



A Aplicação da Linguagem de Modelagem Unificada (U.M.L.): Novas Perspectivas para o Desenvolvimento de Games Educacionais

William Santos

Programa de Pós-Graduação em Modelagem
Computacional
Faculdade de Tecnologia Senai Cimatec,
Salvador/BA

Lynn Alves

Programa de Pós-Graduação em Modelagem
Computacional
Faculdade de Tecnologia Senai Cimatec,
Salvador/BA

Resumo

Este trabalho tem o objetivo de socializar a produção do game D.O.M (Dispositivo Oral Móvel), considerando as Técnicas de Desenvolvimento Ágil – Scrum, modelado em U.M.L. (Linguagem de Modelagem Unificada) como forma de propiciar uma maior integração entre equipes multirrefenciais, principalmente aquelas que atuam no desenvolvimento de games educacionais. Para concretizar este objetivo é feito uma análise da articulação do Scrum com a U.M.L. no desenvolvimento do game D.O.M..

Palavras-chave: Jogos Digitais, Scrum, U.M.L.

Contato dos Autores:

william_tenor@yahoo.com.br
lynnalves@gmail.com

1. Introdução

O desenvolvimento de jogos digitais vem ganhando cada vez mais espaço no Brasil. Segundo a empresa GfK, em 2012, o mercado brasileiro de jogos digitais movimentou cerca de R\$ 1 bilhão. Segundo o relatório do BNDES de 2014, estima-se que o mercado já esteja perto de US\$3 bilhões.

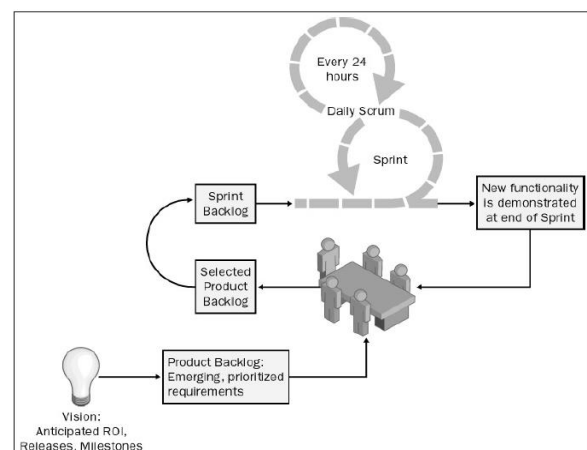
Esse cenário vem demonstrando a necessidade de um maior investimento na área de desenvolvimento de jogos digitais, como também vem exigindo cada vez mais de quem os produz, um conhecimento técnico e um domínio nos processos de engenharia de *software*.

O processo de criação de jogos digitais é diferenciado se comparado aos *softwares* corporativos. Os *softwares* corporativos dependem de métodos cíclicos (iterativos), enquanto os jogos digitais são desenvolvidos através de métodos ágeis (pequenas

equipes) ou pelo processo de *software* pessoal (P.S.P.) (grandes equipes).

O método ágil é caracterizado como um framework conceitual onde o desenvolvimento é feito através de ciclos curtos de vida, minimizando os riscos em sua produção, enquanto o (P.S.P.) 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 ágil favorece a utilização de uma ferramenta bastante eficaz para o processo de produção de um jogo digital: O Scrum. O Scrum é um processo de gerenciamento e controle de desenvolvimento de



softwares e produtos complexos. Segundo Highsmith [2002], o scrum possui um ciclo de vida de 30 dias, chamado *sprint*, onde a cada reunião no final deste período são determinadas as novas atividades a serem realizadas.

Figura 1: Scrum – Highsmith [2002]

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 em produção e o tempo gasto, onde o objetivo é zerar o que precisa ser produzido até o final do período proposto. O *sprint backlog* é a definição das tarefas para próxima entrega.

