



Health Simulator: A Produção do Front End

Eduardo Gomes¹, Susana Senna², Rubens Muller³, Alessandro Lima⁴,
Paulo Ricardo Barros⁵, Marta Bez⁶

Resumo

O uso de softwares do tipo simuladores tem aumentado significativamente, dentre estes, o uso de simuladores virtuais tem se destacado muito, visto que têm como principal intuito apoiar a aprendizagem. Em fase de aprendizado, o aluno pode interagir com um programa que representa um paciente, seu diagnóstico e possíveis condutas. Alunos da graduação, pós-graduação e professores têm trabalhado no projeto Health Simulator que propõe um jogo do tipo simulador virtual tridimensional. A metodologia utilizada no desenvolvimento do simulador baseia-se no sistema em três etapas de Schuytema: Pré-produção, Produção e Pós-produção. Ancorada a esta, utiliza-se práticas ágeis adaptados ao desenvolvimento de jogos digitais. O jogo foi dividido em duas partes, o *Front End* (ambiente do aluno) e o *Back End* (ambiente do professor). Nesse trabalho, será apresentado o *Front End*, seus resultados parciais de componentes de arte.

Palavras-chave: Saúde. Métodos ágeis. Jogos sérios.

Abstract

The use of the type simulator software has increased significantly, among them the use of virtual simulators has excelled too, as have the main aim to support learning. In the learning phase, the student can interact with a program that is a patient, diagnosis and possible behaviors. undergraduate students, graduate and teachers have worked in Health Simulator project that proposes a 3D game virtual simulator. The methodology used in the development of the simulator is based on the system in three steps about Schuytema: Pre -production, Production and Post Production.

¹ Eduardo Gomes - Graduando em Tecnólogo de Jogos Digitais na Universidade Feevale, 2016.

² Susana Senna - Graduada em Design na Universidade Feevale, 2016.

³ Rubens Muller - Graduando em Tecnólogo de Jogos Digitais na Universidade Feevale, 2016.

⁴ Alessandro Lima – Mestre em Design Virtual pela UFRGS, 2015. Pós-Graduado em MBA Comunicação Estratégica e Branding pela Universidade Feevale, 2011. Graduado em Design Gráfico na Uniritter, 2010.

⁵ Paulo Ricardo Barros - Mestre em Ciências da Saúde: Educação e Informática em Saúde pela Universidade Federal de Ciências da Saúde de Porto Alegre, Bacharel em Sistemas de Informação pela Universidade Feevale (2009).

⁶ Marta Bez - Doutora em Informática na Educação pela Universidade Federal do Rio Grande do Sul, 2013. Mestre em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul, 2001. Graduada em Tecnólogo Em Processamento de Dados pela Universidade do Vale do Rio dos Sinos, 1991.



Anchored to this, it uses agile practices suited to the development of digital games. The game was divided into two parts, the *Front End* (student environment) and the *Back End* (teacher's room). In this work, the *Front End*, the partial results of art components will be displayed.

Keywords: Health. Agile method. Serious games.

INTRODUÇÃO

Segundo as Diretrizes Curriculares Nacionais do Curso de Medicina (BEZ, 2013), o currículo e o método pedagógico desejáveis devem permitir o desenvolvimento da observação e habilidade de escuta, tornando o aluno apto a pensar e, conseqüentemente, a aprender, ser, fazer e conviver com a auto aprendizagem. Tsuji e Silva (2010) deixam claro que os estudantes precisam praticar desde o início do curso, através de atividades que apresentem aumento gradativo de complexidade no decorrer no curso. A pedagogia evidencia esta prática utilizando os Métodos Ativos de Aprendizagem. Tal método apresenta como desafio a forma em que os problemas são apresentados aos alunos, permitindo a estes um estudo e aprofundamento do conhecimento independente de horário ou local em que se encontram.

Bez (2013) identifica que a simulação de casos clínicos (reais ou fictícios) pode ser uma boa estratégia de apresentação de conteúdo, pois os professores podem criar situações clínicas que contemplam os mais diversos temas, disponibilizados através da internet. Para tanto, é possível desenvolver aplicativos de simulação que criam ambiente e contexto críveis para o aluno exercitar, utilizando como interface de comunicação entre aluno e máquina, um Paciente Virtual (PV). De acordo com Orton e Mulhausen (2008, p. 75), um PV é “um programa interativo que simula a vida real em cenários clínicos, que permite o aprendizado de atos do profissional da saúde, obtendo a história clínica, exames e realizando diagnóstico e decisões terapêuticas”. Ou seja, é uma metodologia que utiliza um aplicativo de jogo interativo para que o aluno possa aprender enquanto se faz, conhecida como Aprendizagem Baseada em Problemas (ABP).



