

DESENVOLVIMENTO DE UM APLICATIVO EM REALIDADE VIRTUAL PARA MODELOS ÓSSEOS ATRAVÉS DO UNITY

Luis T. G. Pinto¹, Lucas M. Zonta², Alexandre M. L. Ferreira², Marcus V. C. Calca³, José Rafael Pilan³.

¹Pós-Graduando em Desenvolvimento de Sistemas Web pela Universidade do Sagrado Coração.

²Graduandos em Análise e Desenvolvimento de Sistemas pela Faculdade de Tecnologia de Botucatu.

³Mestrando em Agronomia pela Faculdade de Ciências Agronômicas - UNESP/Botucatu

⁴Professor Mestre, Faculdade de Tecnologia de Botucatu.

1. INTRODUÇÃO

A grande maioria dos alunos dos cursos de biologia, radiologia e demais cursos desses segmentos possuem uma base teórica extensa e se torna muito rotineira a utilização de livros e/ou apostilas tornando algumas vezes entediante e árduo seus estudos. Desta forma, torna-se maçante estudar um modelo ósseo por fotos, vídeos ou softwares ultrapassados, que não possuem uma interface amigável (SILVA E SANTANA, 2012).

Segundo Montes e Souza (2017), em cursos de enfermagem, onde a anatomia é ensinada em dois semestres letivos, as taxas de reprovação na disciplina no primeiro período variaram de 43,13% no segundo semestre de 2006, e 55,15% no primeiro semestre de 2006, uma das causas de tais apontamentos é a extensa carga de estudos, onde utiliza-se conteúdos desestimulantes para a aprendizagem. Nem sempre o aluno poderá ter acesso a modelos ósseos físicos, uma vez que o manuseio de cadáveres é trabalhoso, é de difícil disponibilidade devido à grande quantidade de alunos, e pode ser traumatizante em muitas ocasiões.

A Realidade Virtual (RV) pode cooperar no aprendizado do aluno, agregando valor como uma ferramenta de disponibilização de conteúdo de forma intuitiva, pois permite a imersão em um mundo virtual criado via software. A RV é uma “interface avançada do usuário” para acessar aplicações executadas em um computador, tendo como características a visualização de modelos em 3D e movimentação em ambientes tridimensionais em tempo real, e a interação com elementos de um ambiente simulado (TORI; KIRNER; SISCOOTTO, 2006).

O objetivo deste trabalho foi desenvolver um aplicativo para facilitar o entendimento e aprendizado de modelos ósseos utilizando a RV.

2 MATERIAL E MÉTODOS

2.1 UNITY 5.5

O Unity 5.5 é um motor de jogos muito utilizado no segmento de entretenimento, sobre o qual o jogo é desenvolvido. É um item básico para o desenvolvimento de um bom jogo, na atualidade.

Um jogo é um software, e como tal, necessita de três partes básicas, que são: um componente de saída de dados, como por exemplo um monitor ou um óculos RV, um componente para a entrada de dados, como por exemplo um controle de jogos, e o último componente deve receber comandos do jogador e executar as regras e guardar os estados dos usuários que é a máquina de estados (GREGORY, 2009).

2.2 GOOGLE VR

O Google VR engloba o *Daydream* e *Cardboard*, que são duas plataformas RV de propriedade da Google. Os aplicativos do Google VR são desenvolvidos usando *Daydream*. *Daydream for Unity* é uma prévia técnica projetada para oferecer aos desenvolvedores da Unity acesso inicial ao desenvolvimento do Google VR. (UNITY DOCUMENTATION, 2017)

2.3 LINGUAGEM DE PROGRAMAÇÃO C#

O C# é uma linguagem de programação visual dirigida por eventos e orientada a objetos, na qual utiliza uma IDE (*Integrated Development Environment*) para desenvolver um programa de computador. A IDE permite que o programador crie, execute e depure programas em C# convenientemente (DEITEL. et al., 2003).

Também permite um novo grau de intercâmbio entre linguagens, os componentes de softwares de diferentes linguagens podem interagir entre si. Os desenvolvedores podem empacotar até software antigo, para trabalhar com nos programas em C# (DEITEL. et al., 2003).