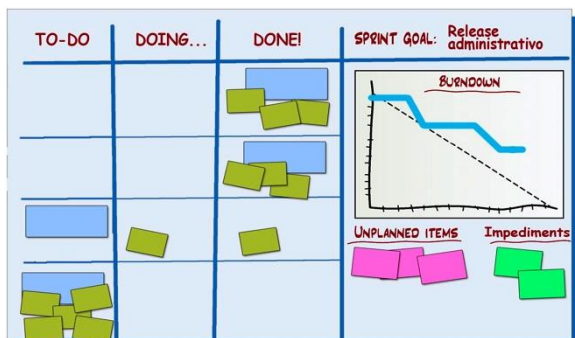


Figura 2: Scrum Resumido

Segundo Keith [2010], 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 a produção de jogos digitais, pois provê uma abordagem incremental onde artistas, projetistas e programadores se focam tanto na diversão como no trabalho do desenvolvimento. Segundo o Censo Gamer [2012], cerca de 60% das empresas produtoras dessas mídias, utilizam de Processos de Desenvolvimento Ágil – Scrum.

Dentro do processo de desenvolvimento de jogos digitais, principalmente os educacionais, o *game design* depende de vários profissionais de áreas diferentes, sua produção é multirreferencial¹ e depende de uma visão holística por parte de todos os membros da equipe, cabendo ao *gamer designer* uma liderança ímpar sobre todos da equipe.

Tabela 1 - Etapas do Game Design

(A) GERENCIAMENTO E DESIGN			
Game Designer, Level Designer (Designer de níveis ou fases), Character Designer (Designer de Personagens), Gerente de Projeto, Gerente de Software			
(B) PROGRAMAÇÃO			
Lead Programmer	Programador	Condutor,	Programadores
(C) ARTE VISUAL			
Lead Artist	Artista	Condutor,	Artistas visuais (modeladores, ilustradores etc.)

¹multirreferencialidade pode ser entendida como uma pluralidade de olhares dirigidos a uma realidade e uma pluralidade de linguagens para traduzir esta mesma realidade e os olhares dirigidos a ela. [Fagundes e Burnham, 2001, P. 48]

(D) MÚSICA
Músico, Efeitos sonoros e diálogos, Programador de áudio
(E) CONTROLE DE QUALIDADE
Q. A. Lead (Condutor de controle de qualidade), Q. A. Technicians (Controladores de qualidade), Playtesters (Jogadores avaliadores)
(F) OUTROS
Especialistas em outras áreas (educadores, consultores etc.), Técnicos em áreas diretamente relacionadas (captura de movimentos, roteirista, etc.)

Fonte: Tavares (2009)

Apesar da grande maioria dos desenvolvedores de jogos digitais utilizar o Scrum, Petrillo [2008] sinaliza alguns problemas no desenvolvimento de games através do Scrum, principalmente no que diz respeito a complexidade dos projetos e a quantidade de pessoas envolvidas neste processo. Os maiores problemas são:

- 1) Literatura especializada com lacunas teóricas – a literatura sobre games não retrata especificamente a complexidade dos projetos.
- 2) Escopo deficiente – os escopos não são pré-estabelecidos de forma correta, causando problemas quanto a criação e retirada de features (seqüência de fases e funcionalidades).
- 3) Cronograma – são esquecidos possíveis problemas que podem ocorrer e outros tipos de situação que podem acumular atrasos no projeto.
- 4) Crunch time (horas extras) - sobrecarga de trabalho no processo de validação.
- 5) Tecnologia utilizada – os jogos possuem avançada utilização na parte gráfica.
- 6) 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.
- 7) Problemas de comunicação – a comunicação entre os membros da equipe apresenta ambiguidades dentro do processo principalmente com a equipe técnica e equipe de arte.

Como forma de evitar e/ou amenizar estes problemas, o uso da Linguagem de Modelagem Unificada (U.M.L.) pode auxiliar o processo do Scrum, diminuindo os problemas de comunicação entre a equipe de desenvolvimento, facilitando o processo de produção e até mesmo fornecendo uma literatura que dê subsídios para demais equipes no desenvolvimento de jogos digitais.