A ABP é uma metodologia de ensino onde o aluno deve ser o agente na produção do seu conhecimento. Ela pode ocorrer tanto em aulas formais, onde professores organizam assuntos e conteúdos que serão desenvolvidos em sala de aula, normalmente com trabalhos em grupo, quanto fora. Desta forma, rompe-se a maneira tradicional e rígida já desenvolvida dos conteúdos, sendo os mesmos incorporados durante o desenvolvimento do projeto.

O filósofo americano John Dewey (1859–1952) foi um dos precursores da Aprendizagem Baseada em Problemas, que procurava demonstrar o “aprender mediante o fazer” (MASSON, MIRANDA, MUNHOZ JR e CASTANHEIRA, 2012). Esta metodologia, conforme Bie (2008), é considerada construtivista, pois permite aos alunos adquirirem conhecimentos, habilidades e valores através de um processo gradativo de investigação, “estruturado em torno de questões complexas e autênticas, aplicadas na vida real, com produtos e tarefas cuidadosamente planejados”. Além disso, o escopo desse método respalda na diversidade informacional, ao invés da unicidade de fonte de conhecimento que se resvale unicamente na figura do professor (PAPINCZACK, 2009).

Todavia, para a produção de simuladores digitais do tipo em três dimensões vários problemas de produção podem surgir, a começar pela demanda por profissionais qualificados para desenvolver um volume grande de modelos, texturas e animações para alimentar tal simulador. Cabe aqui o entendimento sobre como a adoção de filosofia e princípios ágeis podem facilitar o processo de produção de modelos em três dimensões, aplicáveis em sua construção de malha e configuração de exibição de revestimento.

Para se compreender de que forma a filosofia e princípios ágeis podem contribuir positivamente em processos de produção de modelos tridimensionais para jogos digitais, se faz necessário compreender sua produção hoje. Teles (2006, p. 31) aponta que os métodos ágeis são definidos como iterativos, adotando em cada fase evolutiva do projeto um momento para revisões e ajustes antes de prosseguir as demais fases. Ou seja, nos processos ágeis o produto é



desenvolvido aos poucos, analisado, testado e a cada nova iteração de seu desenvolvimento, mais elementos são agregados a este, resultando em uma maior entrega de valor ao cliente. Acredita-se que a utilização de filosofia e princípios ágeis pode ajudar no desenvolvimento de modelos para jogos digitais, pois permite avaliar a produção de modelos de modo geral, compreendendo os requisitos do projeto, não tratando a produção destes modelos como atividades isoladas e independentes do projeto.

Para o desenvolvimento do aplicativo de jogo *Health Simulator*, houve a necessidade em dividir a equipe em duas sub-equipes, denominadas de *Front End* e *Back End*. A primeira se ocupa da produção do aplicativo de jogo, e a segunda ocupa-se da produção dos elementos de servidor. Para este artigo, será aprofundada a área de *Front End*, com ênfase na produção de elementos de arte e design.

Este artigo descreve parte do desenvolvimento do simulador de casos clínicos com foco na produção de modelos de cenários e personagens em três dimensões, suas animações e interfaces de conversação com jogador. A base teórica para o gerenciamento de projetos de jogos e desenvolvimento de modelos tridimensionais com a utilização dos métodos ágeis aplicados é exposta na Fundamentação Teórica. A metodologia utilizada para o desenvolvimento dos modelos é apresentada em Metodologia. A execução prática é descrita dentro de Experimento, seguido de Resultados Parciais. O fechamento do artigo é feito com as Considerações Finais.

FUNDAMENTAÇÃO TEÓRICA

Neste ponto são apresentadas as bases conceituais utilizadas no projeto. Aqui serão apresentadas as pesquisas teóricas que permitiram a execução deste projeto.



Jogos Digitais

A área de jogos desde muito cedo é explorada como recurso de diversão e interatividade que integra sociedades, pelo fato de que é uma atividade que permite a participação de mais de uma pessoa. Desde os primórdios da humanidade, o jogo tem sido mais utilizado como elemento de entretenimento do que algo mais sério, isto pelo caráter lúdico e cômico que a atividade de jogo proporciona (HUIZINGA, 2007, p. 9). Os Jogos Digitais são uma evolução dos jogos tradicionais, sendo um meio de entretenimento digital lúdico composto por regras, no qual são necessários três grandes pilares para o desenvolvimento de um jogo: programação, arte e gestão.

Os programadores são os responsáveis pelo funcionamento do jogo. Segundo Reis,

Algumas características específicas exigidas de um programador de games são: capacidade de se adaptar facilmente a novas tecnologias, trabalhar em projetos não detalhados, otimizar rotinas e, principalmente, lidar bem com outros profissionais envolvidos no projeto, como artistas, designers e escritores (REIS, 2002, p. 11).

