L., a transposição visual de conteúdos, características e necessidades para o sistema, permitida através dos inúmeros diagramas da U.M.L. fizeram com que a comunicação entre a equipe se estabelecesse de uma forma mais colaborativa, além de possibilitar um esquadramento do que e de como deve ser feita a programação do *game*, otimizando este processo e evitando brechas que podem vir a acontecer quando não há um planejamento para esta etapa de produção. Os relatos da equipe envolvida na produção do *game* validam a utilização da U.M.L. principalmente por gerar documentos não só para os programadores como

também para os demais participantes da equipe, documentos estes que podem ser consultados a qualquer momento, diminuindo o processo de retrabalho ou de implementações devido a falta de algum aspecto não contemplado ou esquecido.

Como pôde ser visto, a utilização dos diagramas da U.M.L. e a comunicação através deles, independem da área de formação daqueles que atuam numa equipe de desenvolvimento de jogos digitais permitindo a coordenação e a condução do processo de forma tranquila. O modelo deste estudo mostra que é possível unir a U.M.L. ao processo do Scrum, como forma de reduzir os problemas sinalizados por [4], permitindo um amadurecimento no processo profissional da equipe, de otimizar o desenvolvimento do *game* e de possibilitar uma melhor comunicação e transposição de conhecimentos entre uma equipe multirreferencial em seus diversos saberes, contribuindo assim para futuras experimentações na indústria produtora de *games*.

REFERÊNCIAS

¹Highsmith, J. Agile Software Development Ecosystems. Addison -Wesley, Boston, MA, 2002.

²Keith, C. *Game Development with Scrum* – Addison-Wesley Professional; 1. ed. 2010.

³Censo *Gamer*. Disponível em <http://www.insidecomm.com.br/index.php/br/noticias/midia/82-infografico-censo-gamer-brasil-2012>. Acesso em 23 jul. 2014.

⁴ Petrillo, F.S. Práticas ágeis no processo de desenvolvimento de jogos eletrônicos. UFRGS. Instituto de Informática. Porto Alegre – RS, 2008.

⁵Booch, G., Rumbaugh, J., Jacobson, I. U.M.L.: guia do usuário. O mais avançado tutorial sobre Unified Modeling Language (U.M.L.), Editora Campus, 2000.

⁶Guedes, G. U.M.L. 2 – Guia Prático - 2ª Edição. São Paulo: Editora Novatec, 2007.

⁷Guedes, G. UML 2: Uma abordagem prática. São Paulo: Novatec Editora, 2009.