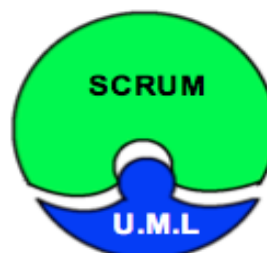


Figura 3: Módulo U.M.L no Scrum
 Fonte: Próprios Autores

Portanto, incorporar alguns “módulos” da U.M.L. no processo do scrum pode complementar esse método, sem quebrar a dinâmica do método ágil, buscando uma melhor eficiência no processo de desenvolvimento de jogos digitais.

2. Por que a U.M.L.?

Segundo Booch, Rumbaugh e Jacobson, [2000] 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 diferente se comparado aos *softwares* corporativos, pelo fato de envolver uma série de especificidades dentro das perspectivas de um game design, principalmente no que diz respeito aos 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 entre uma equipe multirreferencial.

a U.M.L. não é 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. [Guedes, 2007]

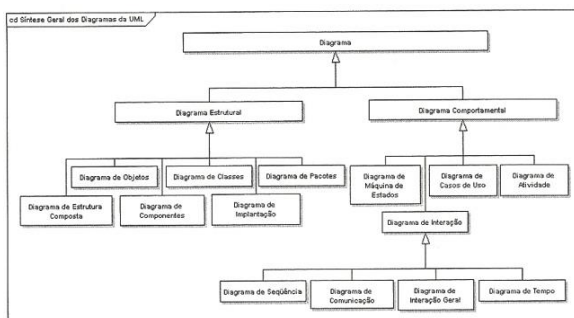


Figura 4: Tipos de Diagramas da U.M.L – Guedes [2009]

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. [Guedes, 2007]

Guedes [2009], sinaliza que realizar a análise da modelagem do sistema são os princípios objetivos da U.M.L.. Seus diagramas fornecem inúmeras visões de como modelar, 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.

3. O Desenvolvimento do game D.O.M.



Figura 5: Game D.O.M.

Este projeto foi financiado pela Secretaria de Educação do Estado da Bahia, desenvolvido pelo Grupo de Pesquisa Comunidades Virtuais que tem no seu portfólio onze jogos produzidos 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. Foi desenvolvido na linguagem de ActionsScript e voltado para o ensino de funções quadráticas no 1º ano do Ensino Médio.

A equipe de desenvolvimento foi 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.

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

A utilização do Scrum no desenvolvimento do D.O.M. possibilitou a equipe uma maior flexibilidade no processo de construção de cada etapa do jogo. A possibilidade de verificação contínua do que é produzido diariamente permitiu uma retroalimentação do processo

Aqui é apresentado um recorte de um dos Sprints Backlogs contendo as atividades direcionadas ao Game



XI Seminário

SJEEC

Jogos Eletrônicos - Educação - Comunicação

1 e 2
de Junho
2015

Designer para determinado período no processo de produção do *game* e o quadro Kanban que orienta o objetos que estão em produção e que vão ser produzidos durante aquele período.

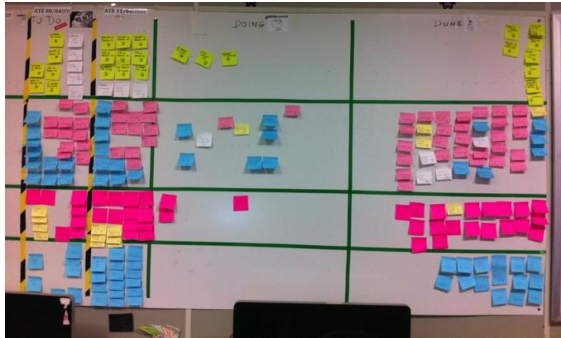


Figura 6: Kanban - Scrum do D.O.M

<p style="text-align: center;">D.O.M. Sprint #1 (02/10 até 14/10) (Game Design / Produção)</p> <ul style="list-style-type: none">=> 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 (componentes emocionais)=> Doc. Mecânica Movimentos personagem=> Doc. Mecânica Jump e demais ações in game=> Doc Individuais Vínculos=> Doc. Conceitos do projeto=> Análises de risco do projeto=> Cronograma de desenvolvimento (Scrum/Metas/Datas - Deadlines)=> Cronograma Playtests e Builds=> Doc. Design Core Mecânica In Game=> Unidades Monetárias do Game=> Level Design preliminar F1 (Nave)=> Level Design 1.0 F1 (Nave)=> Level Design preliminar F2 (Deserto)=> Level Design 1.0 F2 (Deserto)=> Level Design preliminar F2 - 2 (Deserto)=> Level Design 1.0 F2 - 2 (Deserto)
--