São os programadores que vão dar vida aos jogos e fazer com que o sistema de regras, narrativa e animações sejam executados dentro do jogo através do conhecimento em linguagens de programação. Já os artistas são os responsáveis pela parte visual do jogo, pois eles fazem o design de personagens de cenários, as animações e a modelagem de cada elemento gráfico do jogo. Dentro do projeto *Health Simulator*, a modelagem é feita utilizando o software Autodesk 3D Studio Max e as texturas são feitas com o Adobe Photoshop. É de responsabilidade dos artistas a importação dos elementos de arte para dentro da *Engine* de jogo. A *Engine* adotada para o presente projeto é a Unity3D, haja visto sua facilidade de uso com sua codificação em linguagem C#.

Os Game Designers, por sua vez, organizam tudo para que o projeto aconteça, são eles que fazem a documentação do que vai haver dentro do jogo,



as regras, os objetivos, a quantidade de cenários e personagens, bem como o modo que a jogabilidade ocorrerá. Segundo Novak (2011, p. 186) a jogabilidade “pode ser definida como as escolhas, os desafios ou as consequências enfrentadas pelos jogadores ao navegar em um ambiente virtual”. Os Game Designers controlam a produção de arte e design, juntamente com a programação no sentido de que tudo se conecte adequadamente ao fim do projeto, observando os prazos estabelecidos. Segundo Reis (2002, p. 4), “este profissional está presente na grande maioria das equipes de desenvolvimento de jogos digitais, e mesmo quando não há um game designer específico, sua função é executada de alguma forma pela equipe”.

Modelos Tridimensionais para Jogos

Para se criar modelos críveis para ambientes de jogos se deve considerar suas principais características. Segundo Fox (2004, p. 119), “criar um objeto em 3D complexo, como a cabeça ou o corpo do personagem requer primeiro criar a topologia ou trabalho de grade de sua malha e então modelar esta topologia na forma que deseja artisticamente” (FOX, 2004, p. 119). A partir do entendimento sobre como é o modelo, pode-se então partir para sua construção tridimensional.

Dentro desta construção, várias técnicas são possíveis de utilização para desenvolver modelos tridimensionais e digitais. Os métodos adotados no projeto *Health Simulator* para construção de modelos baseiam-se em técnicas de modelagens definidas por *Face-by-Face* (LIMA, 2007) para as personagens, onde os polígonos dos modelos são criados um a um. Já para os cenários outra técnica é utilizada, sendo a *Box Modeling*, que consiste da construção de objetos a partir de objetos primitivos, semelhantes a caixas. Para se criar os polígonos de cada modelo, deve-se pensar que, conforme Fox (2004, p. 120) aponta, se tem várias considerações ao criar a topologia do modelo. A primeira e mais óbvia, é assegurar que a distribuição do detalhe na malha permita os contornos que a forma requer. Porém, mais do que isto, a topologia de uma malha precisa fluir nos contornos de sua superfície. Sua topologia precisa ser ditada pelo movimento da superfície que



precisará executar, considerando ainda outro aspecto da topologia quando modelar, é assegurar que a malha seja composta por polígonos com quatro lados ou *quads*, onde for possível.

Deve-se estar atento a todas as questões levantadas por Fox para que assim se possa desenvolver os modelos adequadamente. Sendo um modelo orgânico, ou inorgânico, os conceitos apresentados por Fox (2004, p. 120) acima são extremamente válidos e úteis. Para o projeto Health Simulator, a única exceção que se pontua aqui refere-se ao fato da construção de polígonos totalmente quadrados e de quatro lados, pois em modelos que são usados em ambientes de jogos, polígonos triangulares são viáveis (LIMA, 2011).

Modelos com polígonos apenas de quatro lados são importantes, mas efetivamente para a renderização (apresentação) cinematográfica, vistos em cinema ou mesmo dentro do próprio jogo digital. Segundo Schuytema (2008, p. 412), “o modo mais fácil de apresentar a história aos jogadores é fazer na forma de animações não interativas (conhecidas como *Cinematics*) ou, talvez, como resumos de missão enquanto o jogador disputa uma campanha”. Com isto em mente, pode-se realizar animações para melhor explicar a narrativa do jogo. Para o *Health Simulator*, optou-se por não apresentar cinemáticas.