A linguagem de programação C# permitiu a realização de *scripts* para a movimentação do usuário dentro do ambiente virtual e a possibilidade de movimentação do modelo ósseo 3D.

2.4 VISUAL STUDIO 2015

É uma IDE (*Integrated Development Environment*) que tem o objetivo de agilizar o desenvolvimento de programas de computador, reconhecer erros de sintaxe e auxiliar no desenvolvimento ágil, com autopreenchimento de comandos e funções (MADUREIRA JUNIOR, 2016).

Para executar uma aplicação no C#, o código é compilado e armazenado em disco como um arquivo executável. O arquivo gerado é intermediário e denominado *Common Intermediate Language* (CIL). Para a execução, ele deve ser carregado e interpretado pela máquina virtual, *Common Language Runtime* (CLR), que tem por finalidade tomar as ações específicas a fim de executar o código de maneira independente, sem a necessidade da ferramenta de desenvolvimento. A função do CLR, é a compilação *just in time* (JIT), que consiste em converter o CIL em instruções de máquina. O CLR oferece outros serviços como a coleta de lixo automática, tratamento de exceções e gerenciamento de recursos a serem utilizados no processo (MADUREIRA JUNIOR, 2016).

2.5 PROCESSOS DE DESENVOLVIMENTO

No Unity desenvolveu-se um ambiente virtual totalmente estático, ao inserir o GoogleVR no Unity pode-se gerar uma interação ente o usuário e o ambiente virtual, para trazer mais dinamismo e aplicar movimentação ao usuário. Desenvolve-se alguns algoritmos usando o Visual Studio. Para o desenvolvimento desses algoritmos foi utilizada a linguagem de programação C#, ao final destes processos foi possível gerar um aplicativo de RV. A Figura 1, a seguir, ilustra estes processos.

Figura 1 - Processos de Desenvolvimento.

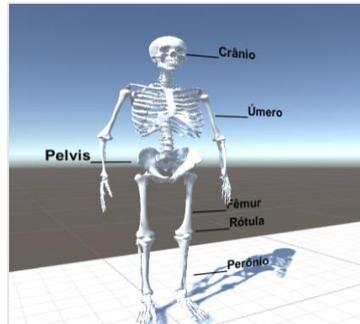


3 RESULTADOS E DISCUSSÕES

A primeira fase do projeto tratou de pesquisar diversos modelos de um esqueleto ósseo humano de qualidade e ao final desta fase escolheu-se um modelo do site free3d.com (2017) que é um modelo gratuito que precisou de pequenos ajustes para se

tornar funcional e colocando a nomenclatura dos devidos ossos como pode-se observar na Figura 2, a seguir.

Figura 2 - Modelo Ósseo.



A próxima etapa deste projeto tratou do desenvolvimento do cenário onde foram utilizados componentes para realizar a colisão de objetos, evitando que os *Game Objects* caiam ou saiam da cena do jogo. A Figura 3, a seguir, ilustra o cenário.

Figura 3 – Cenário.



Em seguida foi possível implementar realidade virtual utilizando o pacote do Google VR, onde este cria a possibilidade de utilização por meio de um óculos de VR, como ilustra a Figura 4, a seguir.

Figura 4 - Óculos VR.



Fonte: OCULUS (2017).

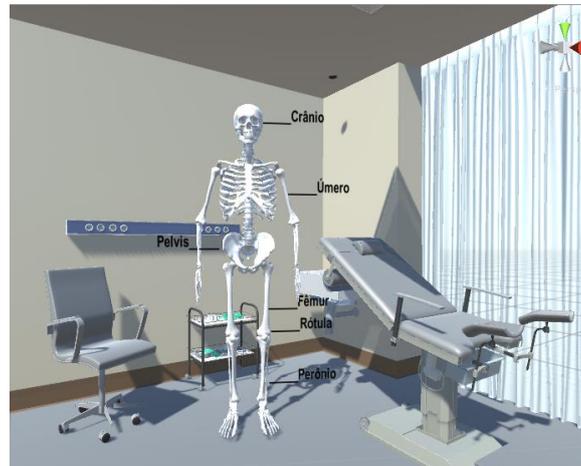
Para deixar o aplicativo mais dinâmico foi implementado *scripts* em C# com o intuito de movimentar o modelo ósseo e o jogador dentro do ambiente virtual, a Figura 5 ilustra o *script* de movimentação básica dentro do ambiente virtual.