Figura 7: Recorte do Sprint do D.O.M

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

A utilização da U.M.L. ocorreu durante as reuniões dos sprints no intuito de facilitar a dinâmica de desenvolvimento do *game* e reduzir as lacunas existentes no scrum. Como cita Booch [2010], a U.M.L. possibilita uma comunicação a partir dos seus

modelos conceituais, permitindo a construção de modelos precisos, sem ambiguidade.

Pensando nisso, os diagramas e outros documentos fornecidos pela U.M.L. puderam servir de referência e de material de consulta na construção de cada objeto do jogo pelos diversos setores de criação. Esses diagramas, por exemplo, permitiram a criação de uma documentação básica para a produção do jogo.

Além da criação de uma documentação necessária antecipada para as diversas demandas da equipe de trabalho, a U.M.L. permitiu também uma maior integração do grupo, que era outra lacuna presente no scrum, considerando que a equipe de produção de um jogo digital que é composta por profissionais de diversas áreas do conhecimento e a U.M.L. possibilitaria 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 destes diagramas priorizava a construção do menu por parte do art designer, enquanto o outro por parte do programador. As duas visualizações dos diferentes diagramas permitiu identificar erros e ambiguidades, ao serem comparados, caso apresentem alguma inconsistência, permitindo que seja revisto anteriormente à sua produção, evitando futuros problemas na programação do menu.

NARRATIVA CASO DE USO 02			
Sigla	NCU02	Nome	Acessar jogos salvos na memória
Objetivo	Descrever o acesso do <i>gamer</i> aos jogos salvos.		
Ator	<i>Gamer</i>		
CENÁRIO PRINCIPAL			
1. O caso de uso se inicia ao <i>gamer</i> inicializar o aplicativo do jogo.			
2. O sistema apresenta na tela de menu um botão CONTINUE, que ao ser clicado direciona o <i>gamer</i> a outra tela.			
CENÁRIOS ALTERNATIVOS			
2.1 Ao clicar na opção continue aparecerá uma lista contendo os 3 (três) últimos jogos salvos e a opção menu inicial			
2.2 Ao clicar em uma das 3 opções de jogo salvo, o <i>gamer</i> será direcionado para aquela etapa			
CENÁRIOS DE EXCEÇÃO			
Desistência de acessar um jogo salvo			
a) Caso o <i>gamer</i> desista de iniciar um jogo salvo ele poderá clicar na opção menu inicial e voltar para tela anterior.			

Figura 8: Narrativa de Caso de Uso

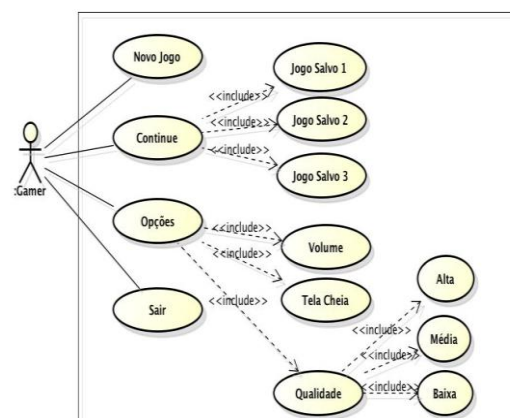


Figura 11: Diagrama de Atividade

Seguindo o referencial construído através dos diagramas da U.M.L, a tela do menu inicial foi produzida pelos designers e programadores como pode ser visto na Figura 12.