Elementos de Interfaces

Todo jogo digital é um composto de interfaces, sejam elas de interação do jogador, ou de resposta a este. As interfaces aqui descritas referem-se a aquelas que levam informações ao jogador, denominadas de Menu ou HUD (*Heads-Up Display*). Segundo Schuytema (2008, p. 52) ainda, “esse sistema mostra interfaces, coleta informações do usuário e coordena a transmissão das informações para os sistemas apropriados do game”. Segundo o autor ainda, “o primeiro lugar a começar é com papel e lápis – rabiscando as informações de que precisamos nessa primeira tela de abertura e fazendo alguns rascunhos rápidos de como queremos que seja a tela” (SHUYTEMA, 2008, p. 228).



Antes de se preocupar com a estética dos elementos de interface, deve-se pensar sobre como a ergonomia que esta terá quando finalizada. Segundo Bevan (1995), usabilidade é um termo no qual descreve a qualidade da interação do usuário com a interface apresentada a ele. Considera-se que quando a usabilidade não está devidamente aplicada, alguns problemas surgem, como a dificuldade dos usuários em realizar tarefas na interface, perda de dados e produtividade prejudicada. Existem muitos fatores que podem comprometer a usabilidade de uma interface, como o uso de recursos de formas inadequadas, por isso os princípios que envolvem a usabilidade devem ser cuidadosamente analisados durante o desenvolvimento do projeto.

A fundamentação de ergonomia se sustenta como sendo interfaces referentes a forma da apresentação das informações ao usuário, procurando facilitar a utilização do sistema, garantindo que as habilidades, capacidades e necessidades do usuário sejam levadas em consideração no desenvolvimento do projeto (CYBIS, 2007). Basicamente, a ergonomia preocupa-se com a comunicação do usuário com a máquina. Ou seja, leva em consideração o *hardware* e *software*, pois todos estes influenciam na maneira que o projeto irá se comunicar com os usuários. Esta interface deve ser versátil o suficiente para que qualquer usuário se adapte facilmente ao sistema, sem dificuldades de executar as tarefas almejadas.

METODOLOGIA

Aqui são apresentadas as metodologias utilizadas na construção do projeto de jogo. Sem tais metodologias, o seu desenvolvimento torna-se impraticável.

Método de Produção de Jogos

O método adotado para o desenvolvimento da presente proposta de jogo digital consiste na adoção do método apresentado por Schuytema (2008, p. 12), que em seu livro *Game Design*, destaca que cada jogo passa por diversas etapas, mas estas podem ser agrupadas em apenas três grandes ciclos de desenvolvimento.



Em seu livro, ele mapeia ainda não apenas a forma metodológica de produção de jogos, mas toda a cultura em torno desta. Tais etapas são apresentadas a seguir:

Pré-produção - momento em que se define os aspectos conceituais do jogo, definindo desde aspectos de arte até a programação, bem como a elaboração de documentações e protótipos de jogo são feitas para validar as escolhas feitas;

Produção - neste momento toda a produção de elementos de arte e programação são realizadas concomitantemente. Nesta etapa, a produção de modelos bem acabados e refinados de arte 2D ou 3D são feitos, bem como elementos de escultura digital e animações dos modelos. Na programação, os recursos necessários para que o jogo funcione com todos os elementos juntos são providenciados. Versões finalizadas do jogo são geradas nesta etapa, a fim de validar o andamento do projeto;

Pós-Produção - nesta etapa se finaliza o jogo, adicionando efeitos e corrigindo problemas que eventualmente vão surgindo durante a produção. Por fim faz-se o lançamento do jogo e sua divulgação a fim de alavancar as vendas.

A partir deste método, pode-se desenvolver o projeto de jogo de modo mais controlado e sistematizado. Aliado a este método, sub-métodos são utilizados para produção específica, como a produção de modelos digitais e mesmo a codificações do jogo digital.

Método de Produção de Modelos de Cena

Para se desenvolver modelos de cena para jogos, deve-se compreender o contexto (ambiente) em que se encontra o modelo a ser desenvolvido. Entender a sua real necessidade dentro do projeto são pontos de alta relevância e não devem ser esquecidos.

Dentro de cada uma das fases de produção são adicionadas as práticas ágeis mais pertinentes ao desenvolvimento do jogo, começando pela gestão do projeto que seguirá práticas de *SCRUM* (KEITH, 2010), bem como, em cada uma



das etapas de desenvolvimento do jogo, serão utilizadas as práticas mais pertinentes tanto para arte (LIMA, 2015), como para o desenvolvimento de código de programação (TELES, 2006).