Figura 5 - Script Movimentação.

```
5 public class Movimentos : MonoBehaviour {
6     public float velocidadeMov;
7     public float velocidadeRot;
8     public GameObject player;
9     // Use this for initialization
10    void Start () {
11        velocidadeMov = 1.0f;
12        velocidadeRot = 10.0f;
13    }
14
15    // Update is called once per frame
16    void Update () {
17        if(Input.GetButton("w")){
18            transform.Translate (0,0,velocidadeMov);
19        }
20        if(Input.GetButton("s")){
21            transform.Translate (0,0,-velocidadeMov);
22        }
23        if(Input.GetButton("a")){
24            transform.Rotate (0,velocidadeRot,0);
25        }
26        if(Input.GetButton("d")){
27            transform.Rotate (0,-velocidadeRot,0);
28        }
29
30        if(Input.GetButton("z")){
31            player.SetActive (false);
32        }
33        if(Input.GetButton("x")){
34            player.SetActive (true);
35        }
36    }
37 }
38 }
```

O resultado é ilustrado na Figura 6, onde o usuário pode interagir com o ambiente virtual caminhando dentro de uma sala virtual e até mesmo movimentando o modelo ósseo.

Figura 6 - Ambiente Virtual.



4 CONCLUSÕES

Foi possível criar um aplicativo de RV onde o usuário pôde se movimentar em um ambiente totalmente criado por software, e o visualizar através de um óculos de RV. A movimentação do modelo ósseo pôde ser realizada através de um controle. A aplicação da RV mostrou-se intuitiva e de fácil manuseio para qualquer usuário, utilizando as ferramentas sem dificuldades, independentemente do grau de conhecimento em Tecnologia da Informação.

5 REFERÊNCIAS

- DEITEL, M. H. et al. **C# Como programar**. São Paulo: Pearson Education, 2003. 1158 p.
- GREGORY, J. **Game Engine Architecture**. A K Peters/CRC Press, 2009.
- FREE3D, All 3d models – Free 3D. Disponível em: <<https://free3d.com/3d-model/skeleton-81335.html>>. Acesso em: 28 ago. 2017.
- MADUREIRA JUNIOR, J. R. et al. **Introdução ao C# com o Microsoft Visual Studio Community 2015**. São Paulo: Perse, 2016. 47 p.
- MONTES, M; SOUZA, C. **Análise da Taxa de Reprovação na Disciplina de Anatomia Humana em cursos da Saúde**. Disponível em: <<http://www.nutes.ufrj.br/abrapec/vienpec/CR2/p45.pdf>>. Acesso em: 28 ago. 2017.
- OCULUS. Disponível em: <<https://developer.oculus.com>>. Acesso em: 27 ago. 2017.
- SILVA, K. D. C. D; SANTANA, O. A. **Objetos de aprendizagem utilizado para o ensino de anatomia humana: uma revisão da literatura**. Disponível em: <<http://www.abed.org.br/congresso2012/apresentacaoposters18ciaed.pdf>>. Acesso em: 28 ago. 2017
- TORI, R; KIRNER, C; SISCOOTTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. In: Symposium On Virtual Reality, 2006.



FACULDADE DE TECNOLOGIA DE BOTUCATU

*6ª Jornada Científica e Tecnológica da FATEC de Botucatu
23 a 27 de Outubro de 2017, Botucatu – São Paulo, Brasil*



UNITY DOCUMENTATION. **Google VR**. Disponível em:

<<https://docs.unity3d.com/Manual/VRDevices->

[GoogleVR.html?_ga=2.75079954.1225177919.1506266617-1411752953.1504441178](https://docs.unity3d.com/Manual/VRDevices-GoogleVR.html?_ga=2.75079954.1225177919.1506266617-1411752953.1504441178)>. Acesso em: 28 ago. 2017