Figura 9: Diag. de Caso de Uso com Relações de Negócios

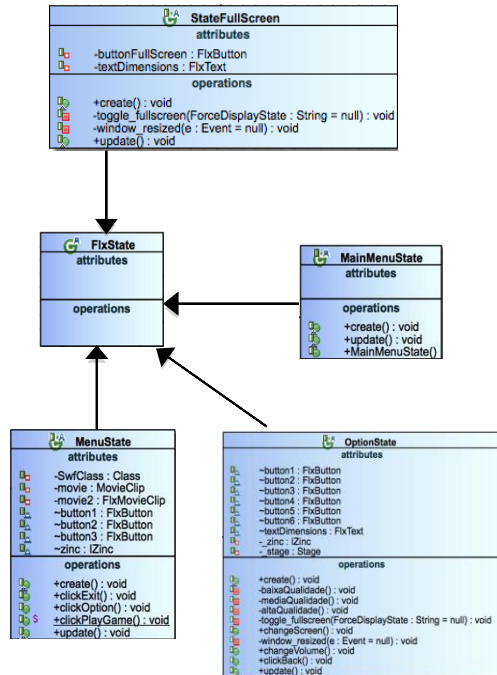


Figura 10: Recorte do Diagrama de Classes Menu Inicial

Outro diagrama utilizado para a criação do menu foi o diagrama de atividades, que possibilitou o direcionamento na formatação da tela de opções que deveria conter a possibilidade de escolha, identificada no diagrama pelo símbolos de tomada de decisão. A utilização de tais diagramas reduziram o tempo de produção e programação de cada tela, evitando tentativas/erros e processos de re-trabalho até a conclusão do produto final do menu. A utilização destes diagramas durante o processo de produção trouxe como contribuição a diminuição do Crunch time do produto.

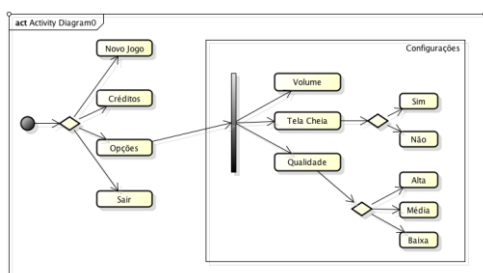


Figura 12: Menu Inicial

A grande variedade de diagramas da U.M.L. também puderam ser utilizados em diversos momentos da criação do jogo. Por exemplo, para determinar o tempo de mudança de cada tela, a cada clique nas opções possíveis, foi utilizado o diagrama de tempo, conforme é apresentado na Figura 13.

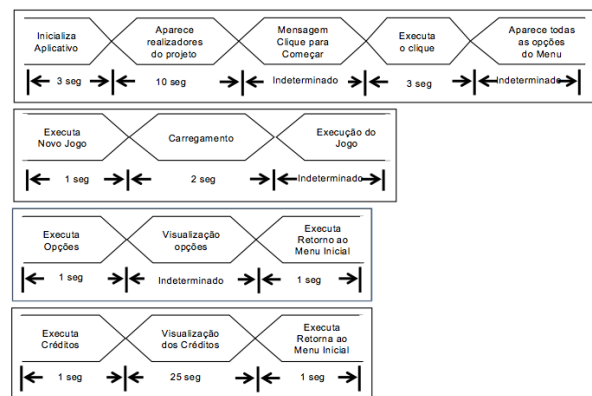


Figura 13: Diagrama de tempo

Outra possibilidade de utilização foram os diagramas de caso de uso, que orientaram a construção da paleta de movimento dos diversos personagens do jogo.