Tanto personagens como modelos de cena utilizam o mesmo método, podendo-se falar de forma unificada. Basicamente utiliza-se o método de Ward (2008) com práticas ágeis (LIMA, 2015) acopladas a seu método. *Ou seja, devem ser aplicados os princípios do pensamento Lean (Lean Thinking ou Pensamento Enxuto)*, definidos por Womack (2004):

Valor – define-se o que é importante desenvolver;

Fluxo de Valor – Segundo Womack (2004, p. 8), “o Fluxo de Valor é o conjunto de todas as ações específicas necessárias para se levar um produto específico ao fim”;

Fluxo – O fluxo (WOMACK, 2004, p. 11) refere-se a focalizar o produto ou serviço e suas necessidades para que seja desenvolvido ou implementado, e não focar em uma ferramenta ou equipamento específico;

Puxar – Para Womack (2004, p. 60), o termo puxar está associado aos princípios de *Lean*, e define que não se deve produzir um bem ou serviço sem que haja demanda pelo cliente (no caso de desenvolvedoras de jogos, sem que alguém da equipe solicite);

Perfeição – O termo perfeição (WOMACK, 2004, p. 85) aqui deve ser entendido como “sinônimo da total eliminação de desperdício”.

Para o desenvolvimento dos modelos tridimensionais, adotou-se o método de Ward (2008), adaptando seu método para o formato de desenvolvimento ágil. Neste formato, é levado em consideração a filosofia e princípios do Pensamento Enxuto (*Lean Thinking*), priorizando-se o cliente (entendido aqui, como sendo o profissional que desenvolve os modelos) e o trabalho focado no projeto como um todo, não suas partes isoladas apenas. A figura 1 demonstra o método de Ward



(2008) sob a ótica ágil, sendo expresso por um gráfico iterativo. Ao centro do gráfico vê-se o que é definido como Caso Controle, que nada mais é do que uma situação hipotética de produção de um modelo tridimensional dentro do ambiente de jogo. Tal situação define os aspectos de produção, pois a partir do entendimento desta situação, passa-se a determinação de seu método de produção.



Figura 1 - Aplicação do método de Ward (2008) pela ótica Ágil.

A figura 1 apresenta também um círculo e três representações de outros círculos encurtados (formato de arco). Cada círculo é identificado por uma cor que demonstra uma iteração do desenvolvimento do modelo tridimensional. Cada iteração se inicia a partir da documentação do que deve ser feito, como forma de registro e gerador de conhecimento para uso em outros casos similares. A progressão de atividades dentro da iteração dá-se em sentido anti-horário, sendo que ao término desta, é o momento de mensurar as Métricas pré-estabelecidas, seguido de *Feedback*. Nesta etapa, é feito o confronto com o Caso Controle e os seus requisitos de projeto, momento a qual o modelo pode retornar ao estágio inicial da iteração para ajustes ou seguir adiante.

Ainda dentro da iteração, Testes dentro da *Engine* são feitos para que se possa avaliar se o modelo possui erros não suportados pela mesma, ocasionando problemas em sua exibição dentro do jogo. Se nos testes dentro da *Engine* o modelo assumir boa apresentação, então pode ser enviado para a próxima



iteração, que assume mesma dinâmica de desenvolvimento. Cada início, meio e fim de uma iteração, assume um ciclo do desenvolvimento e estes ciclos são incrementais, ou seja, a cada ciclo, o modelo recebe mais detalhes.

Terminado o ciclo de iteração Malha Tridimensional, o modelo deve passar a iteração de Otimização que desenvolve os mesmos procedimentos já descritos, bem como as demais iterações de Mapeamento e Texturização. Na figura 1 observa-se também que no término de um ciclo, passa-se ao seguinte e que este, ao ser concluído pode retroceder aos anteriores conforme a necessidade.

Para a iteração de Mapeamento, recursos extras podem ser utilizados, mas a simples forma de manipulação dos arquivos, já pode dar conta disto. A forma como dispor o Mapeamento pode permitir maior flexibilidade ao longo do processo posterior de texturização. Por exemplo, ao utilizar alinhamentos retilíneos do UV, isto facilita a construção da textura. A própria maneira de se construir as texturas também pode contribuir para deixar todo o processo mais característico dos métodos ágeis, usando, por exemplo, formas não destrutivas de texturas (utilização de cores sólidas e padronagens parametrizadas), que oferecem grande flexibilidade de mudanças em projetos envolvendo a personalização de modelos.

Método de Produção de Elementos de Interfaces

A metodologia de desenvolvimento de interfaces utilizada neste trabalho baseia-se no que Garrett (2003) propôs no livro "*The Elements of User Experience: User-centered design for the web and beyond*". A ideia consiste em começar o desenvolvimento por conceitos abstratos e a partir disso chegar a definições concretas. A metodologia de desenvolvimento é dividida em estratégia, escopo, estrutura, esqueleto e superfície, cada qual desenvolvendo uma função específica e importante.

Com base no recorte metodológico proposto por Garrett (2003), a estratégia consiste na pesquisa que envolve todo o projeto. Esta pesquisa consiste na definição do público-alvo, nas necessidades do usuário, o que o projeto quer e precisa comunicar a seus usuários e/ou qual é o real propósito do



desenvolvimento. Estes dados são conhecidos através de uma análise de informações feitas com base nos objetivos do projeto em questão. O escopo é a definição das características do projeto, a manifestação de quais conteúdos, de que forma serão apresentadas e supridas as necessidades do usuário.

No que se refere a estrutura, Garrett (2003) define como sendo o planejamento e a organização dos elementos. Geralmente, a estrutura é elaborada e descrita através de um organograma ou fluxograma a fim de visualizar melhor os resultados esperados do projeto. O esqueleto consiste na parte estrutural das telas, menus e os elementos que constituem o projeto. Nesta etapa surgem questões de navegação do usuário pela plataforma ou projeto desenvolvido, bem como o design gráfico que será proposto. O método gráfico que representa o esqueleto é definido por *wireframes* de navegação.

A partir do método de Garret (2003), Meurer e Szabluck (2008) definiram o método de Projeto E, que é uma evolução a partir de Garret (2003), visto na figura a seguir.



Figura 2 - Método de desenvolvimento de interfaces segundo Meurer & Szabluck (2008).

A metodologia de Projeto E, desenvolvida por Meurer e Szabluk (2008) consiste em uma série de técnicas e métodos funcionais para o desenvolvimento de um projeto dígito-virtual do início ao fim. Essas técnicas e métodos foram separadas em categorias de desenvolvimento, sendo estas, estratégia, escopo,



estrutura, esqueleto, estética e execução. Cada categoria dessas possui uma separação de etapas que conceitua cada etapa. Para a produção das interfaces gráficas de *Health Simulator*, o método de Projeto E foi adotado.

EXPERIMENTO

Como experimento, apresenta-se aqui algumas categorias de modelos tridimensionais, animações e elementos de interfaces produzidos, com a utilização dos métodos de desenvolvimento descritos acima. Cada um dos tipos de experimentos aqui apresentados, possui suas próprias características e deve ser observado seus requisitos de projeto e execução. A seguir apresentam-se as categorias de experimentos.

Cenários

Para os cenários, foi utilizado um sistema comutativo, classificados como consultórios de classe A, B ou C e um Hospital Único de Saúde. Estão em fase de implementação 6 cenários, com o total de 132 objetos para todos os cenários. Segue-se uma contagem poligonal de 7.000 à 9.000 triângulos por cena, totalizando uma média de uma textura por objeto, que poderão compor novos cenários através do reaproveitamento de componentes. A imagem a seguir apresenta alguns resultados de construção de cenários.



Figura 3 - Amostras de resultados para cenários.



Os cenários são produzidos com o Autodesk 3Ds Max e suas texturas são produzidas com o Adobe Photoshop. Os modelos são produzidos primeiramente por seus espaços maiores (constituindo a cena em si, composta de piso, paredes e forro). Depois são agregados diferentes elementos de cena para composição desta, observando a classe a que pertence o cenário (classe A ou C). As texturas são incluídas a seguir, sendo a implementação na *Engine* de Jogo feita praticamente no final do processo. Todos os cenários são preparados para receber mapas de luz do tipo *Lightmap* que aumentam o realismo das cenas.

Personagens

A categoria de personagens conta hoje com 4 tipos (médicos, pacientes, dentistas e enfermeiros), de gêneros masculino e feminino. São construídos com idades diferenciadas (bebê, criança, jovem, adulto e idoso) e etnias (caucasiano, oriental, afrodescendente e indiano), já estão concluídos 81 personagens femininos e 131 personagens masculinos.

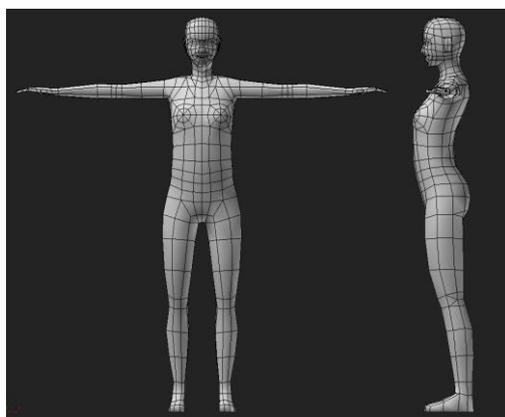


Figura 4 - Amostras de resultados para as personagens.