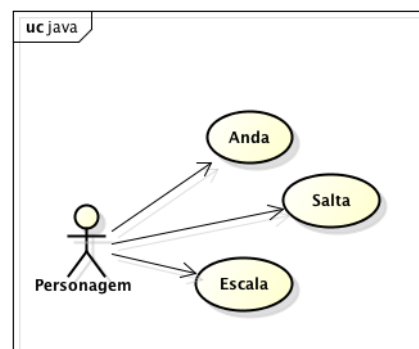


Figura 14: Diagrama de Caso de Uso - Personagem

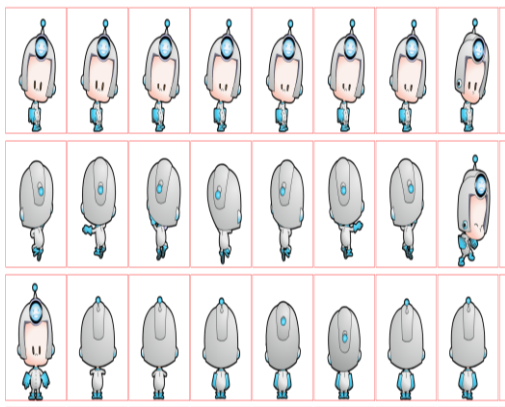


Figura 15: Recorte de Alguns Movimentos do Personagem

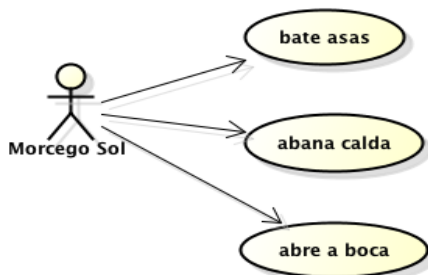


Figura 16: Diagrama de Caso de Uso – Morcego Sol

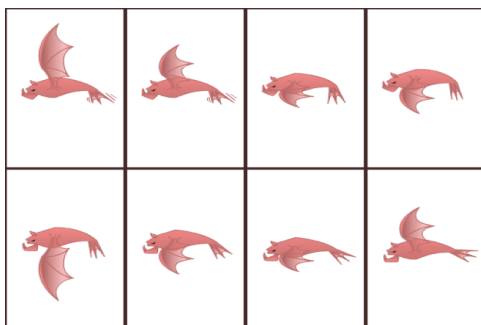


Figura 17: Recorte de Alguns Movimentos do Morcego

Por se tratar de um jogo educacional voltado para o ensino das funções quadráticas, a equipe do D.O.M. necessitava da presença de um consultor da área de matemática para direcionar como deveriam ser trabalhados os conceitos relacionados com a função quadrática no jogo.

Apesar de ser um profissional que habitualmente não está inserido no contexto da produção de um jogo digital, é imprescindível a sua participação nesse processo de desenvolvimento de um jogo com fins educacionais. Para facilitar a inserção deste profissional no grupo de trabalho, a utilização dos diagramas da U.M.L. permitiu a melhor orientação e comunicação na criação dos puzzles que direcionariam o processo de aprendizagem. Um exemplo disso, foi a construção do diagrama de atividade do puzzle que relacionava os coeficientes A, B e C, da equação de 2º grau.

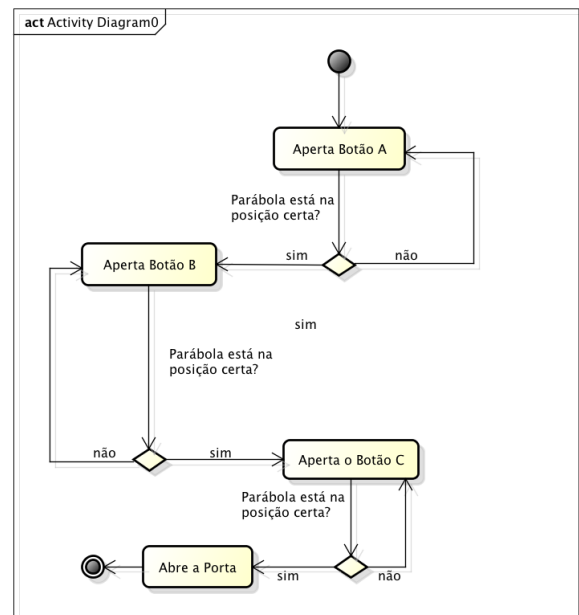


Figura 17: Diagrama de Atividades – Puzzle 1

Neste puzzle o gamer deveria clicar nos botões que representavam esses coeficientes e identificar as consequências na posição da parábola na tela do jogo.