Com a utilização de práticas ágeis associadas a produção de modelos tridimensionais foi possível, com uma equipe relativamente enxuta, se produzir o volume de modelos apresentados neste artigo. Tais práticas viabilizaram a



produção necessária para o aplicativo de jogo, de acordo com a mão-de-obra disponível.

Animações

O projeto *Health Simulator* segue as regras básicas de animação da Disney para a construção de movimentos animados. Tais leis foram desenvolvidas por animadores do estúdio, entre eles, Frank Thomas e Ollie Johnston⁷, durante os anos 30. São ao todo 12 leis básicas:

- Squash & Stretch (dá a ilusão de peso e volume aos personagens enquanto se movem);

- Staging (a apresentação da ideia principal na cena deve ser clara e objetiva);

- Anticipation (prepara o público para a ação que vai acontecer em seguida);

- Straight ahead e Pose to Pose (“Straight ahead” significa desenhar um quadro a quadro na cena do começo ao fim, enquanto “pose to pose” envolve iniciar com o desenho de alguns quadros-chave e, em seguida, preencher os intervalos entre eles.);

- Follow through e Overlapping (“Follow through” é fazer com que partes do corpo do personagem continuem em movimento mesmo quando sua ação é interrompida. “Overlapping” significa que as partes do corpo de um personagem devem se mover em velocidades diferentes);

- Slow in & Slow out (os corpos precisam de aceleração e desaceleração. Para simular isso em animação, são necessários mais quadros desenhados no começo e no fim da ação e menos no meio. Quanto mais quadros desenhados

⁷ Unip. **12 Princípios Fundamentais da Animação**. Disponível em:
<http://adm.online.unip.br/img_ead_dp/30561.PDF> Acessado em 6 de junho de 2016 as 10 horas.



entre o fim e o começo fazem da velocidade mais reduzida nestes momentos. Menos quadros desenhados, mais rápido o movimento);

- *Arcs* (no mundo real, os movimentos dos objetos tendem a seguir uma trajetória de arco. No desenho, o mesmo princípio é aplicado para dar maior realismo);

- *Secondary actions* (adicionar ações secundárias à ação principal pode dar mais naturalidade à cena, e apoiar a ação principal. Lembrando que ela sempre acontece para reforçar a ação que deve ser a estrela da cena, e não para distrair o espectador);

- *Timing* (refere-se ao número de desenhos ou quadros para uma determinada ação, o que se transforma em velocidade no filme);

- *Exageration* (serve não para distorcer a realidade, mas para dar a ela um reforço, de uma forma mais extrema);

- *Solid Drawing* (significa levar em conta as formas no espaço tridimensional na hora de desenhar um objeto ou personagem, dando a eles volume e peso);

- *Appeal* (o apelo é quase a mesma coisa que o carisma em um ator. O espectador deve sentir que a personagem seja real e interessante, mesmo que ele precise ser “feio” ou “mau”, como um vilão).

Para realizar os movimentos animados seguindo estes princípios, os modelos de personagens foram preparados desde sua construção de modelagem, até mesmo sua estrutura óssea e os controladores de deformidade de malha para movimento animado. A figura a seguir mostra alguns modelos com poses e seus corpos já preparados para isto.

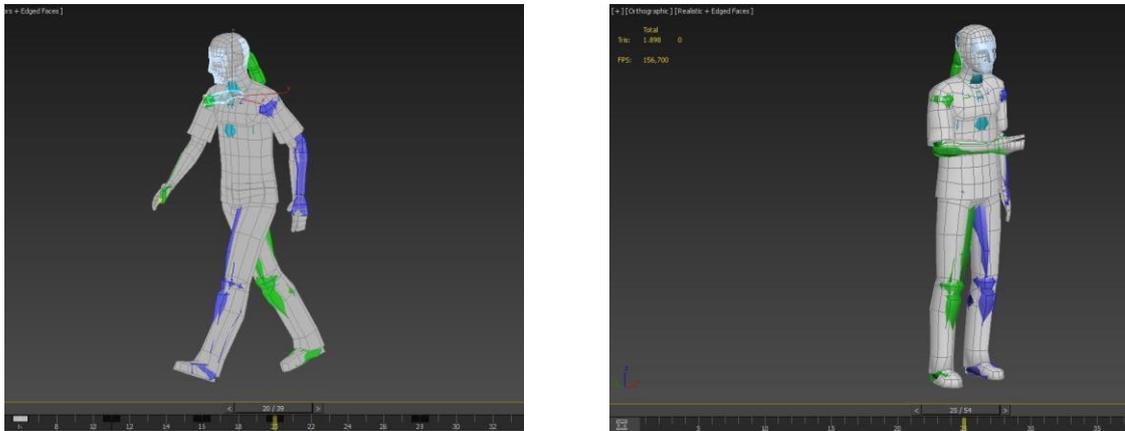


Figura 5 - Amostras de resultados para animações.

Da mesma forma que a construção de malha se apropriou de práticas ágeis para desenvolver de modo mais ágil, o processo de animação também se beneficiou destas práticas. Foram desenvolvidos os movimentos a partir de um único esqueleto para o gênero masculino, e um para o feminino. Todos os modelos de um gênero compartilham os mesmos movimentos, o que permitiu a equipe dedicar-se mais a qualidade dos movimentos e menos a quantidade, haja visto a quantidade de personagens produzidos, já citada neste artigo. O uso de um único esqueleto para os gêneros permitiu a adoção desta técnica, aliado a produção de modelagem que sempre se preocupou com os movimentos animados, buscando meios de evitar retrabalho quando os modelos de fato fossem animados.

Resultados Parciais

O *Health Simulator* tem previsão de finalização para o início de 2017, quando se pretende realizar validações com professores e alunos na área da saúde. Após os testes, ajustes serão feitos para melhorar a qualidade do jogo. No total, conforme já mencionado, estão concluídos 81 personagens femininos e 131 masculinos. De cenários estão concluídos (ou em fase de conclusão) 6 de 19 cenários. Estão concluídos em média 132 objetos de cena, com no mínimo uma textura para cada.



CONSIDERAÇÕES FINAIS

Jogos do tipo simulador virtual têm se revelado uma boa estratégia a ser utilizado no processo de aprendizagem para os alunos na área da saúde. Com este tipo de material, pode-se explorar diversas formas de ambientes e experiências, além de desenvolver habilidades cognitivas. O aluno aprenderá os passos a serem seguidos em atendimentos clínicos através da simulação de casos, de uma forma mais segura, aprendendo com os seus erros e sem prejudicar um paciente real. Pode-se jogar simuladores no início de um curso, realizando tarefas e atividades com a sua dificuldade aumentando gradativamente, permitindo encontrar problemas de sua área, tendo assim, melhor atuação futura.

REFERÊNCIAS

Designerd. **Design de Games: Você sabe o que é HUD?**. Disponível em: <<http://www.designerd.com.br/design-de-games-voce-sabe-o-que-e-hud/>>. Acessado em 30 de maio de 2016 as 9 horas.

Unip. **12 Princípios Fundamentais da Animação**. Disponível em: <http://adm.online.unip.br/img_ead_dp/30561.PDF>. Acesso em: 6 de junho de 2016 as 10 horas.

BECK, Kent. **Programação eXtrema (xp) explicada: acolha as mudanças**. Porto Alegre: Boockman, 2004.

BEZ, M. R. **Construção de um Modelo para o Uso de Simuladores na Implementação de Métodos Ativos de Aprendizagem das Escolas de Medicina**. Porto Alegre, 2013. 314 f. Tese (Doutorado PGIE/CINTED - UFRGS, Porto Alegre).

LACERDA, Guilherme Silva De; WILDT, Daniel De Freitas; RIBEIRO, Vinicius Gadis. **Uma Introdução às Metodologias Ágeis de Software**. 2004.

LACERDA, Guilherme S. et al. Adoção do CMMI e das metodologias ágeis em empresas brasileiras. **Revista Avances en Sistemas de Informática**, v. 8, n. 3, 2011.

LIMA, Alessandro. **Design de Personagens para Games Next-Gen V1**. 1. ed. Rio de Janeiro: Ciência Moderna, 2011.



MASSON, Terezinha Jocelen. MIRANDA, Leila Figueiredo de. JUNIOR, Antônio Hortêncio Munhoz. CASTANHEIRA, Ana Maria Porto. **Metodologia de Ensino: Aprendizagem Baseada em Projetos (PBL)**. São Paulo: Cobenge - XL Congresso Brasileiro de Educação em Engenharia, 2012.

NOVAK, Jeannie. **Desenvolvimento de Games**. 2 ed. ed. São Paulo: Cengage Learning, 2010.

ORTON, E.; MULHAUSEN, P. (2008) E-learning virtual patients for geriatric education. **Gerontology & Geriatrics Education**, v. 28, n. 3, p.73-88.

SCHUYTEMA, Paul. **Design de Games: uma abordagem prática**. São Paulo: Cengage Learning, 2008.

TELES, Vinícius Magalhães. **Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. São Paulo: Novatec, 2006.

TSUJI, H.; SILVA, R. H. A. (2010) **Aprender e ensinar na escola vestida de branco: do modelo biomédico ao humanístico**. São Paulo: Phorte, 2010. 240p.

WARD, Antony. **Game Character Development**. Boston: Cengage, 2008.