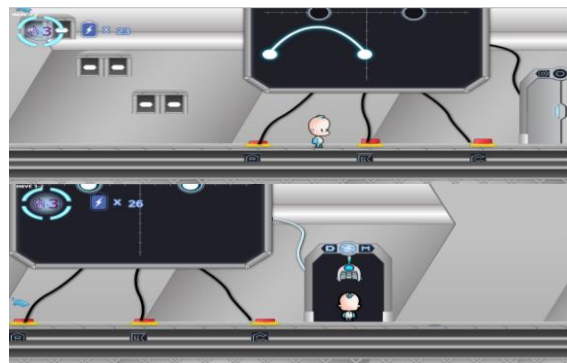


Figura 18: Puzzle 1



Para a construção de outro puzzle do game que envolvia um tipo de simulador para funções quadráticas através da mudança dos coeficientes da função, foi utilizado um diagrama de classe. Este diagrama direcionou a criação do script da programação para construção do gráfico da função quadrática e do mecanismo de interação/simulação durante o jogo.

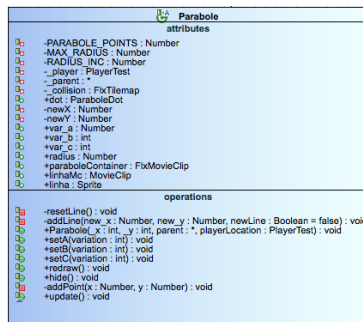


Figura 19: Diagrama de classe



Figura 20: Parábola - Puzzle 2



Figura 21: Parábola - Puzzle 3

4. Conclusão

O modelo proposto nesse estudo mostra a articulação da U.M.L. com o processo de desenvolvimento ágil - Scrum, apesar de serem considerados métodos

antagônicos, considerando suas diretrizes metodológicas.

Diversos outros métodos já foram propostos por pesquisadores da área de produção de jogos, como o Game Waterfall Process, o Extreme Game Development e o Game Unified Process, mas nenhum deles representa um método absoluto. A inclusão da U.M.L. no processo de desenvolvimento ágil não descredibiliza o Scrum, mas é uma tentativa válida na busca de uma maior eficiência neste processo de desenvolvimento, que vem apresentando diversas falhas.

A utilização dos diagramas comportamentais (estado, caso de uso e atividade), de interação (tempo e sequência) e estruturais (classes), puderam ser facilmente inseridos no processo durante os *sprints backlogs* sem tirar a dinâmica do processo ágil. A transposição visual permitida por esses diagramas fez com que a comunicação entre a equipe se estabelecesse de uma forma mais colaborativa, além de possibilitar um esquadramento do que e de como deveria 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.

A utilização da U.M.L. dentro do processo do Scrum no desenvolvimento do jogo D.O.M. trouxe uma grande contribuição para a qualidade do produto final, a partir do direcionamento das etapas de criação de cada produto, permitindo um amadurecimento no processo profissional da equipe e da transposição de conhecimentos entre uma equipe multirreferencial em seus diversos saberes, otimizando o desenvolvimento do jogo.

Que este estudo possa contribuir como referencial para demais equipes de desenvolvimento de jogos digitais no que diz respeito a ser uma literatura que traz um método ou a possibilidade de articulação de outros métodos existentes, para uma maior eficiência no processo de criação de um jogo digital.

Referências

BOOCH, G. U.M.L. - *Guia Do Usuário*. 2ª Edição. São Paulo, 2010.

BOOCH, G., RUMBAUGH, J., JACOBSON, I. *U.M.L.: guia do usuário. O mais avançado tutorial sobre Unified Modeling Language (U.M.L.)*, elaborado pelos próprios criadores da linguagem. Editora Campus, 2000.

CENSO GAMER. Disponível em <http://www.insidecomm.com.br/index.php/br/noticias/mid>



XI Seminário

SJEEDC

Jogos Eletrônicos - Educação - Comunicação



ia/82-infografico-censo-gamer-brasil-2012. Acesso em 23 jul. 2014.

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

GUEDES, G. *UML 2: uma abordagem prática* / São Paulo: Novatec Editora, 2009.

HIGHSMITH, J. *Agile Software Development Ecosystems*. Addison -Wesley, Boston, MA, 2002.

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

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

TAVARES, R. *Fundamentos de Game Design para Educadores e não Especialistas*, In Lucia Santaella e Mirna Feitoza, Org. Mapa do Jogo - São Paulo: Cengage Learning, 2009.